Contracting optimally an interval matrix without loosing any positive semi-definite matrix is a tractable problem

Luc Jaulin[1] and Didier Henrion[2,3],

[1] : Laboratoire l'Ingénierie des Systèmes Automatisés, Université d'Angers, France.

[2] : Laboratoire d'Analyse et d'Architecture des Systèmes, Centre National de la Recherche Scientifique, Toulouse, France

[3] : Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Prague

jaulin@univ-angers.fr

henrion@laas.fr

Abstract: In this paper, we show that the problem of computing the smallest interval submatrix of a given interval matrix $[\mathbf{A}]$ which contains all symmetric positive semi-definite (PSD) matrices of $[\mathbf{A}]$, is a *linear matrix inequality* (LMI) problem, a convex optimization problem over the cone of positive semidefinite matrices, that can be solved in polynomial time. From a constraint viewpoint, this problem corresponds to projecting the global constraint $\text{PSD}(\mathbf{A})$ over its domain $[\mathbf{A}]$. Projecting such a global constraint, in a constraint propagation process, makes it possible to avoid the decomposition of the PSD constraint into primitive constraints and thus increases the efficiency and the accuracy of the resolution.

Keywords: constraint propagation, interval analysis, interval matrix, linear matrix inequalities, optimization, positive semi-definite constraint.

1. Introduction

Many problems of estimation, control, robotics, ... can be represented by continuous *constraint satisfaction problems* (CSP) [10, 16]. A CSP is composed of a set of variables $\mathcal{V} = \{x_1, \dots, x_n\}$, a set of constraints $\mathcal{C} = \{c_1, \dots, c_m\}$ and a set of interval domains $\{[x_1], \dots, [x_n]\}$. The aim of propagation techniques is to contract as much as possible the domains for the variables without loosing any solution [3, 19, 21]. Denote by $[\mathbf{x}]$ the box defined by the Cartesian product of all domains and by $[\mathbf{x}]\sqcap c_j$, the smallest box which contains all points in $[\mathbf{x}]$ that satisfy c_j. The operator $\sqcap$ will be called *square intersection*. The principle generally used to contract the $[x_i]$'s is *arc consistency*. It consists in computing the box

$$((((((\mathbf{[x]}\sqcap c_1) \sqcap c_2) \sqcap \ldots) \sqcap c_m) \sqcap c_1) \sqcap c_2) \ldots, \tag{1.1}$$

until a steady box (also called the fixed point) is reached.

When no algorithm is available for computing $[\mathbf{x}]\sqcap c_j$, the constraint c_j should be decomposed into constraints $c_{j1}, c_{j2}, \ldots$ on which the square intersection $\sqcap$ can be computed. When such a decomposition is performed, the steady box that is reached is generally much bigger than the one that would have been obtained without the decomposition. Extending the class of constraints for which the square intersection $\sqcap$ can be computed efficiently is therefore an important task that should be considered in the constraint community.

In this paper, we consider the constraint *positive semi-definite (PSD)* for matrices, *i.e.*, for a given interval matrix $[\mathbf{A}]$, we shall provide a polynomial algorithm which computes the smallest interval matrix which contains all positive semi-definite matrices of $[\mathbf{A}]$. The PSD constraint often occurs in control theory (see e.g., [17, 15]) or in optimization (see *e.g.* the non-convexity check in [8]), but, to our knowledge it has never been considered in the constraint propagation community. The approach to be proposed is based on linear matrix inequalities (LMI) briefly presented in Section 2. Some important notions and properties of intervals in lattices and sublattices are given in Section 3. These notions will be used in Section 4 to establish some new links between interval matrices and interval symmetric matrices. Section 5 provides a polynomial algorithm that solves our problem. An illustrative example is given in Section 6.

2. Linear matrix inequalities

This section presents some notions of linear matrix inequalities. A much more detailed presentation can be found in [2] and in [4]. Denote by $\mathcal{M}^n$ the set of all matrices of $\mathbb{R}^{n\times n}$. $\mathcal{M}^n$ is a vector space with dimension n^2. Its canonical basis is $\{\mathbf{E}^{ij}\}_{i,j\in\{1,\ldots,n\}}$, where $\mathbf{E}^{ij}$ is the matrix with zeros everywhere except the (i,j) entry which is equal to 1. The set

$$\mathcal{S}^n \triangleq \left\{\mathbf{A}\in\mathcal{M}^n | \mathbf{A}=\mathbf{A}^{\mathrm{T}}\right\}, \tag{2.1}$$

of all symmetric matrices of $\mathcal{M}^n$ is a vector space isomorphic to $\mathbb{R}^{\frac{n(n+1)}{2}}$. The family $\left\{\mathbf{E}_{\mathrm{S}}^{ij}\right\}_{j\geq i}$, where

$$\mathbf{E}_{\mathrm{S}}^{ij} = \left(\mathbf{E}^{ij}+\mathbf{E}^{ji}\right) \text{ if } i\neq j \text{ and } \mathbf{E}_{\mathrm{S}}^{ij}=\mathbf{E}^{ij} \text{ otherwise} \tag{2.2}$$

is the *canonical basis* of $\mathcal{S}^n$.

Example 1. *The canonical basis of $\mathcal{S}^2$ is*

$$\mathbf{E}_S^{11} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \mathbf{E}_S^{12} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \mathbf{E}_S^{22} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \tag{2.3}$$

Definition 1. *A matrix $\mathbf{A}$ of $\mathcal{S}^n$ is* positive semi-definite *(PSD), denoted by $\mathbf{A} \succeq 0$, if*

$$\forall \mathbf{z} \in \mathbb{R}^n, \mathbf{z}^T \mathbf{A} \mathbf{z} \geq 0. \tag{2.4}$$

Theorem .1. *The set of PSD matrices $\mathcal{S}_+^n \triangleq \{\mathbf{A} \in \mathcal{S}^n | \mathbf{A} \succeq 0\}$ is a convex cone*[1] *of $\mathcal{S}^n$.*

Proof: We have

$$\begin{aligned} \mathcal{S}_+^n &= \left\{\mathbf{A} \in \mathcal{S}^n | \forall \mathbf{z} \in \mathbb{R}^n, \mathbf{z}^{\mathrm{T}} \mathbf{A} \mathbf{z} \geq 0\right\} & (2.5) \\ &= \bigcap_{\mathbf{z} \in \mathbb{R}^n} \left\{\mathbf{A} \in \mathcal{S}^n \mid \mathbf{z}^{\mathrm{T}} \mathbf{A} \mathbf{z} \geq 0\right\}. & (2.6) \end{aligned}$$

Now, for a given $\mathbf{z} \in \mathbb{R}^n$, we have the following equivalences

$$\mathbf{z}^{\mathrm{T}} \mathbf{A} \mathbf{z} \geq 0 \Leftrightarrow \sum_{i,j} z_i z_j a_{ij} \geq 0 \Leftrightarrow \sum_i z_i^2 a_{ii} + 2 \sum_{j>i} z_i z_j a_{ij} \geq 0. \tag{2.7}$$

Thus, $\mathcal{S}_+^n$ is an intersection of an infinite number of half-spaces of $\mathbb{R}^{\frac{n(n+1)}{2}}$. As a result, $\mathcal{S}_+^n$ is a cone of $\mathcal{S}^n$ the vertex of which is zero. ■

Definition 2. *A linear matrix inequality (LMI) has the form*

$$\mathbf{A}(\mathbf{x}) \triangleq \mathbf{A}_0 + x_1 \mathbf{A}_1 + \cdots + x_m \mathbf{A}_m \succeq 0, \tag{2.8}$$

where $\mathbf{x} \in \mathbb{R}^m$ is a vector of variables and the matrices $\mathbf{A}_i$ all belong to $\mathcal{S}^n$. An LMI set *is a subset of $\mathbb{R}^m$ which can be defined by an LMI.*

The following theorem illustrates some well-known properties of LMI sets.

