SPECTRA - a Maple library for solving linear matrix inequalities in exact arithmetic*

Didier Henrion† Simone Naldi‡ Mohab Safey El Din§

Abstract

This document describes our freely distributed Maple library SPECTRA, for Semidefinite Programming solved Exactly with Computational Tools of Real Algebra. It solves linear matrix inequalities with symbolic computation in exact arithmetic and it is targeted to small-size, possibly degenerate problems for which symbolic infeasibility or feasibility certificates are required.

Keywords

Computer algebra, symbolic computation, linear matrix inequalities, semidefinite programming, low rank matrices, real algebraic geometry.

*This work was partly supported by project GeoLMI of the French National Research Agency (ANR). The first author was partly supported by project 16-19526S of the Grant Agency of the Czech Republic (GAČR) The second author was partly supported by grant PL 549/3-1 (Convexity in Real Algebraic Geometry) of the German Research Foundation (DFG).

†LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France; Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic.

‡Technische Universität Dortmund, Fakultät für Mathematik, Vogelpothsweg 87, 44227 Dortmund, Germany.

§Sorbonne Universités, UPMC Univ Paris 06, CNRS, INRIA, LIP6, PolSys Team, F-75005, Paris, France.

1 Introduction

Given symmetric matrices $A_0, A_1, \ldots, A_n$ of size m with rational coefficients, let

$$\mathscr{S} := \{x \in \mathbb{R}^n : A(x) := A_0 + A_1 x_1 + \cdots + A_n x_n \succeq 0\}$$

denote the corresponding convex *spectrahedron*, defined by the linear matrix inequality (LMI) enforcing that A is positive semidefinite, or equivalently that the eigenvalues of A, as functions of x, are all nonnegative. Spectrahedra are a broad generalization of polyhedra [21]. Like polyhedra, spectrahedra have facets, edges and vertices. However, while the facets of a polyhedron are necessarily flat, the facets of a spectrahedron can be curved outwards or inflated, see Figure 1 for an example of a spectrahedron of dimension $n = 3$ defined by an LMI of size $m = 5$.

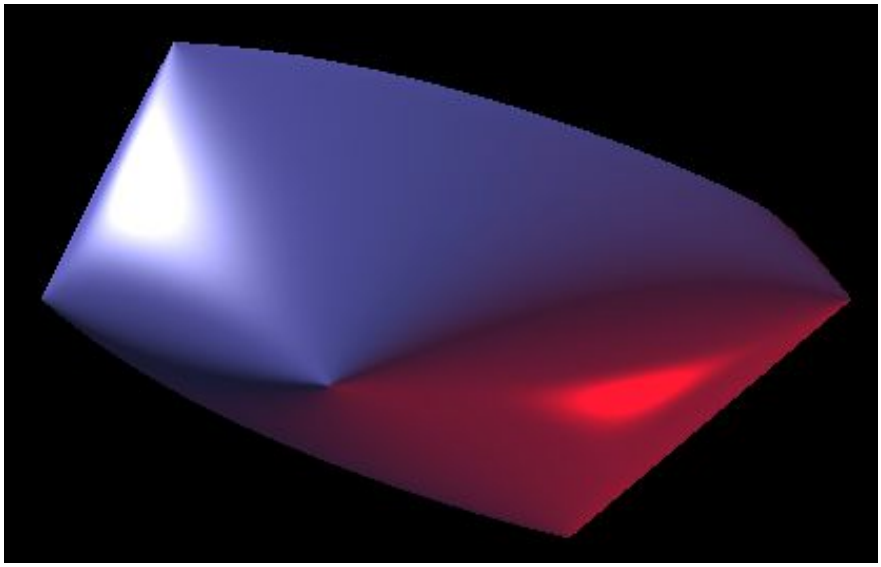

Figure 1: A spectrahedron.

Optimization of a linear function on a spectrahedron is called semidefinite programming (SDP), a broad generalization of linear programming (LP) with many applications in control engineering, signal processing, combinatorial optimization, mechanical structure design, etc, see [22, 20]. The algebra and geometry of spectrahedra is an active area of study in real algebraic geometry, especially in connection with the problem of moments and the decomposition of real multivariate polynomials as sums-of-squares (SOS), see [11, 1, 17] and references therein. SDP-based methods have recently been developed in the setting of error analysis of roundoff during floating-point computations, see [12, 3], or in non-commutative real algebraic geometry, see [4, 10].

