

An Algorithm for Determining the Least Minimum Singular Value of a Polytope of Matrices

Yash Shrivastava

Center for Industrial Control Science
 Newcastle, NSW 2308, Australia
 yash@ee.newcastle.edu.au

Minyue Fu

Dept. of ECE Univ. of Newcastle
 NSW 2308, Australia
 eemf@ee.newcastle.edu.au

Abstract

Given m square matrices A_1, \dots, A_m , let \mathcal{A} denote the set of all their convex combinations. Then we consider the problem of determining a member of \mathcal{A} whose minimum singular value is the smallest. A related problem, known as robust nonsingularity problem, is to determine if every member of \mathcal{A} is nonsingular. Clearly a solution to our problem automatically solves the robust nonsingularity problem. Unfortunately, the robust nonsingularity problem has been demonstrated to be NP-hard which in turn makes our problem NP-hard. To avoid this computational intractability, we provide an algorithm that computes a lower bound and an upper bound on the least minimum singular value within a prescribed tolerance. Of course, if the prescribed tolerance is set to zero then our algorithm would compute the least minimum singular value. Our method makes use of the so-called simplicial algorithms.

1. Introduction

In this paper we will be concerned with the least minimum singular value problem. Before defining the problem precisely, we have a few words regarding notation. Throughout this paper \mathbb{R}, \mathbb{R}^m and $\mathbb{R}^{n \times n}$ (resp. \mathbb{C}, \mathbb{C}^m and $\mathbb{C}^{n \times n}$) will denote the set of real (resp. complex) numbers, m -vectors and $n \times n$ matrices respectively. A^H stands for complex conjugate transpose of a complex matrix, vector or number A , $\sigma_{\min}(\cdot)$ denotes the minimum singular value of a matrix and $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of a Hermitian matrix.

It is well known, [1], that for a real or complex nonsingular matrix A , $\sigma_{\min}(A)$ measures the distance of A from the set of singular matrices, i.e.

$$\sigma_{\min}(A) = \min \{ \|E\| : A + E \text{ is a singular matrix} \} \quad (1.1)$$

where $\|\cdot\|$ denotes the spectral norm of a matrix. Hence if the minimum singular value of a matrix is large, then it can tolerate relatively large perturbations in its elements before becoming singular. On the other hand, if the minimum singular value of a matrix is small, then numerical problems occur in various computations (e.g. inversion) involving the matrix.

Both the least minimum singular value problem and the robust nonsingularity problem, consider a polytope of matrices. Recall that a polytope of matrices is represented

as

$$\begin{aligned} \mathcal{A} &= \text{convex hull of } \{A_1, \dots, A_m\} \\ &= \left\{ A(\mu) = \sum_{i=1}^m \mu_i A_i : \mu \in \Gamma \right\} \end{aligned} \quad (1.2)$$

where $A_i \in \mathbb{C}^{n \times n}$, $i = 1, \dots, m$ and

$$\Gamma = \left\{ \mu : \mu \in \mathbb{R}^m, \mu_i \geq 0, i = 1, \dots, m, \sum_{i=1}^m \mu_i = 1 \right\}. \quad (1.3)$$

The set Γ in (1.3) is known as an $(m-1)$ dimensional unit simplex. In rest of the paper a family \mathcal{A} , as in (1.2), will be denoted by $co(A_1, \dots, A_m)$. The polytope matrix family \mathcal{A} is said to be *robustly nonsingular* if all its members are nonsingular. The concept of minimum singular value can be extended in a natural way for a polytopic family. We define the *least minimum singular value (LMSV)* of a polytope matrix family \mathcal{A} as follows:

$$LMSV(\mathcal{A}) = \min_{A \in \mathcal{A}} \sigma_{\min}(A). \quad (1.4)$$

The importance of LMSV stems from (1.1) which gives the following:

$$LMSV(\mathcal{A}) = \min \{ \|E\| : A + E \text{ is a singular matrix for some } A \in \mathcal{A} \} \quad (1.5)$$

i.e. $LMSV(\cdot)$ of a polytope matrix family measures its distance from the set of singular matrices. Note that \mathcal{A} is robustly nonsingular iff $LMSV(\mathcal{A}) > 0$. The primary objective of this paper is to develop an algorithm that calculates $LMSV(\mathcal{A})$ for any given polytope matrix family \mathcal{A} .

The computation of LMSV is related to some well known robust stability problems. The first one is the robust stability of a polytope matrix family \mathcal{A} in (1.2). The family \mathcal{A} is said to be *robustly stable* if all its members are stable (i.e. have eigenvalues in the open left half complex plane). A number of authors, [2]-[6], have observed that the robust stability problem can in fact be transformed into an equivalent robust nonsingularity problem. The common theme behind these transformations is to create an associated polytope matrix family such that original family is robustly stable if and only if the associated family is robustly nonsingular. Several such transformations have been formulated, e.g. power transformations [2, 3, 4], bialternate product [4, 5] and Kronecker sum [5, 6].