Theorem .2. *An LMI set is convex and the intersection of two LMI sets is an LMI set.*

[1] Recall that a *cone* of $\mathbb{R}^n$ with vertex $\mathbf{x}_0$ is a subset $\mathcal{C}$ of $\mathbb{R}^n$ such that $\mathbf{v} \in \mathcal{C} \Rightarrow \forall \alpha > 0, \mathbf{x}_0 + \alpha (\mathbf{v} - \mathbf{x}_0) \in \mathcal{C}$.

Proof: The LMI set $\mathbb{S} \triangleq \{\mathbf{x} \in \mathbb{R}^m | \mathbf{A}_0 + x_1\mathbf{A}_1 + \cdots + x_m\mathbf{A}_m \succeq 0\}$ can be written as $\mathbb{S} = \mathbf{f}^{-1}\left(\mathcal{S}_+^n\right)$,where

$$\mathbf{f} : \begin{cases} \mathbb{R}^m & \rightarrow \quad \mathcal{M}^n \\ \mathbf{x} & \mapsto \quad \mathbf{A}_0 + x_1\mathbf{A}_1 + \cdots + x_m\mathbf{A}_m \end{cases} \tag{2.9}$$

is affine. Since the reciprocal set of a convex set by an affine function is convex, $\mathbb{S}$ is convex.

A block diagonal matrix is PSD if and only if all its blocks are PSD. Thus, the intersection $\mathbb{S}_a \cap \mathbb{S}_b$ of the two LMI sets

$$\mathbb{S}_a = \{\mathbf{x} \in \mathbb{R}^m \mid \mathbf{A}_0 + x_1\mathbf{A}_1 + \cdots + x_m\mathbf{A}_m \succeq 0\} \tag{2.10}$$
$$\mathbb{S}_b = \{\mathbf{x} \in \mathbb{R}^m \mid \mathbf{B}_0 + x_1\mathbf{B}_1 + \cdots + x_m\mathbf{B}_m \succeq 0\} \tag{2.11}$$

is given by

$$\begin{aligned} & \left\{\mathbf{x} \in \mathbb{R}^m \mid \begin{pmatrix} \mathbf{A}_0 + x_1\mathbf{A}_1 + \cdots + x_m\mathbf{A}_m & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_0 + x_1\mathbf{B}_1 + \cdots + x_m\mathbf{B}_m \end{pmatrix} \succeq 0\right\} \\ = & \left\{\mathbf{x} \in \mathbb{R}^m \mid \begin{pmatrix} \mathbf{A}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_0 \end{pmatrix} + x_1 \begin{pmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_1 \end{pmatrix} + \cdots + x_m \begin{pmatrix} \mathbf{A}_m & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_m \end{pmatrix} \succeq 0\right\} \end{aligned}$$

which is an LMI set. ■

We now give three examples of constraints that can be defined by LMIs.

Example 2. *A constraint of the form* $\mathbf{x} \in [\mathbf{x}]$ *is an LMI. For instance, the constraint* $x_1 \in [1, 2]; x_2 \in [3, 4]$ *can be written in an LMI form as*

$$\begin{pmatrix} x_1 - 1 & 0 & 0 & 0 \\ 0 & 2 - x_1 & 0 & 0 \\ 0 & 0 & x_2 - 3 & 0 \\ 0 & 0 & 0 & 4 - x_2 \end{pmatrix} \succeq 0, \tag{2.12}$$

i.e.,

$$\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix} + x_1 \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + x_2 \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \succeq 0. \tag{2.13}$$

Example 3. *A set of linear inequalities is an LMI. For instance*

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + b_1 & \geq 0 \\ a_{21}x_1 + a_{22}x_2 + b_2 & \geq 0 \end{cases}$$

is equivalent to the following LMI

$$\begin{pmatrix} a_{11}x_1 + a_{12}x_2 + b_1 & 0 \\ 0 & a_{21}x_1 + a_{22}x_2 + b_2 \end{pmatrix} \succeq 0, \tag{2.14}$$

i.e.,

$$\begin{pmatrix} b_1 & 0 \\ 0 & b_2 \end{pmatrix} + x_1 \begin{pmatrix} a_{11} & 0 \\ 0 & a_{21} \end{pmatrix} + x_2 \begin{pmatrix} a_{12} & 0 \\ 0 & a_{22} \end{pmatrix} \succeq 0. \tag{2.15}$$

Example 4. *An ellipsoid of $\mathbb{R}^n$ is an LMI set. To get the LMI associated with an ellipsoid, we can use the Schur complement theorem (see [2, 4]) which claims that, for any matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ with appropriated dimensions,*

$$\begin{cases} \mathbf{C} & \succ 0 \\ \mathbf{A} - \mathbf{B}^T\mathbf{C}^{-1}\mathbf{B} & \succeq 0 \end{cases} \Leftrightarrow \begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{C} \end{pmatrix} \succeq 0. \tag{2.16}$$

Here, $\mathbf{A} \succ 0$ means that all eigenvalues of $\mathbf{A}$ are all strictly positive. Note that a constraint of the form $\mathbf{A} \succ 0$ can be approximated by $\mathbf{A} \succeq \varepsilon$ where $\varepsilon > 0$ is as small as desired. As an illustration, consider the ellipse defined by $3x_1^2 + 2x_2^2 - 2x_1x_2 \leq 5$. We have

$$\begin{aligned} & 3x_1^2 + 2x_2^2 - 2x_1x_2 \leq 5 \\ \Leftrightarrow \quad & 5 - \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^T \begin{pmatrix} 3 & -1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \geq 0 \\ \Leftrightarrow \quad & 1 - \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^T \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}^{-1} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \geq 0. \end{aligned} \tag{2.17}$$

Using the Schur complement theorem with

$$\mathbf{A} = 1, \mathbf{B} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \text{ and } \mathbf{C} = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix} \succ 0, \tag{2.18}$$

we get the LMI

$$\begin{pmatrix} 1 & x_1 & x_2 \\ x_1 & 2 & 1 \\ x_2 & 1 & 3 \end{pmatrix} \succeq 0, \tag{2.19}$$

i.e.,

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 3 \end{pmatrix} + x_1 \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + x_2 \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \succeq 0. \tag{2.20}$$

Many other convex sets can be represented by LMIs. Even though the general problem of knowing whether a given semialgebraic convex set admits an LMI formulation remains open, the excellent textbooks [2, 4], collect an impressive amount of LMI-representable geometric sets (*e.g.,* ellipses, parabolas or disks) or more general convex sets relevant to control engineering, structural optimization or combinatorial optimization.

