Linear matrix inequalities for interval constraint propagation

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 solved in polynomial time (more precisely with a worst-case complexity of $n^{8.5}$). In a constraint viewpoint, this problem corresponds to projecting the global constraint 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) [9], [15]. A CSP [12] is composed by a set of variables $\mathcal{V} = \{x_1, \ldots, x_n\}$, a set of constraints $\mathcal{C} = \{c_1, \ldots, c_m\}$ and a set of interval domains $\{[x_1], \ldots, [x_n]\}$. The aim of propagation techniques is to contract as much as possible the domains for the variables without loosing any solution [3], [4], [18], [20]. 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 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 $c_{j1}, c_{j_2},\ldots$ on which the square intersection $\sqcap$ can be computed. When such a decomposition is performed, the fixed point that is reached is generally much bigger than the one that obtained without the decomposition.

Extending the class of constraints for which the square intersection $\sqcap$ can be computed efficiently is an important task that should be considered in the constraint community.

In this paper, we consider the constraint *positive semi-definite* 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 $[\mathbf{A}]$. This constraint often occurs in control theory (see e.g., [16], [14]) or in optimization (see *e.g.* the non-convexity check in [8]). The approach to be proposed is based on linear matrix inequalities (LMI) briefly presented in Section 2. Some important notions and properties of interval (symmetric) matrices are given in Section 3. Section 4 provides a polynomial algorithm that solves our problem. An illustrative example is given in Section 5.

2. Linear matrix inequalities

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 $\left\{\mathbf{E}^{ij}_{\mathrm{S}}\right\}_{j\geq i}$, where

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

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

Example 2.1. *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 2.2. *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 2.3. *The set of PSD matrices* $\mathcal{S}_+^n \triangleq \{\mathbf{A} \in \mathcal{S}^n | \mathbf{A} \succeq 0\}$ *is a convex cone 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 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 2.4. *A linear matrix inequality (LMI) has the form*

$$\mathbf{A}(\mathbf{x}) \triangleq \mathbf{A_0} + x_1 \mathbf{A}_1 + \ldots + x_m \mathbf{A}_m \succeq 0, \tag{2.8}$$

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

Proof: The LMI set $\mathbb{S} \triangleq \{\mathbf{x} \in \mathbb{R}^m | \mathbf{A_0} + x_1 \mathbf{A}_1 + \ldots + x_m \mathbf{A}_m \succeq 0\}$ can be written as $\mathbb{S} = \mathbf{f}^{-1}\left(\mathcal{S}_+^n\right)$,where

$$\mathbf{f} : \begin{cases} \mathbb{R}^m & \rightarrow \quad \mathcal{M}^n \\ \mathbf{x} & \mapsto \quad \mathbf{A_0} + x_1 \mathbf{A}_1 + \ldots + x_m \mathbf{A}_m \end{cases} \tag{2.9}$$

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

$$\mathbb{S}_a = \{\mathbf{x} \in \mathbb{R}^m \mid \mathbf{A_0} + x_1\mathbf{A}_1 + \ldots + x_m\mathbf{A}_m \succeq 0\} \tag{2.10}$$

$$\mathbb{S}_b = \{\mathbf{x} \in \mathbb{R}^m \mid \mathbf{B_0} + x_1\mathbf{B}_1 + \ldots + x_m\mathbf{B}_m \succeq 0\} \tag{2.11}$$

is given by

$$\begin{aligned} & \left\{\mathbf{x} \in \mathbb{R}^m \mid \begin{pmatrix} \mathbf{A_0} + x_1\mathbf{A}_1 + \ldots + x_m\mathbf{A}_m & \mathbf{0} \\ \mathbf{0} & \mathbf{B_0} + x_1\mathbf{B}_1 + \ldots + x_m\mathbf{B}_m \end{pmatrix} \succeq 0\right\} \\ = & \left\{\mathbf{x} \in \mathbb{R}^m \mid \begin{pmatrix} \mathbf{A_0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B_0} \end{pmatrix} + x_1 \begin{pmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_1 \end{pmatrix} + \ldots + x_m \begin{pmatrix} \mathbf{A}_m & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_m \end{pmatrix} \succeq 0\right\} \end{aligned}$$

which is an LMI set. ■

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

Example 2.6. *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 2.7. *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

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

i.e.,

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

Example 2.8. *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], [5]) which claims that, for any matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ with appropriated dimensions, the following equivalence:*

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

is always true. 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 ellipsoid defined by $3x_1^2 + 2x_2^2 - 2x_1x_2 < 5$. We have

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

Using the Schur complement theorem with

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

we get the LMI

$$\begin{pmatrix} 1 & x_1 & x_2 \\ x_1 & 2 & 1 \\ x_2 & 1 & 3 \end{pmatrix} \succ 0,$$

i.e.,

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

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 [5], [2] collect an impressive amount of LMI-representable geometric sets (*e.g.,* ellipses, parabolae or disks) or more general convex sets relevant to control engineering, structural optimization or combinatorial optimization.