Our software SPECTRA aims at either proving that $\mathscr{S}$ is empty, or finding at least one point in $\mathscr{S}$, using *exact arithmetic*. Indeed, SPECTRA is exclusively based on computations with exact arithmetic. This is in sharp contrast with existing SDP solvers which are based on approximate computations and floating point arithmetic, such as the projection and rounding heuristics of e.g. [16] or [9], the primal-dual interior-point SDP solvers in SeDuMi [19] or

NCAlgebra [7], or the implementation in arbitrary precision arithmetic of the interior-point SDP solver in SDPA-GMP [23, 13]. Since exact computations are potentially expensive, SPECTRA should be used when the number n of variables or the size m of the LMI are small. It should not be considered as a competitor to numerical algorithms such as interior-point methods in terms of practical performance when the input has large size (measured by matrix size and number of unknowns). Finite or multiple-precision implementations of the interior point method can handle examples of LMI that are unreachable by SPECTRA.

However, SPECTRA should be primarily used either in potentially degenerate situations, for example when it is expected that $\mathscr{S}$ has empty interior, or when a rigorous certificate of infeasibility or feasibility is required. In these situations, finite precision or arbitrary precision implementations of interior point methods are not able to guarantee the existence of a feasible solution and hence to solve the associated LMI rigorously. For instance, the tests performed in [13, Sec.V] on ill-posed instances from SDPLIB [2] show that the absence of interior point leads to numerical instabilities.

We describe now how we represent exactly the solution of an LMI. The input provided to SPECTRA is the set of matrices $A_0, A_1, \ldots, A_n$ with rational coefficients describing the pencil A and hence the spectrahedron $\mathscr{S}$. If $\mathscr{S}$ is empty, SPECTRA returns the empty list. Otherwise, the output generated by SPECTRA is a finite set

$$\mathscr{Z} := \left\{ \left(\frac{q_1(z)}{q_0(z)}, \frac{q_2(z)}{q_0(z)}, \cdots, \frac{q_n(z)}{q_0(z)} \right) : q(z) = 0,\ z \in \mathbb{C} \right\}. \tag{1}$$

described by a collection of univariate polynomials with integer coefficients $q, q_0, q_1, \ldots, q_n \in \mathbb{Z}[z]$ and such that $\mathscr{Z}$ meets $\mathscr{S}$ in at least one real point x^*. Such a description is called a rational parametrization. It allows to isolate the (generally irrational) coordinates of x^* in rational intervals of length given a priori, as small as required.

If $\mathscr{S}$ is not empty, SPECTRA is guaranteed to compute a point x^* minimizing the rank of A in $\mathscr{S}$. It solves exactly the (non-convex) optimization problem

$$\begin{array}{rcll} r(A) & := & \min & \operatorname{rank} A(x) \\ & & \text{s.t.} & x \in \mathscr{S}. \end{array} \tag{2}$$

This is in sharp contrast with interior-point methods which are designed to compute a point of maximal rank. As detailed below in Section 2, the rank of the matrix $A(x^*)$, respectively the algebraic degree of the entries of $A(x^*)$, can be certified exactly by SPECTRA. This cannot be achieved with classical approaches via interior-point methods, even in arbitrary precision arithmetic.

The outline of the remainder of the paper is as follows. In Section 2 we survey some background material and extract the essential theoretical results of [8] on which SPECTRA relies. In Section 3 we provide instructions to download and install SPECTRA, and we illustrate its use on two elementary examples. More advanced examples are described in Section 4. The performance of SPECTRA on larger examples is reported in Section 5. Finally, in Section 6 we describe formally the exact input and output syntaxes of `SolveLMI`, the main function of SPECTRA.