Following the seminal work of Karmarkar [11] presenting a polynomial-time algorithm for solving linear programming problems, a lot of research activity focused on extending these results to more general convex optimization problems. This culminated in the manuscript [13] where general interior-point methods are described that can be used to solve LMI optimization problems (amongst others) in polynomial-time at any given accuracy. Since LMI problems are generalization of linear programming problems to the cone of PSD matrices, LMI programming is generally referred to as semidefinite programming in the technical literature. A projective method based on the results of [13] and having worst-case complexity $O\left(m^3 n \log\left(\frac{1}{\varepsilon}\right)\right)$, where ε is the required relative accuracy, was first implemented in the INRIA Scilab freeware [6], and then in the commercial LMI Toolbox for Matlab [7]. Primal-dual interior-point algorithms were also designed for LMI problems, see [20] for a survey and [18] for a high-quality solver called SeDuMi having worst-case complexity $O\left((n^{3.5}m + n^{2.5}m^2)\log\left(\frac{1}{\varepsilon}\right)\right)$. In practice, most of the computational time is spent solving Newton-like steps at each iteration, whereas numerical experiments tend to show that the number of iterations of primal-dual methods is almost problem-independent and oscillates between 10 and 50.

Corollary 1. *The* box-LMI problem, *which consists in finding the smallest box* $[\mathbf{x}]$ *which encloses a set* $\mathbb{S}$ *defined by an LMI constraint, has a polynomial complexity in the worst-case.*

Proof: Since

$$[\mathbf{x}] = \left[\min_{\mathbf{x}\in\mathbb{S}} x_1, \max_{\mathbf{x}\in\mathbb{S}} x_1\right] \times \cdots \times \left[\min_{\mathbf{x}\in\mathbb{S}} x_m, \max_{\mathbf{x}\in\mathbb{S}} x_m\right], \tag{2.21}$$

computing $[\mathbf{x}]$ amounts to solving $2m$ LMI minimization problems, each of them having a polynomial complexity in the worst-case. ■

3. Lattices and intervals

This section recalls some definitions and properties related to lattices. These notions will be needed in Section 4 to understand links existing between interval matrices and interval symmetric matrices.

A *lattice* $(\mathcal{E}, \leq)$ is a partially ordered set, closed under least upper and greatest lower bounds (see [5], for more details). The least upper bound (or infimum) of x and y is called the *join* and is denoted by $x \vee y$. The greatest lower bound (or *supremum*) is called the *meet* and is written as $x \wedge y$.

Example 5. *The set $\mathbb{R}^n$ is a lattice with respect to the partial order relation given by*

$$\mathbf{x} \leq \mathbf{y} \Leftrightarrow \forall i \in \{1, \dots, n\}, x_i \leq y_i.$$

We have

$$\begin{aligned} \mathbf{x} \wedge \mathbf{y} &= (x_1 \wedge y_1, \dots, x_n \wedge y_n) \text{ and} \\ \mathbf{x} \vee \mathbf{y} &= (x_1 \vee y_1, \dots, x_n \vee y_n) \end{aligned}$$

where $x_i \wedge y_i = \min(x_i, y_i)$ and $x_i \vee y_i = \max(x_i, y_i)$.

A (possibly empty) subset $\mathcal{D}$ of $\mathcal{E}$ is a *sublattice* of $\mathcal{E}$ if

$$\forall x \in \mathcal{D}, \forall y \in \mathcal{D}, x \vee y \in \mathcal{D} \text{ and } x \wedge y \in \mathcal{D}.$$

Example 6. *The set*

$$\mathcal{D}_1 \triangleq \left\{(x_1, x_2) \in \mathbb{R}^2 | x_1 - x_2 \leq 0\right\}$$

is a sublattice of $\mathbb{R}^2$ whereas the set

$$\mathcal{D}_2 \triangleq \left\{(x_1, x_2) \in \mathbb{R}^2 | x_1 + x_2 \leq 0\right\}$$

is not a sublattice of $\mathbb{R}^2$ (for instance, $(-1, 1)$ and $(1, -1)$, both belong to $\mathcal{D}_2$, whereas $(-1, 1) \vee (1, -1) = (1, 1)$ is not an element of $\mathcal{D}_2$).

A lattice $\mathcal{E}$ is *complete* if for all (finite of infinite) subset $\mathcal{A}$ of $\mathcal{E}$, the least upper bound (denoted $\wedge\mathcal{A}$) and the greatest lower bound (denoted $\vee\mathcal{A}$) belong to $\mathcal{A}$. When a lattice $\mathcal{E}$ is not complete, it is possible to add new elements (corresponding the supremum or infimum of infinite subsets of $\mathcal{E}$ that do not belong to $\mathcal{E}$) to make it complete. The completed lattice will be denoted $\overline{\mathcal{E}}$. By convention, for the emptyset, we set $\wedge\emptyset = \vee\mathcal{E}$ and $\vee\emptyset = \wedge\mathcal{E}$.

Example 7. *The set $\mathbb{R}$ is not a complete sublattice whereas $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$ is.*

A *closed interval* (or *interval* for short) $[x]$ of a lattice $\mathcal{E}$ is a subset of $\mathcal{E}$ which satisfies

$$[x] = \{x \in \mathcal{E} \mid \wedge [x] \leq x \leq \vee[x]\}.$$

Note that here, $\wedge[x]$ and $\vee[x]$ belong to $\overline{\mathcal{E}}$, but may not belong to $\mathcal{E}$. Both $\emptyset$ and $\mathcal{E}$ are intervals of $\mathcal{E}$. The set of all intervals of $\mathcal{E}$ will be denoted by $\mathcal{IE}$. An interval is a sublattice of $\mathcal{E}$. An interval $[x]$ of $\mathcal{E}$ will also be denoted by

$$[x] = [\wedge[x], \vee[x]]_{\mathcal{E}}.$$

Example 8. *The sets* $\emptyset = [\infty, -\infty]_{\mathbb{R}}$; $\mathbb{R} = [-\infty, \infty]_{\mathbb{R}}$; $[0,1]_{\mathbb{R}}$ *and* $[0,\infty]_{\mathbb{R}}$ *are intervals of* $\mathbb{R}$. *The set* $\{2,3,4,5\} = [2,5]_{\mathbb{N}}$ *is an interval of the set of integers* $\mathbb{N}$. *The set* $\{4,6,8,10\} = [4,10]_{2\mathbb{N}}$ *is an interval of* $2\mathbb{N}$, *the set of even integers.*

The *interval hull* (or *hull*, for short) of a subset $\mathcal{A}$ of $\mathcal{E}$ is the smallest interval of $\mathcal{E}$ which contains $\mathcal{A}$, *i.e.*,

$$\text{hull}_{\mathcal{E}}(\mathcal{A}) \triangleq \bigcap \{[\mathbf{A}] \in \mathcal{IE} \mid \mathcal{A} \subset [\mathbf{A}]\} = [\wedge\mathcal{A}, \vee\mathcal{A}]_{\mathcal{E}}.$$