Following the seminal work of Karmarkar [10] 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 [19] for a survey and [17] 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 2.9. *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

$$[\mathbf{x}] = \left[\min_{\mathbf{x}\in\mathbb{S}} x_1, \max_{\mathbf{x}\in\mathbb{S}} x_1\right] \times \ldots \times \left[\min_{\mathbf{x}\in\mathbb{S}} x_m, \max_{\mathbf{x}\in\mathbb{S}} x_m\right], \tag{2.14}$$

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

3. Interval matrices and interval symmetric matrices

An interval of $\mathbb{R}$ is a closed connected set of $\mathbb{R}$. An *interval matrix* $[\mathbf{A}]$ is a set of matrices in $\mathcal{M}^n$ which can be written as

$$[\mathbf{A}] = \left\{ \mathbf{A} \in \mathcal{M}^n \middle| \sum_{i,j\in\{1,\ldots,n\}} a_{ij}\mathbf{E}^{ij}, a_{ij} \in [a_{ij}] \right\} = \sum_{i,j\in\{1,\ldots,n\}} [a_{ij}].\mathbf{E}^{ij}, \tag{3.1}$$

where $[a_{ij}]$ are n^2 intervals. The set of all interval matrices will be denoted by $\mathcal{IM}^n$. An *interval symmetric matrix* $[\mathbf{B}]$ is a subset of $\mathcal{M}^n$ which can be written as

$$[\mathbf{B}] = \left\{ \mathbf{B} \in \mathcal{M}^n | \sum_{j \geq i} b_{ij} \mathbf{E}_{\mathrm{S}}^{ij}, b_{ij} \in [b_{ij}] \right\} = \sum_{j \geq i} [b_{ij}].\mathbf{E}_{\mathrm{S}}^{ij}, \tag{3.2}$$

where $[b_{ij}]$ are $\frac{n(n+1)}{2}$ intervals. The set of all interval symmetric matrices will be denoted by $\mathcal{IS}^n$.

If $\mathcal{A}$ is a subset of $\mathcal{M}^n$ and $\mathcal{B}$ is a subset of $\mathcal{S}^n$, then we define the two operators

$$\mathrm{hull}_{\mathcal{M}_n}(\mathcal{A}) \triangleq \bigcap \{ [\mathbf{A}] \in \mathcal{IM}^n \mid \mathcal{A} \subset [\mathbf{A}] \}, \tag{3.3}$$

$$\mathrm{hull}_{\mathcal{S}_n}(\mathcal{B}) \triangleq \bigcap \{ [\mathbf{B}] \in \mathcal{IS}^n \mid \mathcal{B} \subset [\mathbf{B}] \}. \tag{3.4}$$

With these notations, the problem to be solved in this paper is to compute the interval matrix

$$\mathrm{hull}_{\mathcal{M}_n} \left([\mathbf{A}] \cap \mathcal{S}_+^n \right), \tag{3.5}$$

for a given $[\mathbf{A}] \in \mathcal{IM}^n$.

Figure 3.1 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.

Theorem 3.1. *If* $[\mathbf{B}] \in \mathcal{IS}^n$, $\mathcal{B} \subset \mathcal{S}^n$, $[\mathbf{A}] \in \mathcal{IM}^n$, *then*