The second robust stability problem related to the computation of LMSV is a generalization of the first one, namely, we consider the following family of matrices:

$$\mathcal{A}_\delta = \left\{ E + \sum_{i=1}^m \mu_i A_i : \mu \in \Gamma, E \in \mathbb{C}^{n \times n}, \|E\| \leq \delta \right\} \quad (1.6)$$

where Γ is given by (1.3), μ represents some parametric uncertainty, E is some unstructured uncertainty which is norm bounded by $\delta > 0$. The robust stability problem is to determine if every member of the matrix family in (1.6) is Hurwitz stable. From the well-known *zero exclusion principle* (see for example [7]), the robust stability of \mathcal{A}_δ is equivalent to the following:

- (i) \mathcal{A}_δ has at least one member which is Hurwitz stable;
- (ii) $\min_{A \in \mathcal{A}} \sigma_{\min}(j\omega I - A) > \delta$ for all ω .

That is, the robust stability problem boils down to computing the LMSV. Besides being directly connected to a robust stability problem, robust nonsingularity problem is also useful in sensitivity analysis of linear systems [8].

Note that determining $LMSV(\mathcal{A})$ is NP-hard. This is because even a special case of robust nonsingularity problem has been demonstrated to be NP-hard [9]. To avoid this computational intractability, we take the following approach. We compute a lower bound and an upper bound for $LMSV(\mathcal{A})$. If these bounds are unsatisfactory, a branch and bound technique [10, 11] is used to systematically improve the bounds. The improvement can be done till the bounds are within a prescribed tolerance. The idea behind this is that if the upper bound calculated turns out to be very small then, even if the set is robustly nonsingular, numerical problems may occur in dealing with this set. Of course, if the prescribed tolerance is zero then our algorithm will compute the least minimum singular value, but will require an exponential number of steps. Our method makes use of the so-called simplicial algorithms [12].

The organization of the paper is as follows. Section 2 derives the lower bound. Section 3 discusses simplicial algorithms and the computation of upper bound. Section 4 gives the branch and bound technique to improve the bounds. Section 5 presents some examples and Section 6 concludes. Due to space constraints, most of the proofs have been omitted. The reader should contact the authors for proofs and additional discussion.

2. Lower Bound

In this section we derive a lower bound on LMSV of polytope matrix family. The following key lemma plays an important role in this derivation.

Lemma 2.1 *Let $B \in \mathbb{C}^{n \times n}$ be a Hermitian matrix and λ_k be an eigenvalue of B with the corresponding unit eigenvector $x_k \in \mathbb{C}^n$. Then perturbations in λ_k relate to perturbations in B in the following fashion.*

$$\delta \lambda_k = x_k^H (\delta B) x_k + \text{higher order terms} \quad (2.1)$$

Since the minimum singular value of a matrix is always nonnegative and $\sigma_{\min}^2(A) = \lambda_{\min}(A^H A)$, it easily follows that

$$(LMSV(\mathcal{A}))^2 = \min_{A \in \mathcal{A}} \lambda_{\min}(A^H A). \quad (2.2)$$

Further, for smooth variations of parameters μ in Γ , $\lambda_{\min}(A^H(\mu)A(\mu))$ varies continuously and is also differentiable except when there are multiple eigenvalues corresponding to the $\lambda_{\min}(\cdot)$. However, note that minimum of $\lambda_{\min}(A^H(\mu)A(\mu))$ over \mathcal{A} is not achieved at any nondifferentiable interior point¹. This fact can be roughly argued as follows: Suppose the minimum of $\lambda_{\min}(A^H(\mu)A(\mu))$ is achieved at a nondifferentiable interior point $\mu^* \in \Gamma$. Then, there exists a line $L \subset \Gamma$ passing through μ^* such that the line derivative of $\lambda_{\min}(A^H(\mu)A(\mu))$ on Γ does not exist at μ^* . This nondifferentiability is caused by the crossing of two continuously differentiable eigenvalues of $A^T(\mu)A(\mu)$. That is, one eigenvalue achieves the minimum on one side of L , the other eigenvalue does so on the other side, and they meet at μ^* . By the continuous differentiability of these two eigenvalues, there exists $\hat{\mu} \in L$ in an arbitrarily small neighborhood of μ^* such that

$$\lambda_{\min}(A^H(\hat{\mu})A(\hat{\mu})) < \lambda_{\min}(A^H(\mu^*)A(\mu^*))$$

which contradicts the initial assumption that the minimum of $\lambda_{\min}(A^H(\mu)A(\mu))$ is achieved at μ^* .