Theorem 3.1. *If* $\mathcal{D}$ *is a sublattice of the lattice* $\mathcal{E}$, *if* $\mathcal{D}_1$ *is a subset of* $\mathcal{D}$, *then*

$$\begin{array}{ll} (i) & hull_{\mathcal{D}}(\mathcal{D}_1) = hull_{\mathcal{E}}(\mathcal{D}_1) \cap \mathcal{D}, \\ (ii) & hull_{\mathcal{E}}(hull_{\mathcal{D}}(\mathcal{D}_1)) = hull_{\mathcal{E}}(\mathcal{D}_1). \end{array} \tag{3.1}$$

Proof

Proof of (i): The set $\text{hull}_{\mathcal{E}}(\mathcal{D}_1) \cap \mathcal{D}$ is equal to $[\wedge\mathcal{D}_1, \vee\mathcal{D}_1]_{\mathcal{E}} \cap \mathcal{D}$ which can be rewritten as $\{x \in \mathcal{E} \mid \wedge \mathcal{D}_1 \leq x \leq \vee\mathcal{D}_1\} \cap \mathcal{D}$. Since $\mathcal{D} \subset \mathcal{E}$, this set is equal to $\{x \in \mathcal{D} \mid \wedge \mathcal{D}_1 \leq x \leq \vee\mathcal{D}_1\}$ or equivalently to $\text{hull}_{\mathcal{D}}(\mathcal{D}_1)$.

Proof of (ii): We have $\text{hull}_{\mathcal{E}}(\text{hull}_{\mathcal{D}}(\mathcal{D}_1)) = \text{hull}_{\mathcal{E}}([\wedge\mathcal{D}_1, \vee\mathcal{D}_1]_{\mathcal{D}})$ which is equal to $[\wedge([\wedge\mathcal{D}_1, \vee\mathcal{D}_1]_{\mathcal{D}}), \vee([\wedge\mathcal{D}_1, \vee\mathcal{D}_1]_{\mathcal{D}})]_{\mathcal{E}}$. Now, $\wedge([\wedge\mathcal{D}_1, \vee\mathcal{D}_1]_{\mathcal{D}}) = \wedge\mathcal{D}_1$ and $\vee([\wedge\mathcal{D}_1, \vee\mathcal{D}_1]_{\mathcal{D}}) = \vee\mathcal{D}_1$. Thus $\text{hull}_{\mathcal{E}}(\text{hull}_{\mathcal{D}}(\mathcal{D}_1)) = [\wedge\mathcal{D}_1, \vee\mathcal{D}_1]_{\mathcal{E}} = \text{hull}_{\mathcal{E}}(\mathcal{D}_1)$. ■

Example 9. *Take*

$$\mathcal{D} \triangleq \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 - x_2 = 0\}. \tag{3.2}$$

and $\mathcal{D}_1 = \{(1,1), (3,3)\}$. *The set* $hull_{\mathcal{D}}(\mathcal{D}_1)$, *is an interval of* $\mathcal{D}$, *but not an interval of* $\mathbb{R}^2$ *(see Figure 1 for an illustration). On this figure, it is clear that both equations of (3.1) are satisfied.*

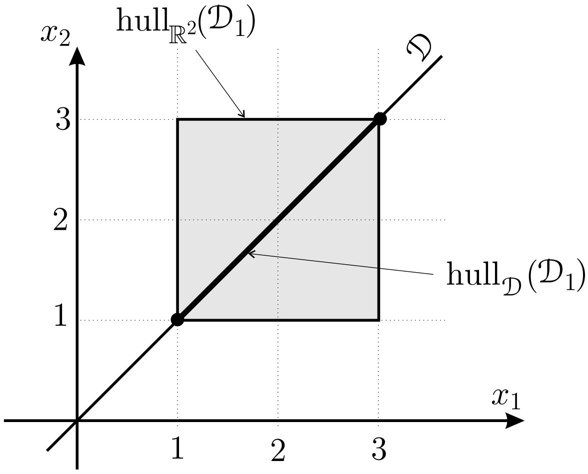

Figure 1. An interval of the line $\mathcal{D}$ is not necessarily an interval of $\mathbb{R}^2$

4. Interval matrices and interval symmetric matrices

In this section, we present some definitions related to interval matrices. Some of them are slightly different from that of the literature [14] but the adaptation is needed to establish some properties used by our algorithm presented in Section 5.

The set of interval matrices $\mathcal{M}_n$ is a lattice and that the set of symmetric matrices $\mathcal{S}_n$ is a sublattice of $\mathcal{M}_n$ with respect to the partial relation order

$$\mathbf{A} \leq \mathbf{B} \Leftrightarrow \forall (i,j) \in \{1,\dots,n\}^2, a_{ij} \leq b_{ij}. \tag{4.1}$$

An *interval matrix* $[\mathbf{A}]$ is an interval of $\mathcal{M}^n$. It can be written indifferently as

$$[\mathbf{A}] = \left\{ \mathbf{A} \in \mathcal{M}^n | \sum_{i,j\in\{1,\dots,n\}} a_{ij}\mathbf{E}^{ij}, a_{ij} \in [a_{ij}] \right\} = \sum_{i,j\in\{1,\dots,n\}} [a_{ij}]\,\mathbf{E}^{ij}, \tag{4.2}$$

$$= \left[\begin{pmatrix} a_{11}^- & \dots & a_{1n}^- \\ \vdots & & \vdots \\ a_{n1}^- & \dots & a_{nn}^- \end{pmatrix}, \begin{pmatrix} a_{11}^+ & \dots & a_{1n}^+ \\ \vdots & & \vdots \\ a_{n1}^+ & \dots & a_{nn}^+ \end{pmatrix} \right]_{\mathcal{M}^n}. \tag{4.3}$$

where $[a_{ij}]$ are n^2 intervals of $\mathbb{R}$. The set of all interval matrices will be denoted by $\mathcal{IM}^n$.

An *interval symmetric matrix* $[\mathbf{B}]$ is an interval of $\mathcal{S}^n$. It can be written indifferently as

$$[\mathbf{B}] = \left\{ \mathbf{B} \in \mathcal{M}^n | \sum_{j \geq i} b_{ij} \mathbf{E}_{\mathrm{S}}^{ij}, b_{ij} \in [b_{ij}] \right\} = \sum_{j \geq i} [b_{ij}] \; \mathbf{E}_{\mathrm{S}}^{ij} \tag{4.4}$$

$$= \left[\begin{pmatrix} b_{11}^- & \dots & b_{1n}^- \\ \vdots & & \vdots \\ b_{1n}^- & \dots & b_{nn}^- \end{pmatrix}, \begin{pmatrix} b_{11}^+ & \dots & b_{1n}^+ \\ \vdots & & \vdots \\ b_{1n}^+ & \dots & b_{nn}^+ \end{pmatrix} \right]_{\mathcal{S}^n}, \tag{4.5}$$

where $[b_{ij}]$ are $\frac{n(n+1)}{2}$ intervals of $\mathbb{R}$. The set of all interval symmetric matrices will be denoted by $\mathcal{IS}^n$. Recall that $[\mathbf{B}]$ is a subset of $\mathcal{S}^n$ and contains only symmetric matrices.