$$\begin{array}{ll}
(i) & hull_{\mathcal{S}^n}(\mathcal{B}) = hull_{\mathcal{M}^n}(\mathcal{B}) \cap \mathcal{S}^n, \\
(ii) & hull_{\mathcal{M}^n} \circ hull_{\mathcal{S}^n}(\mathcal{B}) = hull_{\mathcal{M}^n}(\mathcal{B}), \\
(iii) & [\mathbf{A}] \cap \mathcal{S}^n = hull_{\mathcal{S}^n}([\mathbf{A}] \cap \mathcal{S}^n) = \sum_{j \geq i} ([a_{ij}] \cap [a_{ji}]) \mathbf{E}_S^{ij}, \\
(iv) & hull_{\mathcal{M}_n}([\mathbf{A}] \cap \mathcal{S}_n^+) = hull_{\mathcal{M}_n}(hull_{\mathcal{S}_n}(([\mathbf{A}] \cap \mathcal{S}_n) \cap \mathcal{S}_n^+)), \\
(v) & [\mathbf{B}] \cap \mathcal{S}_n^+ \text{ is an LMI set of } \mathcal{S}^n, \\
(vi) & hull_{\mathcal{M}^n}([\mathbf{B}]) = \sum_{j \geq i} [b_{ij}] \mathbf{B}^{ij} + \sum_{j < i} [b_{ji}] \mathbf{E}^{ij}.
\end{array} \tag{3.6}$$

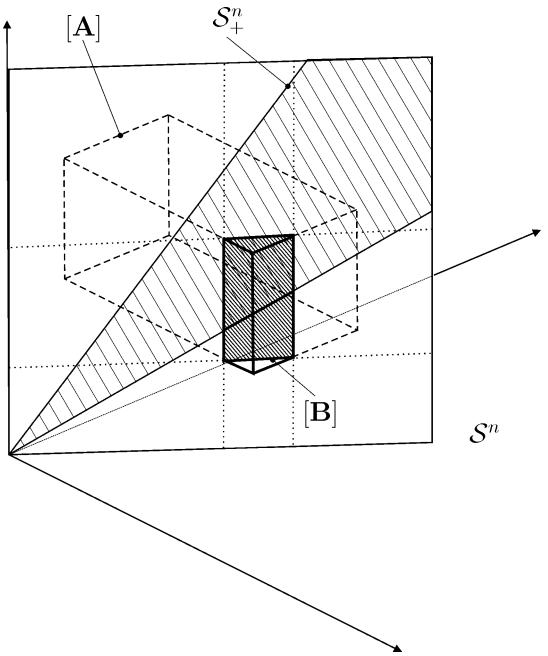

Figure 3.1: In the set of $n \times n$ matrices $\mathcal{M}^n$, the set of symmetric matrices $\mathcal{S}^n$ is a vector space, the set of positive semi-definite matrices $\mathcal{S}_+^n$ is a convex cone of $\mathcal{S}^n$. If $[\mathbf{A}]$ is an interval matrix, $[\mathbf{B}] = [\mathbf{A}] \cap \mathcal{S}^n$ is an interval symmetric matrix

Proof: Here, we shall only prove properties (iii), (iv), (v) and (vi) that are useful to understand the algorithm.

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 $[\mathbf{A}] \cap \mathcal{S}^n = \mathrm{hull}_{\mathcal{S}^n} ([\mathbf{A}] \cap \mathcal{S}^n)$.

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

$$\text{hull}_{\mathcal{M}_n}\left([\mathbf{A}] \cap \mathcal{S}_n^+\right) = \text{hull}_{\mathcal{M}_n}\left(([\mathbf{A}] \cap \mathcal{S}_n) \cap \mathcal{S}_n^+\right). \tag{3.7}$$

From (ii), we get (iv). ■

Proof of (v) : We have

$$[\mathbf{B}] \cap \mathcal{S}_n^+ = \left\{ \mathbf{B} \in \mathcal{S}_n | \sum_{j \geq i} b_{ij} \mathbf{E}_{\mathrm{S}}^{ij} \succeq 0, b_{ij} \in [b_{ij}] \right\}. \tag{3.8}$$

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 2.6) 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 2.5 it is thus an LMI set. ■

Proof of (vi): We have,

$$[\mathbf{B}] = \sum_{j \geq i} [b_{ij}] \mathbf{E}_{\mathrm{S}}^{ij} = \sum_{j > i} [b_{ij}] \left(\mathbf{E}^{ij} + \mathbf{E}^{ji}\right) + \sum_{i=j} [b_{ij}] \mathbf{E}^{ij}. \tag{3.9}$$

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

$$\mathcal{IS}^n \ni [b_{ij}] \left(\mathbf{E}^{ij} + \mathbf{E}^{ji}\right) \subset [b_{ij}] \mathbf{E}^{ij} + [b_{ij}] \mathbf{E}^{ji} \in \mathcal{IM}^n. \tag{3.10}$$