From the foregoing discussion it is clear that if the minimum of $\lambda_{\min}(A^H(\mu)A(\mu))$ is achieved at an interior point of \mathcal{A} , then the first order necessary conditions must be satisfied at that point. Towards this end we have the following lemma.

Lemma 2.2 *For a polytope matrix family \mathcal{A} , as in (1.2) and (1.3), the first order necessary conditions at an interior point of \mathcal{A} for the minimum of $\lambda_{\min}(A^H(\mu)A(\mu))$ are given by*

$$\begin{aligned} x_{\min}^H(\mu) [A^H(\mu)(A_j - A_k) + (A_j^H - A_k^H)A(\mu)] x_{\min}(\mu) \\ = 0 \quad \forall j \neq k, j, k = 1, \dots, m \end{aligned} \quad (2.3)$$

where $x_{\min}(\mu) \in \mathbb{C}^n$ is the unit eigenvector associated with the $\lambda_{\min}(A^H(\mu)A(\mu))$.

As a first step towards deriving a lower bound LMSV of polytope matrix family, we have the following theorem.

Theorem 2.1 *Suppose for a polytope matrix family \mathcal{A} , as in (1.2) and (1.3), the minimum of $\lambda_{\min}(A^H(\mu)A(\mu))$ is achieved at an interior point of Γ . Then*

$$(LMSV(\mathcal{A}))^2 \geq \max_{C \in \mathcal{A}} \min_j \lambda_{\min} \left(\frac{A_j^H C + C^H A_j}{2} \right). \quad (2.4)$$

Now, to give a general lower bound LMSV of polytope matrix family, we need a word on notation. For a polytope matrix family \mathcal{A} , as in (1.2) and (1.3), \mathcal{A}_i^k will denote the k -dimensional boundaries of \mathcal{A} (i.e. $(k+1)$ out of m coefficients μ_j are nonzero on these boundaries and rest

¹A point $\mu \in \Gamma$ is said to be an interior point if $\mu \notin \partial\Gamma$, i.e., not on the boundary of Γ .

are zero). Note that \mathcal{A}_i^0 , \mathcal{A}_i^1 and \mathcal{A}_i^{m-2} are respectively vertices, edges and facets of \mathcal{A} ; \mathcal{A}_i^{m-1} is \mathcal{A} itself. Finally $V(\mathcal{A}_i^k)$ denotes the vertices of \mathcal{A}_i^k . This brings us to the promised characterization of a lower bound.

Theorem 2.2 For a polytope matrix family \mathcal{A} , as in (1.2) and (1.3), the following holds.

$$(LMSV(\mathcal{A}))^2 \geq \min_{i,k} \max_{C \in \mathcal{A}_i^k} \min_{A_j \in V(\mathcal{A}_i^k)} \lambda_{\min} \left(\frac{A_j^H C + C^H A_j}{2} \right). \quad (2.5)$$

Proof: Follows simply by repeated applications of Theorem 2.1 and by noticing that boundaries, \mathcal{A}_i^k , themselves are polytopes of a smaller dimension and A_j are the vertices of the polytope \mathcal{A}_i^{m-1} . ■

Though Theorem 2.2 gives a lower bound on LMSV, it is hard to implement since the maximization over $C \in \mathcal{A}_i^k$ is in general not a convex problem. The following theorem simplifies the characterization of the lower bound in Theorem 2.2, and indeed it is this characterization we use for our lower bound calculations.

Theorem 2.3 For a polytope matrix family \mathcal{A} , as in (1.2) and (1.3), the following holds.

$$(LMSV(\mathcal{A}))^2 \geq \max \left(\min_{i,j \in \{1, \dots, m\}} \Lambda_{ij}, 0 \right) \quad (2.6)$$

where we have defined a symmetric matrix $\Lambda \in \mathbb{R}^{m \times m}$ having the ij th element

$$\Lambda_{ij} = \lambda_{\min} \left(\frac{A_j^H A_i + A_i^H A_j}{2} \right). \quad (2.7)$$

Remark 2.1 Note that diagonal elements Λ_{ii} of Λ in (2.7) are actually equal to $(\sigma_{\min}(A_i))^2$ and thus are trivially an upper bound for $(LMSV(\mathcal{A}))^2$. Hence if the least element of Λ is a diagonal element Λ_{i^*,i^*} , then $(LMSV(\mathcal{A}))^2 = \Lambda_{i^*,i^*}$.

Remark 2.2 As the size of polytope \mathcal{A} shrinks, i.e. $\|A_i - A_j\| \rightarrow 0$, elements Λ_{ij} of Λ in (2.7) approach Λ_{ii} . Consequently it follows from Remark 2.1 that the lower bound of Theorem 2.3 and thus also of Theorem 2.2 asymptotically approach $(LMSV(\mathcal{A}))^2$ as the size of \mathcal{A} shrinks to zero.