Example 10. *We have*

$$\begin{pmatrix} 2 & 3 \\ 1 & 2 \end{pmatrix} \in \left[\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 3 & 3 \\ 3 & 3 \end{pmatrix} \right]_{\mathcal{M}^n} \tag{4.6}$$

whereas

$$\begin{pmatrix} 2 & 3 \\ 1 & 2 \end{pmatrix} \notin \left[\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 3 & 3 \\ 3 & 3 \end{pmatrix} \right]_{\mathcal{S}^n}. \tag{4.7}$$

Figure 2 gives a graphical illustration of some properties of $\mathcal{IS}^n$ and $\mathcal{IM}^n$. For instance,

- the set of symmetric matrices $\mathcal{S}^n$ is a sub-vector space of $\mathcal{M}^n$,
- the set of PSD matrices $\mathcal{S}_+^n$ is a convex cone of $\mathcal{S}^n$,
- if $[\mathbf{A}]$ is an interval matrix, $[\mathbf{B}] = [\mathbf{A}] \cap \mathcal{S}^n$ is an interval symmetric matrix,
- an interval symmetric matrix $[\mathbf{B}]$ is not necessarily an interval matrix,
- the intersection of an interval symmetric matrix with $\mathcal{S}_+^n$ is not necessarily an interval symmetric matrix.

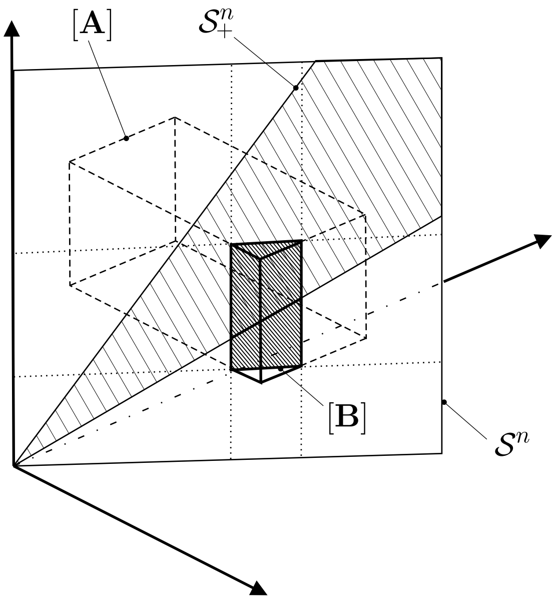

Figure 2. Representation of the set of matrices $\mathcal{M}^n$,
the set of symmetric matrices $\mathcal{S}^n$ and
the set of positive semi-definite matrices $\mathcal{S}_+^n$.
$\mathcal{S}^n$ is a vector space of $\mathcal{M}^n$ and $\mathcal{S}_+^n$ is a convex cone of $\mathcal{S}^n$.
If $[\mathbf{A}]$ is an interval matrix, $[\mathbf{B}] = [\mathbf{A}] \cap \mathcal{S}^n$ is an interval symmetric matrix .
The three axis correspond to the canonical matrices $\mathbf{E}_{ij}$ of $\mathcal{M}^n$.

Recall that the problem to be solved in this paper is to compute the interval matrix

$$\text{hull}_{\mathcal{M}_n}\left([\mathbf{A}]\cap\mathcal{S}_+^n\right), \tag{4.8}$$

for a given $[\mathbf{A}] \in \mathcal{IM}^n$. To reach our goal, we now give a theorem, which provides some important links between interval matrices and interval symmetric matrices.

Theorem .3. *If* $[\mathbf{B}] \in \mathcal{IS}^n$, $\mathcal{B} \subset \mathcal{S}^n$, $[\mathbf{A}] \in \mathcal{IM}^n$, *then*

$$\begin{array}{ll} (i) & hull_{\mathcal{S}^n}(\mathcal{B}) = hull_{\mathcal{M}^n}(\mathcal{B}) \cap \mathcal{S}^n, \\ (ii) & hull_{\mathcal{M}^n}(hull_{\mathcal{S}^n}(\mathcal{B})) = hull_{\mathcal{M}^n}(\mathcal{B}), \\ (iii) & [\mathbf{A}] \cap \mathcal{S}^n = hull_{\mathcal{S}^n}([\mathbf{A}] \cap \mathcal{S}^n) = \sum_{j\geq i}([a_{ij}] \cap [a_{ji}])\,\mathbf{E}_S^{ij}, \\ (iv) & hull_{\mathcal{M}_n}([\mathbf{A}]\cap\mathcal{S}_n^+) = hull_{\mathcal{M}_n}(hull_{\mathcal{S}_n}(([\mathbf{A}]\cap\mathcal{S}_n)\cap\mathcal{S}_n^+)), \\ (v) & [\mathbf{B}]\cap\mathcal{S}_n^+ \text{ is an LMI set of } \mathcal{S}^n, \\ (vi) & hull_{\mathcal{M}^n}([\mathbf{B}]) = \sum_{j\geq i}[b_{ij}]\mathbf{E}^{ij} + \sum_{j<i}[b_{ji}]\mathbf{E}^{ij}. \end{array} \tag{4.9}$$

Proof: Since $\mathcal{S}^n$ is a sublattice of $\mathcal{M}^n$, (i) and (ii) follow directly from Theorem 3.1. We shall thus only prove properties (iii), (iv), (v) and (vi).

Proof of (iii): The set $[\mathbf{A}] \cap \mathcal{S}^n$ is defined by

$$\begin{aligned}
& \left\{\mathbf{A} \in \mathcal{M}^n | \sum_{i,j} a_{ij}\mathbf{E}^{ij}, a_{ij} \in [a_{ij}], a_{ij} = a_{ji}\right\} \\
= & \left\{\mathbf{A} \in \mathcal{M}^n | \sum_{j>i} \left(a_{ij}\mathbf{E}^{ij} + a_{ji}\mathbf{E}^{ji}\right) + \sum_{i} a_{ii}\mathbf{E}^{ii}, a_{ij} \in [a_{ij}], a_{ij} = a_{ji}\right\} \\
= & \left\{\mathbf{A} \in \mathcal{M}^n | \sum_{j>i} a_{ij} \left(\mathbf{E}^{ij} + \mathbf{E}^{ji}\right) + \sum_{i} a_{ii}\mathbf{E}^{ii}, a_{ij} \in [a_{ij}] \cap [a_{ji}]\right\} \\
= & \left\{\mathbf{A} \in \mathcal{M}^n | \sum_{j\geq i} a_{ij}\mathbf{E}_{\mathrm{S}}^{ij}, a_{ij} \in [a_{ij}] \cap [a_{ji}]\right\} \\
= & \sum_{j\geq i} \left([a_{ij}] \cap [a_{ij}]\right) \mathbf{E}_{\mathrm{S}}^{ij},
\end{aligned}$$

which is an interval symmetric matrix. Thus $[\mathbf{A}] \cap \mathcal{S}^n = \mathrm{hull}_{\mathcal{S}^n}\left([\mathbf{A}] \cap \mathcal{S}^n\right)$.