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

$$\sum_{j > i} \left([b_{ij}] \mathbf{E}^{ij} + [b_{ij}] \mathbf{E}^{ji}\right) + \sum_{i=j} [b_{ij}] \mathbf{E}^{ij} \tag{3.11}$$

$$= \sum_{j > i} [b_{ij}] \mathbf{E}^{ij} + \sum_{i > j} [b_{ji}] \mathbf{E}^{ij} + \sum_{i=j} [b_{ij}] \mathbf{E}^{ij} \tag{3.12}$$

$$= \sum_{j \geq i} [b_{ij}] \mathbf{E}^{ij} + \sum_{j < i} [b_{ji}] \mathbf{E}^{ij}. \tag{3.13}$$

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

$$[\mathbf{B}'_{\mathrm{M}}] \cap \mathcal{S}^n = \sum_{j \geq i} \left([b'_{ij}] \cap [b'_{ji}]\right) \mathbf{E}_{\mathrm{S}}^{ij}. \tag{3.14}$$

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 $[\mathbf{B}_{\mathrm{M}}] \cap \mathcal{S}^n \supset [\mathbf{B}]$.

4. Projection algorithm for the PSD constraint

From (iv) of Theorem 3.1, we have $\text{hull}_{\mathcal{M}_n}([\mathbf{A}]\cap\mathcal{S}_n^+) = \text{hull}_{\mathcal{M}_n}(\text{hull}_{\mathcal{S}_n}(([\mathbf{A}]\cap\mathcal{S}_n)\cap\mathcal{S}_n^+))$. Thus, the following set algorithm computes $\text{hull}_{\mathcal{M}_n}([\mathbf{A}]\cap\mathcal{S}_n^+)$.

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 $[\mathbf{C}] := \text{hull}_{\mathcal{S}_n}([\mathbf{B}]\cap\mathcal{S}_n^+)$;
3 $[\mathbf{D}] := \text{hull}_{\mathcal{M}_n}([\mathbf{C}])$;
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 3.1 (iii), Step 1 is equivalent to the statement

$$\text{for } i\in\{1,\ldots,n\},\ \text{for } j\in\{i,\ldots,n\},\ [b_{ij}] := [a_{ij}]\cap[a_{ji}]. \tag{4.1}$$

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

$$\left\{\mathbf{B}\in\mathcal{S}^n \mid \sum_{j\geq i} b_{ij}\mathbf{E}_{\mathrm{S}}^{ij} \succeq 0, b_{ij}\in[b_{ij}]\right\}. \tag{4.2}$$

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 $O((n^{3.5}m+n^{2.5}m^2)\log\left(\frac{1}{\varepsilon}\right))$ where $m = \text{card}(\{b_{ij}\mid j\geq i\}) = \frac{n(n+1)}{2}$.

Step 3 generates the smallest matrix $[\mathbf{D}]\in\mathcal{IM}^n$ which encloses $[\mathbf{C}]\in\mathcal{IS}^n$. From (vi) of Theorem 3.1, this can be performed by the following statements

$$\begin{array}{l}\text{for } i\in\{1,\ldots,n\},\\ \quad\text{for } j\in\{1,\ldots,i-1\}, \quad [d_{ij}] := [c_{ji}]\\ \quad\text{for } j\in\{i,\ldots,n\}, \quad [d_{ij}] := [c_{ij}]\\ \text{endfor } i.\end{array} \tag{4.3}$$

Theorem 4.1. *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. This task is performed by SeDuMi which has a worst-case complexity $O\left((n^{3.5}m+n^{2.5}m^2)\log\left(\frac{1}{\varepsilon}\right)\right)$. 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* $[\mathbf{A}]$ *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* $O(n^{6.5})$.

5. Examples

Consider the interval matrix

$$[\mathbf{A}] = \begin{pmatrix} [-7,3] & [-1,4] & [-5,4] \\ [-4,2] & [-8,3] & [2,9] \\ [-4,9] & [-1,6] & [4,9] \end{pmatrix} = [-7,3].\mathbf{E}^{11} + [-1,4].\mathbf{E}^{12} + \dots \tag{5.1}$$

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

$$[\mathbf{B}] = \begin{pmatrix} [-7,3] & [-1,2] & [-4,4] \\ \square & [-8,3] & [2,6] \\ \square & \square & [4,9] \end{pmatrix} = [-7,3].\mathbf{E}_{\mathrm{S}}^{11} + [-1,2].\mathbf{E}_{\mathrm{S}}^{12} + \dots \tag{5.2}$$