3. Upper bound

In this section we derive an upper bound on LMSV of polytope matrix family. Note that $\sigma_{\min}(A)$ for any $A \in \mathcal{A}$ is trivially an upper bound on $LMSV(\mathcal{A})$. However, choosing an $A \in \mathcal{A}$ blindly for upper bound calculations would lead to a conservative upper bound. Hence we apply *simplicial algorithms* to calculate an approximate local minimum of function $\lambda_{\min}(A(\mu)^H A(\mu))$ over parameters $\mu \in \Gamma$ and use it for upper bound calculations. Simplicial

algorithms will be described later in this section. But first we need a few preliminary results to make our problem amenable to application of simplicial algorithms.

It is well known, [1], that any matrix $A \in \mathbb{C}^{n \times n}$ can be written as $\sum_{i=1}^n \sigma_i u_i v_i^H$, where σ_i are real numbers such that $\sigma_1 \geq \dots \geq \sigma_n \geq 0$ and u_1, \dots, u_n (v_1, \dots, v_n) are n -dimensional orthonormal vectors, i.e. $u_i^H u_i = 1$ ($v_i^H v_i = 1$) $\forall i$, and $u_i^H u_j = 0$ ($v_i^H v_j = 0$) $\forall i \neq j$. Such a decomposition of A is known as a singular value decomposition (SVD). Few easy consequences of this decomposition are that $\lambda_{\min}(A^H A) = (\sigma_n)^2$ with associated unit eigenvector $x_{\min} = v_n$; and $Av_n = \sigma_n u_n$. Then, with $\sum_{i=1}^n \sigma_i(\mu) u_i(\mu) v_i(\mu)^H$ as SVD of a matrix $A(\mu)$ belonging to polytope matrix family \mathcal{A} of (1.2) and (1.3), we have the following equivalent characterization of first order necessary conditions of Lemma 2.2.

Lemma 3.1 For a polytope matrix family \mathcal{A} , as in (1.2) and (1.3), the first order necessary conditions at an interior point of \mathcal{A} for the minimum of $\lambda_{\min}(A(\mu)^H A(\mu))$ are given by

$$2(\sigma_n(\mu))^2 - \sigma_n(\mu)[u_n(\mu)^H A_i v_n(\mu) + v_n(\mu)^H A_i^H u_n(\mu)] = 0, \text{ for } i = 1, \dots, m. \quad (3.1)$$

For a polytope matrix family \mathcal{A} , as in (1.2) and (1.3), we associate a function $f : \Gamma \rightarrow \mathbb{R}^m$ as follows. The i th component of f is given by

$$f_i(\mu) = 2(\sigma_n(\mu))^2 - \sigma_n(\mu)[u_n^H(\mu) A_i v_n(\mu) + v_n^H(\mu) A_i^H u_n(\mu)].$$

Then from Lemma 3.1 it is clear that a zero point of function f (i.e. μ^* such that $f(\mu^*) = 0$) in the interior of Γ corresponds to a local optimum of $\lambda_{\min}(A(\mu)^H A(\mu))$ in \mathcal{A} . Actually more can be said about the properties of function f . But before presenting the properties of f , we need the following definition.

Definition 3.1 A function $f : \Gamma \rightarrow \mathbb{R}^m$ is said to be a complementary function on Γ , if it satisfies $\mu^T f(\mu) = 0$ for all $\mu \in \Gamma$. Further, a point $\mu^* \in \Gamma$ is said to be a complementary point of such a function f if $f_i(\mu^*) = 0$ whenever $\mu_i^* > 0$ and $f_i(\mu^*) \leq 0$ whenever $\mu_i^* = 0$.

Theorem 3.1 Function f associated with the polytope matrix family \mathcal{A} of (1.2) and (1.3) is indeed a complementary function on Γ . Further, a point $\mu^* \in \Gamma$ corresponds to a local minimum of $\lambda_{\min}(A(\mu)^H A(\mu))$ only if μ^* is a complementary point of f .

Remark 3.1 Suppose μ^* be a complementary point of f and Z be the set of all those indices j for which $\mu_j^* = 0$. Further if $f_i(\mu^*) < 0$ for an $i \in Z$, then μ^* can not correspond to a local maximum of $\lambda_{\min}(A^H(\mu) A(\mu))$.

From Theorem 3.1 and Remark 3.1 it is clear that a good way to compute an upper bound on $LMSV(\mathcal{A})$ is to search for complementary points of function f associated with \mathcal{A} . Also note that calculation of $f(\mu)$ gives

$\sigma_{\min}(A(\mu))$ as a by-product. We apply simplicial algorithms to get approximate complementary points of function f .