Proof of (iv): Since $\mathcal{S}_n^+ \subset \mathcal{S}_n$,

$$\mathrm{hull}_{\mathcal{M}_n}\left([\mathbf{A}]\cap\mathcal{S}_n^+\right) = \mathrm{hull}_{\mathcal{M}_n}\left(\left([\mathbf{A}]\cap\mathcal{S}_n\right)\cap\mathcal{S}_n^+\right). \tag{4.10}$$

From (ii), we get (iv).

Proof of (v) : We have

$$[\mathbf{B}] \cap \mathcal{S}_n^+ = \left\{\mathbf{B} \in \mathcal{S}_n | \sum_{j\geq i} b_{ij}\mathbf{E}_{\mathrm{S}}^{ij} \succeq 0, b_{ij} \in [b_{ij}]\right\}. \tag{4.11}$$

Now, the constraint $b_{ij} \in [b_{ij}]$ which should be satisfied for all (i, j) such that $j \geq i$ is an LMI (see Example 2) and the constraint $\sum_{j\geq i} b_{ij}\mathbf{E}_{\mathrm{S}}^{ij} \succeq 0$ is also an LMI. Thus $[\mathbf{B}] \cap \mathcal{S}_n^+$ is the intersection of two LMI sets. From Theorem .2 it is thus an LMI set.

Proof of (vi): We have,

$$[\mathbf{B}] = \sum_{j\geq i}[b_{ij}]\mathbf{E}_{\mathrm{S}}^{ij} = \sum_{j>i}[b_{ij}]\left(\mathbf{E}^{ij} + \mathbf{E}^{ji}\right) + \sum_{i=j}[b_{ij}]\mathbf{E}^{ij}. \tag{4.12}$$

Now, from the subdistributivity property, we have the inclusion,

$$\mathcal{IS}^n \ni [b_{ij}]\left(\mathbf{E}^{ij} + \mathbf{E}^{ji}\right) \subset [b_{ij}]\mathbf{E}^{ij} + [b_{ij}]\mathbf{E}^{ji} \in \mathcal{IM}^n. \tag{4.13}$$

Thus, $[\mathbf{B}]$ is a subset of the interval matrix

$$\sum_{j>i}\left([b_{ij}]\mathbf{E}^{ij}+[b_{ij}]\mathbf{E}^{ji}\right)+\sum_{i=j}[b_{ij}]\mathbf{E}^{ij} \tag{4.14}$$

$$=\sum_{j>i}[b_{ij}]\mathbf{E}^{ij}+\sum_{i>j}[b_{ji}]\mathbf{E}^{ij}+\sum_{i=j}[b_{ij}]\mathbf{E}^{ij} \tag{4.15}$$

$$=\sum_{j\geq i}[b_{ij}]\mathbf{E}^{ij}+\sum_{j<i}[b_{ji}]\mathbf{E}^{ij}. \tag{4.16}$$

Let us now show that the interval matrix $[\mathbf{B}_{\mathrm{M}}] \triangleq \sum_{j\geq i}[b_{ij}]\mathbf{E}^{ij}+\sum_{j<i}[b_{ji}]\mathbf{E}^{ij}$ is the smallest which contains $[\mathbf{B}]$. Consider an interval matrix $[\mathbf{B}'_{\mathrm{M}}] \triangleq \sum_{i,j}[b'_{ij}]\mathbf{E}^{ij}$ included in $[\mathbf{B}_{\mathrm{M}}]$. Then, from (iii)

$$[\mathbf{B}'_{\mathrm{M}}]\cap\mathcal{S}^n=\sum_{j\geq i}\left([b'_{ij}]\cap[b'_{ji}]\right)\mathbf{E}_{\mathrm{S}}^{ij}. \tag{4.17}$$

which is a subset of $[\mathbf{B}]$. The inclusion is an equality, if for all $(i,j)\,, j\geq i$, $\left[b'_{ij}\right]\cap[b'_{ji}]=[b_{ij}]$, i.e., $\left[b'_{ij}\right]=[b_{ij}]$ and $[b'_{ji}]=[b_{ij}]$. As a result, $[\mathbf{B}_{\mathrm{M}}]$ is the smallest interval matrix which satisfies $[\mathbf{B}_{\mathrm{M}}]\cap\mathcal{S}^n\supset[\mathbf{B}]$. ∎

5. Projection algorithm for the PSD constraint

This section proposes a polynomial algorithm for projecting the PSD constraint. To our knowledge, no other algorithm can be found in the literature to perform this task.

From (iv) of Theorem .3, we have $\mathrm{hull}_{\mathcal{M}_n}([\mathbf{A}]\cap\mathcal{S}_n^+) = \mathrm{hull}_{\mathcal{M}_n}(\mathrm{hull}_{\mathcal{S}_n}(([\mathbf{A}]\cap\mathcal{S}_n)\cap\mathcal{S}_n^+))$. Thus, the following set algorithm computes $\mathrm{hull}_{\mathcal{M}_n}([\mathbf{A}]\cap\mathcal{S}_n^+)$.

Algorithm PSD(in: $[\mathbf{A}]\in\mathcal{IM}^n$, out: $[\mathbf{D}]\in\mathcal{IM}^n$)
1 $[\mathbf{B}]:=[\mathbf{A}]\cap\mathcal{S}_n$;
2 $[\mathbf{C}]:=\mathrm{hull}_{\mathcal{S}_n}([\mathbf{B}]\cap\mathcal{S}_n^+)$;
3 $[\mathbf{D}]:=\mathrm{hull}_{\mathcal{M}_n}([\mathbf{C}])$;
4 Return $[\mathbf{D}]$.

Step 1 computes $[\mathbf{B}]\in\mathcal{IS}^n$ which is the intersection between $[\mathbf{A}]\in\mathcal{IM}^n$ and $\mathcal{S}_n$. According to Theorem .3 (iii), Step 1 is equivalent to the statement

$$\text{for } i\in\{1,\ldots,n\},\ \text{for } j\in\{i,\ldots,n\},\ [b_{ij}]:=[a_{ij}]\cap[a_{ji}]. \tag{5.1}$$

Step 2 computes the smallest interval symmetric matrix $[\mathbf{C}]$ which encloses all matrices of [**B**] that are PSD. This amounts to solving a box-LMI problem, where the LMI set is

$$\left\{\mathbf{B} \in \mathcal{S}^n \mid \sum_{j \geq i} b_{ij}\mathbf{E}_{\mathrm{S}}^{ij} \succeq 0, b_{ij} \in [b_{ij}]\right\}. \tag{5.2}$$

In our implementation, the $n\,(n+1)$ LMI optimization problems are solved using the SeDuMi solver which implements a primal-dual interior-point algorithm. It has a worst-case complexity

$$O(\left(n^{3.5}m + n^{2.5}m^2\right)\log\left(\frac{1}{\varepsilon}\right)),$$

where m =card($\{b_{ij} \mid j \geq i\}$) $= \frac{n(n+1)}{2}$ and ε is the required accuracy.