The square symbol $\square$ is used here to recall that $[\mathbf{B}]$ belongs to $\mathcal{IS}^n$ and not to $\mathcal{IM}^n$. Note that $[\mathbf{B}]$ should not be confused with

$$\mathrm{hull}_{\mathcal{M}^n}([\mathbf{B}]) = \begin{pmatrix} [-7,3] & [-1,2] & [-4,4] \\ [-1,2] & [-8,3] & [2,6] \\ [-4,4] & [2,6] & [4,9] \end{pmatrix}. \tag{5.3}$$

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

$$c_{11}^- = \min b_{11} \quad \text{st:} \quad \left\{ \begin{array}{c} b_{11} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + b_{12} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \dots + b_{33} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \succeq 0 \\ b_{11} \in [-7,3], b_{12} \in [-1,2], b_{13} \in [-4,4], \\ b_{22} \in [-8,3], b_{23} \in [2,6], b_{33} \in [4,9] \end{array} \right. \tag{5.4}$$

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

$$[\mathbf{C}] = \begin{pmatrix} [0.0000, 3.0000] & [-1.0000, 2.0000] & [-4.0000, 4.0000] \\ \square & [0.4444, 3.0000] & [2.0000, 5.1962] \\ \square & \square & [4.0000, 9.0000] \end{pmatrix}. \tag{5.5}$$

This result has been obtained with the LMI solver SeDuMi [17] with the YALMIP [11] Matlab interface in less than 3 seconds on a PC Pentium IV computer.

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

$$[\mathbf{D}] = \begin{pmatrix} [0.0000, 3.0000] & [-1.0000, 2.0000] & [-4.0000, 4.0000] \\ [-1.0000, 2.0000] & [0.4444, 3.0000] & [2.0000, 5.1962] \\ [-4.0000, 4.0000] & [2.0000, 5.1962] & [4.0000, 9.0000] \end{pmatrix}. \tag{5.6}$$

The corresponding source code is given below.

```
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 order to show the efficiency of PSD with respect to the dimension n of $[\mathbf{A}]$, let us generate $n_{\max}$ interval matrices

$$[\mathbf{A}_n] = \mathbf{I}_n + [-\Delta_n^-, \Delta_n^+]$$

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 taken randomly inside the interval $[0, n]$. The logarithm of the computing times

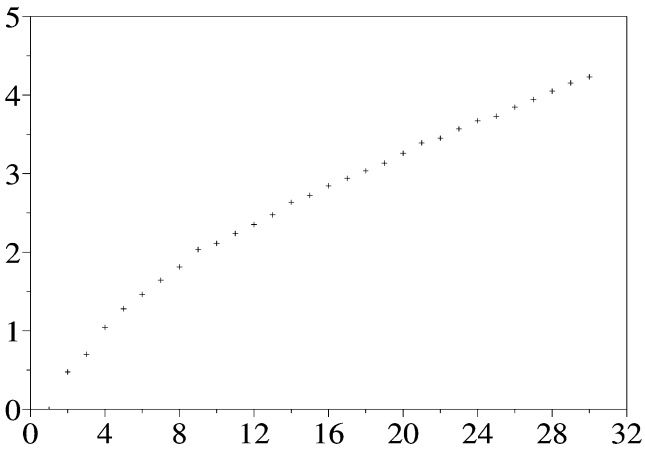

Figure 5.1: $\log_{10}(T(n))$ with respect to n, where $T(n)$ is the computing time of PSD

$T(n)$ obtained by PSD on a PC Pentium IV are given on Figure 5.1. Note that for $n = 30$

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

This is consistent with Theorem 4.1 that claims that $\lim_{n\to\infty} \alpha_n$ is a real number smaller than 8.5.

6. 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.

This algorithm can be used in control theory where the constraint PSD naturally appears in the problem. But is can also be used by global optimization algorithms such as that of Hansen [8] or αBB [1] to build a nonconvexity contractor. 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 can be 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).

Also, 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 should be developed. To our knowledge, such a solver does not exist yet.

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] F. Benhamou and W. Older. Applying interval arithmetic to real, integer and Boolean constraints. *Journal of Logic Programming*, pages 1–24, 1997.

[4] 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`.

[5] 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`.

[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] 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.

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

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

[12] A. K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8(1):99–118, 1977.

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

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

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

[16] 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`.

[17] 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`.

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

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

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