A simplicial algorithm operates on an underlying triangulation of the domain Γ . Triangulation of simplex Γ partitions it into many smaller simplices in an orderly fashion. Various types of triangulations are known. The particular triangulation we employ is known as the *V-triangulation*. V-triangulation of a simplex is completely determined by two parameters: a starting point v and the grid size g^{-1} . The size of resultant simplices gets smaller as the grid size decreases. An example of V-triangulation for a 2-dimensional unit simplex with grid size $g^{-1} = 1/2$ is given in Figure 3.1. Here e_1, e_2 and e_3 are the points represented by the parameter vector μ^T taking values $[1,0,0]$, $[0,1,0]$ and $[0,0,1]$ respectively. A detailed description of V-triangulation can be found in [12, pages 51–54].

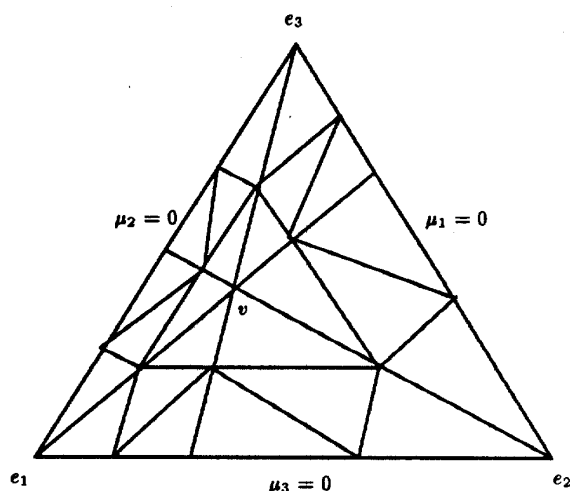


Figure 3.1: V-triangulation with grid size $g^{-1} = 1/2$

Just like the underlying triangulations, various types of simplicial algorithms are known in the literature. Of these algorithms, the one known as vector labeling algorithm on V-triangulation has been found to be good for finding complementary points of certain functions that appear in Economics. It is this algorithm which we employ for upper bound calculations. Starting from the initial point of underlying V-triangulation, the algorithm evaluates function f at certain vertices of simplices obtained by triangulation and follows a piecewise linear path till it finds an approximate complementary point of f . The algorithm is a variable grid refinement type, i.e. if the approximate complementary point found is not satisfactory, then one can restart the algorithm with this approximate solution as the starting point and a smaller grid size to obtain a better solution. Ultimately as the grid size shrinks to zero, the algorithm converges to a complementary point of f , provided f is a continuous function. Note that for a continuous complementary function f , the existence of a complementary point is guaranteed from an easy modification of Brouwer's fixed point theorem (see [12, pages 22–

23]). A detailed description of vector labeling algorithm on V-triangulation can be found in [12, pages 84–93].

Few remarks on the use of vector labeling algorithm on V-triangulation for upper bound calculations are in order. Since the algorithm calculates f at various points in Γ and thus also provides value of $\sigma_{\min}(A(\mu))$ at those points, an upper bound can be taken as the minimum of $\sigma_{\min}(A(\mu))$ among these points. Indeed it is this upper bound that we use in our algorithm, to be presented in the next section. Secondly, as refining just the upper bound without refining the lower bound is not of much value, we need not restart the algorithm with finer grid size to find more accurate complementary points. In fact, we use $1/2$ as the grid size of the underlying triangulation. We conclude this section with the following remark.

Remark 3.2 Since the upper bound calculated is actually minimum singular value of a matrix in the polytope, it asymptotically approaches $LMSV(\mathcal{A})$ as the size of the polytope \mathcal{A} shrinks to zero (see Remark 2.2 for definition).

4. Algorithm for Improving the Bounds

In this section, we discuss an algorithm to improve bounds on $LMSV(\mathcal{A})$. The algorithm is based on the branch and bound techniques of [10, 11]. The algorithm computes $LMSV(\mathcal{A})$ asymptotically. The algorithm partitions \mathcal{A} into various matrix polytopes of the same form, i.e. a polytope \mathcal{B} belonging to the partition is of the form $co(B_1, \dots, B_m)$ where $B_i \in \mathcal{A}$. This partitioning corresponds to an equivalent partitioning of $(m-1)$ -dimensional unit simplex Γ associated with \mathcal{A} into various $(m-1)$ -dimensional simplices of smaller size.

The algorithm computes $LMSV(\mathcal{A})$ to within an absolute accuracy of $\epsilon > 0$, using two functions $LB(\mathcal{B})$ and $UB(\mathcal{B})$ defined over polytopes \mathcal{B} in the partition. These two functions satisfy the following conditions.

(C1) $LB(\mathcal{B}) \leq LMSV(\mathcal{B}) \leq UB(\mathcal{B})$, i.e. the functions LB and UB compute a lower bound and an upper bound on $LMSV(\mathcal{B})$ respectively.