Step 3 generates the smallest matrix $[\mathbf{D}] \in \mathcal{IM}^n$ which encloses $[\mathbf{C}] \in \mathcal{IS}^n$. From (vi) of Theorem .3, this can be performed by the following statements

$$\begin{array}{l} \text{for } i \in \{1, \dots, n\}, \\ \quad\quad \text{for } j \in \{1, \dots, i-1\}, \quad [d_{ij}] := [c_{ji}] \\ \quad\quad \text{for } j \in \{i, \dots, n\}, \quad\quad [d_{ij}] := [c_{ij}] \\ \text{endfor } i. \end{array} \tag{5.3}$$

Theorem .4. *The algorithm* PSD *has a worst-case complexity of* $n^{8.5}\log\left(\frac{1}{\varepsilon}\right)$, *where* ε *is the relative required accuracy.*

Proof: PSD needs the resolution of $n\,(n+1)$ LMI optimization problems, each of which performed by SeDuMi which has a worst-case complexity $O\left(\left(n^{3.5}m + n^{2.5}m^2\right)\log\left(\frac{1}{\varepsilon}\right)\right)$. Since the number of variables of each LMI is $m = \frac{n(n+1)}{2}$, the worst-case complexity of PSD is $O\left(n^{8.5}\log\left(\frac{1}{\varepsilon}\right)\right)$. ■

Remark 1. *The algorithm* PSD *can be used to test if the interval matrix* [**A**] *contains at least one PSD matrix. Of course, only the first of* $n\,(n+1)$ *optimization problems at Step 2 has to be solved. Thus, an optimal (no pessimism exists) nonconvexity check can be implemented with a complexity* $O\left(n^{6.5}\right)$.

6. Example

Consider the interval matrix

$$\begin{aligned} [\mathbf{A}] &= \left[\begin{pmatrix} -7 & -1 & -5 \\ -4 & -8 & 2 \\ -4 & -1 & 4 \end{pmatrix}, \begin{pmatrix} 3 & 4 & 4 \\ 2 & 3 & 9 \\ 9 & 6 & 9 \end{pmatrix}\right]_{\mathcal{M}^n} && (6.1) \\ &= [-7,3]_{\mathbb{R}}\,\mathbf{E}^{11} + [-1,4]_{\mathbb{R}}\,\mathbf{E}^{12} + \dots && (6.2) \end{aligned}$$

Step 1 of our algorithm computes the intersection $[\mathbf{A}] \cap \mathcal{S}_n$. It generates the interval symmetric matrix

$$[\mathbf{B}] = \left[\begin{pmatrix} -7 & -1 & -4 \\ -1 & -8 & 2 \\ -4 & 2 & 4 \end{pmatrix}, \begin{pmatrix} 3 & 2 & 4 \\ 2 & 3 & 6 \\ 4 & 6 & 9 \end{pmatrix} \right]_{\mathcal{S}^n} \tag{6.3}$$

$$= [-7, 3]_{\mathbb{R}} \, \mathbf{E}_{\mathrm{S}}^{11} + [-1, 2]_{\mathbb{R}} \, \mathbf{E}_{\mathrm{S}}^{12} + \ldots \tag{6.4}$$

Note that $[\mathbf{B}]$ should not be confused with

$$\mathrm{hull}_{\mathcal{M}^n}([\mathbf{B}]) = \left[\begin{pmatrix} -7 & -1 & -4 \\ -1 & -8 & 2 \\ -4 & 2 & 4 \end{pmatrix}, \begin{pmatrix} 3 & 2 & 4 \\ 2 & 3 & 6 \\ 4 & 6 & 9 \end{pmatrix} \right]_{\mathcal{M}^n}. \tag{6.5}$$

which belongs to $\mathcal{IM}^n$ and not to $\mathcal{IS}^n$. $[\mathbf{B}]$ is a set of symmetric matrices whereas $\mathrm{hull}_{\mathcal{M}^n}([\mathbf{B}])$ contains matrices that are not symmetric.

Step 2 solves $2\frac{n(n+1)}{2} = 12$ LMI problems in $\frac{n(n+1)}{2} = 6$ variables. The first one, which computes the lowest possible value for b_{11} such that $\mathbf{B} \in \mathcal{S}_+^n \cap [\mathbf{B}]$, is given by

$$c_{11}^- = \min b_{11}$$
$$\text{st:} \begin{cases} b_{11} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + b_{12} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \cdots + b_{33} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \succeq 0 \\ b_{11} \in [-7, 3]_{\mathbb{R}}, b_{12} \in [-1, 2]_{\mathbb{R}}, b_{13} \in [-4, 4]_{\mathbb{R}}, \\ b_{22} \in [-8, 3]_{\mathbb{R}}, b_{23} \in [2, 6]_{\mathbb{R}}, b_{33} \in [4, 9]_{\mathbb{R}}. \end{cases} \tag{6.6}$$

After completion of the 12 LMI minimization problems, the resulting interval symmetric matrix reads:

$$[\mathbf{C}] = \left[\begin{pmatrix} 0.0000 & -1.0000 & -4.0000 \\ -1.0000 & 0.4444 & 2.0000 \\ -4.0000 & 2.0000 & 4.0000 \end{pmatrix}, \begin{pmatrix} 3.0000 & 2.0000 & 4.0000 \\ 2.0000 & 3.0000 & 5.1962 \\ 4.0000 & 5.1962 & 9.0000 \end{pmatrix} \right]_{\mathcal{S}^n}. \tag{6.7}$$

This result has been obtained with the LMI solver SeDuMi [18] with the YALMIP [12] Matlab interface in less than 3 seconds on a PC Pentium IV computer.

Step 3 generates $[\mathbf{D}] = \mathrm{hull}_{\mathcal{M}_n}([\mathbf{C}])$. The result obtained is

$$[\mathbf{D}] = \left[\begin{pmatrix} 0.0000 & -1.0000 & -4.0000 \\ -1.0000 & 0.4444 & 2.0000 \\ -4.0000 & 2.0000 & 4.0000 \end{pmatrix}, \begin{pmatrix} 3.0000 & 2.0000 & 4.0000 \\ 2.0000 & 3.0000 & 5.1962 \\ 4.0000 & 5.1962 & 9.0000 \end{pmatrix} \right]_{\mathcal{M}^n}. \tag{6.8}$$

In order to help the reader to solve his own testcases and to show how easy it is to implement the PSD algorithm, we decided to give the corresponding Matlab source code on the following table.

```
B = sdpvar(n,n);
Binf = [-7 -1 -5; -4 -8 2; -4 -1 4];
Bsup = [3 4 4; 2 3 9; 9 6 9];
Binf = max(Binf',Binf); Bsup = min(Bsup',Bsup);
Cinf = zeros(n); Csup = zeros(n);
L = lmi(B>0);
for i = 1:n, for j = 1:i
     L = L+lmi(B(i,j)>Binf(i,j))+lmi(B(i,j)<Bsup(i,j));
end, end
for i = 1:n,for j = 1:i
     sol = solvesdp(L,[],B(i,j));
     Cinf(i,j) = double(B(i,j)); Cinf(j,i) = Cinf(i,j);
     sol = solvesdp(L,[],-B(i,j));
     Csup(i,j) = double(B(i,j)); Csup(j,i) = Csup(i,j);
end,end;
```

In the Matlab code, commands `sdpvar`, `lmi` and `solvesdp` are YALMIP instructions, used to define an LMI variable, an LMI constraint, and solve an LMI problem, respectively. To demonstrate the efficiency of PSD with respect to the dimension n of $[\mathbf{A}]$, let us generate 30 interval matrices

$$[\mathbf{A}_n] = \mathbf{I}_n + [-\Delta_n^-, \Delta_n^+], \; n = 1, \ldots, 30$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix and Δ_n^- and Δ_n^+ are $n \times n$ matrices whose coefficients are integer numbers taken randomly inside the interval $[0, n]$. The logarithm of the computing times $T(n)$ obtained by PSD on a PC Pentium IV are given on Figure 3. Note that for $n = 30$, the computing time is 16996 seconds (*i.e.*, about 4 hours and 43 minutes), thus

$$\alpha_n \triangleq \frac{\log_{10}(T(n))}{\log_{10} n} = \frac{\log_{10}(16996)}{\log_{10} 30} = 2.86.$$

This is consistent with Theorem .4 that claims that $\lim_{n\to\infty} \alpha_n$ is a real number smaller than 8.5.

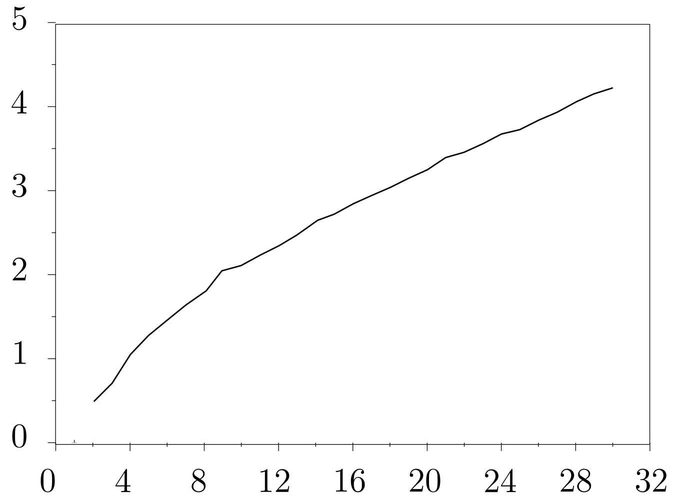

Figure 3. $\log_{10}(T(n))$ with respect to n, where $T(n)$ is the computing time of PSD.

Remark 2. *The computational complexity obtained experimentally is far less than the worst-case complexity namely because sparsity can be exploited in the primal-dual interior-point method when solving for the Newton step at each iteration.*

7. Conclusion

In this paper, we have shown that LMI's can be used to deal with global constraints involving matrices. The approach has been illustrated on the unary constraint PSD (Positive Semi-Definite) for a matrix. An algorithm which computes the smallest interval matrix which contains all PSD matrices that belong to a given interval matrix has been given.

To get validated results (to take into account the finite representation of numbers in the computers), an LMI solver with outward rounding and other validated procedures (for instance, based on the approach proposed in [9]) should be developed. To our knowledge, such a solver does not exist yet.

Global optimization algorithms such as that of Hansen [8] or αBB [1] could take advantage of the contraction algorithm PSD, proposed in this paper. Recall that when it is known that at the global minimum, the Hessian matrix is PSD, Hansen's algorithm or αBB try to test whether or not the interval Hessian matrix, at the current box, may contain any PSD matrix (this is the *non-convexity test*). If it concludes that it cannot, the corresponding box is removed. A nonconvexity contractor based on the algorithm PSD could be used to contract the current box, pruning parts of the box where the Hessian cannot be PSD.

This would make it possible to continue the propagation process before bisection (which should always be considered as a last resort).

We believe that adding our nonconvexity contractor to existing interval global optimization algorithms could considerably increase their efficiency, especially when the number of variables is high. This point, which is beyond the scope of this paper, remains to be studied.

References

[1] C. Adjiman, S. Dallwig, C. Floudas, and A. Neumaier. A global optimization method, αBB, for general twice-differentiable constrained NLPs - I theoretical advances. *Computers and Chemical Engineering*, 22:1137–1158, 1998.

[2] A. Ben-Tal and A. Nemirovskii. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications.* SIAM, Philadelphia, PA, 2001.

[3] C. Bliek, P. Spellucci, L. Vincente, A. Neumaier, L. Granvilliers, E. Huens, P. V. Hentenryck, D. Sam-Haroud, and B. Faltings. *Algorithms for Solving Nonlinear Constrained and Optimisation Problems: State of the Art.* A 222 page progress report of the COCONUT project, 2001. Available at `http://www.mat.univie.ac.at/~neum/glopt/coconut/cocbib.html`.

[4] S. Boyd and L. Vanbenberghe. *Convex optimization.* Lecture notes at Stanford University and University of California at Los Angeles, 2003. Available at `http://www.stanford.edu/~boyd`.

[5] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order.* Cambridge University Press, (ISBN 0521784514), 2002.

[6] P. Gahinet and A. Nemirovskii. *LMI Lab: a package for manipulating and solving LMIs.* INRIA Rocquencourt, France, 1993.

[7] P. Gahinet, A. Nemirovskii, A. J. Laub, and M. Chilali. *The LMI Control Toolbox for use with Matlab.* The MathWorks, Inc., Natick, MA, 1995.

[8] E. R. Hansen. *Global Optimization using Interval Analysis.* Marcel Dekker, New York, NY, 1992.

[9] C. Jansson. A rigorous lower bound for the optimal value of convex optimization problems. *To appear in journal of global optimization*, 2003.

[10] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London, 2001.

[11] M. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[12] J. Löfberg. *YALMIP, Yet another LMI parser.* University of Linköping, Sweden, 2001. Available at `http://www.control.isy.liu.se/~johanl`.

[13] Y. Nesterov and A. Nemirovskii. *Interior-point polynomial methods in convex programming.* SIAM, Philadelphia, PA, 1994.

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

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

[16] D. Sam-Haroud. *Constraint consistency techniques for continuous domains.* PhD dissertation 1423, Swiss Federal Institute of Technology in Lausanne, Switzerland, 1995.

[17] C. Scherer and S. Weiland. *Course on LMIs in Control.* Lecture Notes at Delft University of Technology and Eindhoven University of Technology, 2002. Available at `http://www.cs.ele.tue.nl/SWeiland/lmi.htm`.

[18] J. F. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999. Available at `http://fewcal.kub.nl/sturm`.

[19] P. van Hentenryck, L. Michel, and Y. Deville. *Numerica - A Modelling Language for Global Optimization.* MIT Press, Cambridge, Massachusetts, 1997.

[20] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[21] M. VanEmden. Algorithmic power from declarative use of redundant constraints. *Constraints*, 4(4):363–381, 1999.