(C2) As the size of \mathcal{B} goes to zero, the difference between upper and lower bounds uniformly converges to zero, i.e.

$$\forall \epsilon > 0, \exists \delta > 0 \text{ s.t. } \|B_i - B_j\| \leq \delta \forall i, j \Rightarrow UB(\mathcal{B}) - LB(\mathcal{B}) \leq \epsilon.$$

Roughly speaking, the bounds LB and UB become sharp as the polytope shrinks to a point.

The algorithm starts by computing $LB(\mathcal{A})$ and $UB(\mathcal{A})$. If $UB(\mathcal{A}) - LB(\mathcal{A}) \leq \epsilon$, the algorithm terminates. Otherwise \mathcal{A} is partitioned into sub-polytopes $\mathcal{B}_1, \dots, \mathcal{B}_p$ and $LB(\mathcal{B}_i)$ and $UB(\mathcal{B}_i)$ are computed. Then

$$\min_{1 \leq i \leq p} LB(\mathcal{B}_i) \leq LMSV(\mathcal{A}) \leq \min_{1 \leq i \leq p} UB(\mathcal{B}_i),$$

and we have new bounds on $LMSV(\mathcal{A})$. If the difference between these new bounds is less than or equal to ϵ , then the algorithm terminates. Otherwise, partitioning of \mathcal{A} is refined and new bounds are calculated. Once the size of the polytopes constituting the partitioning gets smaller than δ then by condition (C2) above it follows that

$$\min_{1 \leq i \leq p} UB(\mathcal{B}_i) - \min_{1 \leq i \leq p} LB(\mathcal{B}_i)$$

$$\leq \max_{1 \leq i \leq p} (UB(\mathcal{B}_i) - LB(\mathcal{B}_i)) \leq \epsilon,$$

and $LMSV(\mathcal{A})$ is determined within an absolute accuracy of ϵ . However, for this “ δ -grid”, the number of polytopes forming the partition (and thus the number of lower and upper bound calculations) increases exponentially with $1/\delta$. To avoid this combinatorial explosion a heuristic rule is applied for partitioning \mathcal{A} : Given a partition $\mathcal{A} = \bigcup_{i=1}^p \mathcal{B}_i$ that is to be refined, pick a polytope \mathcal{B}_i from the partition such that $LB(\mathcal{B}_i) = \min_{1 \leq i \leq p} LB(\mathcal{B}_i)$, and split it into two halves. The rationale behind this heuristic is that since we are trying to find the minimum of a function, we should concentrate on the “most promising” polytope. Further the splitting of a polytope \mathcal{B} is carried out in the following fashion. Suppose the least element of Λ (see Section 2) associated with \mathcal{B} is Λ_{ij} , i.e. $LB(\mathcal{B}) = \sqrt{\Lambda_{ij}}$ if $\Lambda_{ij} \geq 0$ and zero otherwise. Clearly, if $i = j$ then this is also the upper bound (see Remark 2.1) and the algorithm can be terminated, i.e. there is no need to partition \mathcal{B} . Hence without loss of generality assume $i \neq j$ and $\mathcal{B} = \text{co}(B_1, \dots, B_m)$ is to be split into two subpolytopes. Then the subpolytopes \mathcal{B}_I and \mathcal{B}_{II} are given as

$$\begin{aligned} \mathcal{B}_I &= \text{co}(D_1, \dots, D_m) \\ \mathcal{B}_{II} &= \text{co}(F_1, \dots, F_m) \end{aligned}$$

where

$$D_k = \begin{cases} B_k & \text{if } k \neq i \\ G & \text{otherwise} \end{cases}, \quad F_k = \begin{cases} B_k & \text{if } k \neq j \\ G & \text{otherwise} \end{cases}$$

$$\begin{aligned} \text{and } G &= \left(\frac{\Lambda_{jj} - \Lambda_{ji}}{(\Lambda_{ii} - \Lambda_{ij}) + (\Lambda_{jj} - \Lambda_{ji})} \right) B_i \\ &+ \left(\frac{\Lambda_{ii} - \Lambda_{ij}}{(\Lambda_{ii} - \Lambda_{ij}) + (\Lambda_{jj} - \Lambda_{ji})} \right) B_j. \end{aligned}$$

An example of the splitting is shown in Figure 4.1, where a polytope $\mathcal{B} = \text{co}(B_1, B_2, B_3)$ is split along edge (B_1, B_2) , i.e. $i = 1$ and $j = 2$.

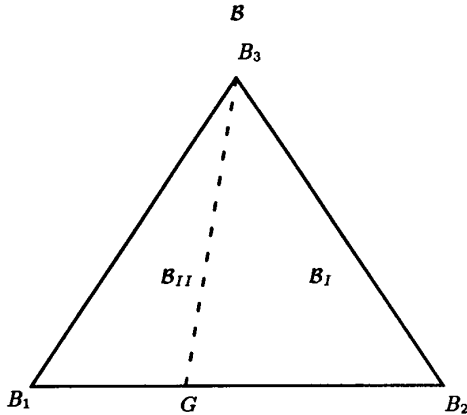


Figure 4.1: Splitting of polytope \mathcal{B} along edge (B_1, B_2)

The rationale behind this splitting is that after the splitting we should have

$$\lambda_{\min}(G^H G) \leq \lambda_{\min} \left(\frac{B_i^H G + G^H B_i}{2} \right)$$

$$\text{and } \lambda_{\min}(G^H G) \leq \lambda_{\min} \left(\frac{B_j^H G + G^H B_j}{2} \right), \quad (4.1)$$

so that the term in Λ s of the subpolytopes corresponding to G, G is a better candidate than those corresponding to B_i, G and B_j, G for determining their respective LB s. Thus either the lower bound of subpolytopes is trivially an upper bound also or some new edge is split when further splitting of these subpolytopes is required. Solving (4.1) approximately using first order approximations, G found, is indeed as given above. It can be shown that this splitting reduces the size of subpolytopes created in the partitioning uniformly in the long run and thus guarantees that the algorithm terminates in finite number of steps.

Note that after splitting, to calculate the lower bounds of the new polytopes created we need to do just $(m+1)$ eigenvalue calculations, namely those involving the new vertex G with itself and the vertices of the old polytope (rest of the entries in their Λ s are known from previous iteration). Also note that the diagonal elements Λ_{ii} of Λ involved in the lower bound calculation give minimum singular values of the vertices (see Remark 2.1) and can also be used for the upper bound calculations. In fact we use

$$\begin{aligned} LB(\mathcal{A}) &= \text{square root of the lower bound in (2.6)} \\ \text{and } UB(\mathcal{A}) &= \min(\text{upper bound from simplicial} \\ &\quad \text{algorithm}, \sqrt{\min_{1 \leq i \leq m} \Lambda_{ii}}). \end{aligned}$$

With these definitions in mind, the algorithm can be described as follows:

```

k = 0;
L0 = {A};
L0 = LB(A);
U0 = UB(A);
while Uk - Lk > epsilon, {
    pick B in Lk such that LB(B) = Lk;
    split B into BI and BII;
    Lk+1 = (Lk - B) union {BI, BII};
    Lk+1 = min_{B in Lk+1} LB(B);
    Uk+1 = min_{B in Lk+1} UB(B);
    k = k + 1;
}

```

In the above description, k denotes iteration index, \mathcal{L}_k is the list of polytopes and L_k and U_k are respectively the lower bound and the upper bound at the end of k iterations. We conclude this section by mentioning that as iterations proceed, some of the polytopes can be eliminated from consideration as $LMSV(\mathcal{A})$ can not be achieved in them. The polytopes that can be pruned are the ones satisfying $LB(\mathcal{B}) > U_k$. Pruning reduces the computation and storage requirement of the algorithm.

5. Some Examples

In this section we apply the algorithm of the previous section to two examples.

Example 5.1: Consider a polytope $\mathcal{A} = \text{co}(A_1, A_2, A_3)$ given by

$$A_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & -2/3 & -1/2 \\ 0 & 1/2 & -2/3 \end{bmatrix}, A_2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1/3 & 1/2 \\ 0 & -1/2 & 1/3 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1/3 & -1/2 \\ 0 & 1/2 & 1/3 \end{bmatrix}.$$

This polytope has been deliberately constructed so that it contains exactly one singular matrix in the interior. By calculating the determinant of a generic member of the polytope, one can easily verify that the singular matrix corresponds to parameters $\mu_1 = 1/3, \mu_2 = 1/2, \mu_3 = 1/6$. Running our algorithm on this polytope with a tolerance (ϵ) of 0.1, we obtain an upper bound of 8.4574e-04 and a lower bound of 0 in just 5 eigenvalue calculations. Further the coordinates of the point which achieves this upper bound are $\mu_1 = 0.3323, \mu_2 = 0.5010, \mu_3 = 0.1667$ (quite close to the correct solution). No splittings of the polytope were required in this case. Note that as the value of $LMSV(\mathcal{A})$ in this example is exactly zero, the lower bound can not be improved with repeated refinement of partitioning of \mathcal{A} . This example thus illustrates the power of our simplicial algorithm for upper bound calculations. In this case just 5 SVD are needed to calculate an approximate complementary point of function f associated with \mathcal{A} . Secondly, this example highlights a point stressed in Section 1 that even though determining $LMSV(\mathcal{A})$ exactly would require an exponential number of steps, just 5 SVD are enough to tell us that the polytope is nearly singular (as the upper bound obtained is very small).

Example 5.2: Consider a polytope $\mathcal{A} = \text{co}(A_1, A_2, A_3)$ where A_1, A_2, A_3 are respectively

$$\begin{bmatrix} 1.8482 + 0.8278j & 0.6313 + 0.6295j & 1.4314 + 0.2332j \\ 0.1738 + 0.1254j & 1.9958 + 0.7362j & 1.0976 + 0.3063j \\ 1.3301 + 0.0159j & 1.2203 + 0.7254j & 1.1497 + 0.3510j \end{bmatrix},$$

$$\begin{bmatrix} 2.2598 + 0.5717j & 0.5435 + 0.9554j & 0.7183 + 0.8420j \\ 1.2145 + 0.8024j & 2.3122 + 0.7483j & 1.4885 + 0.1598j \\ 1.0714 + 0.0331j & 1.6653 + 0.5546j & 2.3231 + 0.2128j \end{bmatrix},$$

$$\begin{bmatrix} 1.4558 + 0.8870j & 1.4814 + 0.3877j & 1.2096 + 0.5901j \\ 1.0239 + 0.6521j & 2.3811 + 0.4997j & 1.0669 + 0.9554j \\ 1.4121 + 0.1503j & 0.9563 + 0.1475j & 1.7973 + 0.5561j \end{bmatrix}.$$

Running our algorithm on this polytope with a tolerance (ϵ) of 0.1, we obtain an upper bound of 0.2009 and a lower bound of 0.1381 in 53 eigenvalue calculations. Further the coordinates of the point which achieves this upper bound are $\mu_1 = 0.8646, \mu_2 = 0.1354, \mu_3 = 0$. No pruning was performed in this example and three splittings (i.e. three iterations) of the polytope were required. It is clear that the polytope of this example is robustly nonsingular. If one was just interested in checking the robustly nonsingularity of the polytope alone, we can just use the lower bound refinement part of our algorithm, which in this case gives a lower bound of 0.0431 after just 2 splittings and needing 14 eigenvalue calculations.

We conclude this section by noting that: (1) as the dimension n of the matrices in the polytope increases, the computational burden of our algorithm increases in proportion to the increased burden associated with each SVD, and

(2) as the polytope size m increases, the computational burden of our algorithm at each polytope splitting step increases linearly with m .

6. Conclusion

We have given an algorithm that calculates bounds on least minimum singular value of a polytope matrix family. A novel formula for lower bound on LMSV has been derived and a novel use of simplicial algorithm is made for calculating upper bound on LMSV. We conclude this paper by noting that our method can be easily modified to handle matrix families of type (1.2) whose parameter set Γ is a product of unit simplices (known as simplotope) by making use of simplicial algorithms on simplotopes.

References

- [1] G.H. Golub and C.F. Van Loan, *Matrix computations*, Johns Hopkins University Press, Baltimore, 1989.
- [2] R.W. Brockett, Ed., "Lie algebras and lie groups in control theory", *Geometric Methods in Systems Theory*, D.Q. Mayne and R.W. Brockett, Eds. Dordrecht: Reidel, 1973, pp 43-82.
- [3] A.I. Barkin and A.L. Zelentsovsky, "Method of power transformations for analysis of stability of nonlinear control systems", *Syst. Contr. Lett.*, vol. 3, pp 303-310, 1983.
- [4] L. Saydy, A.L. Tits and E.H. Abed, "Robust stability of linear systems relative to guarded domains", *Proc. 27th Conf. Decision Contr.*, Austin, TX, 1988, pp 544-551.
- [5] L.Qiu and E.J. Davison, "A new method for the stability robustness determination of state space models with real perturbations", *Proc. 27th Conf. Decision Contr.*, Austin, TX, 1988.
- [6] R.X. Qian and C.L. DeMarco, "An approach to robust stability of matrix polytopes through copositive homogeneous polynomials", *IEEE Trans. Automat. Contr.*, vol. 37, pp 848-852, 1992.
- [7] B. R. Barmish, "A generalization of Kharitonov's four polynomial concept for robust stability problems with linearly dependent coefficient perturbations", *IEEE Trans. Automat. Contr.*, vol. 34, pp 157-165, 1989.
- [8] A. Deif, *Sensitivity Analysis in Linear Systems*, Springer-Verlag, Berlin, 1986.
- [9] S. Poljak and J. Rohn, "Checking robust nonsingularity is NP-hard", *Math. Contr., Signals, Syst.*, vol. 6, pp 1-9, 1993.
- [10] E.L. Lawler and D.E. Wood, "Branch-and-bound methods: a survey", *Operations Research*, vol. 14, pp 699-719, 1966.
- [11] E. Balas, "A note on the branch-and-bound principle", *Operations Research*, vol. 16, pp 442-445, 1968.
- [12] T. Doup, *Simplicial Algorithms on the Simplotope*, Springer-Verlag, Berlin, 1988.