2 Background material and main theoretical results

The algorithm implemented in SPECTRA computes points in the *determinantal varieties*

$$\mathscr{D}_r := \{x \in \mathbb{C}^n : \operatorname{rank} A(x) \leq r\}$$

for $r = 0, 1, \ldots, m-1$. By construction it holds

$$\mathscr{D}_0 \subset \mathscr{D}_1 \subset \cdots \mathscr{D}_{m-1}.$$

Since the determinant of A vanishes on the boundary $\partial\mathscr{S}$ of $\mathscr{S}$, it holds

$$\partial\mathscr{S} \subset \mathscr{D}_{m-1} \cap \mathbb{R}^n.$$

When $\mathscr{S}$ is not empty, the value $r(A)$ of the optimization problem (2) is the minimum integer r such that $\mathscr{D}_r \cap \mathbb{R}^n$ intersects $\mathscr{S}$, namely the smallest rank on the spectrahedron. Our main geometrical result [8, Theorem 2] states that the spectrahedron contains at least one of the connected components of the real part of the determinantal variety of smallest rank:

Theorem 1 (Smallest rank on a spectrahedron) *Assume that $\mathscr{S}$ is not empty. Let $\mathscr{C}$ be a connected component of $\mathscr{D}_{r(A)} \cap \mathbb{R}^n$ such that the intersection $\mathscr{C} \cap \mathscr{S}$ is not empty. Then $\mathscr{C} \subset \mathscr{S}$ and hence $\mathscr{C} \subset (\mathscr{D}_{r(A)} \setminus \mathscr{D}_{r(A)-1}) \cap \mathbb{R}^n$.*

As a consequence of this result, an algorithm computing at least one point in each connected component of $\mathscr{D}_{r(A)} \cap \mathbb{R}^n$ will compute at least one point in the spectrahedron $\mathscr{S}$. Since the value of $r(A)$ is not known beforehand in general, SPECTRA proceeds iteratively by computing at least one point in the real determinantal variety $\mathscr{D}_r \cap \mathbb{R}^n$ for increasing values of the expected rank $r = 0, 1, \ldots, m-1$.

More specifically, SPECTRA computes points in the determinantal varieties $\mathscr{D}_r$ by projecting onto the subspace of x variables the *incidence varieties*

$$\mathscr{V}_r := \{(x, y) \in \mathbb{C}^n \times \mathbb{C}^{m(m-r)} : A(x)Y(y) = 0, \ \operatorname{rank} Y(y) = m - r\}$$

for $r = 0, 1, \ldots, m-1$. The reader familiar with SDP duality will recognize the classical complementarity conditions, see e.g. [22, 20]. The dual matrix

$$Y(y) = \begin{pmatrix} y_{1,1} & \cdots & y_{1,m-r} \\ \vdots & & \vdots \\ y_{m,1} & \cdots & y_{m,m-r} \end{pmatrix}$$

depends linearly on the dual variables y, and some normalization constraint should be added to ensure that $\operatorname{rank} Y(y) = m - r$. Unlike $\mathscr{D}_r$, the incidence variety $\mathscr{V}_r$, up to genericity conditions on the pencil A , turns to be *smooth and equidimensional*. This crucial geometric property allows for the application of a recursive method which is guaranteed to find at least one point in each connected component of the incidence variety. This leads to the main algorithmic result [8, Theorem 3] on which SPECTRA relies:

Theorem 2 (Exact algorithm for finding a point in a spectrahedron) *Suppose that for each $r = 0, 1, \ldots, m-1$, the incidence variety $\mathscr{V}_r$ is smooth and equidimensional and that its defining polynomial system generates a radical ideal. Suppose also that the determinantal variety $\mathscr{D}_r$ has the expected dimension $n - \binom{m-r+1}{2}$. Then, there is a probabilistic algorithm that takes A as input and returns:*

1. *either the empty list, if and only if $\mathscr{S}$ is empty, or*
2. *a vector x^* such that $A(x^*) = 0$, if and only if the linear system $A(x) = 0$ has a solution, or*
3. *a rational parametrization $q, q_0, q_1, \ldots, q_n \in \mathbb{Z}[z]$ such that there exists $z^* \in \mathbb{R}$ with $q(z^*) = 0$ and:*
   - *$A(q_1(z^*)/q_0(z^*), \ldots, q_n(z^*)/q_0(z^*)) \succeq 0$ and*
   - *rank $A(q_1(z^*)/q_0(z^*), \ldots, q_n(z^*)/q_0(z^*)) = r(A)$.*

The probabilistic nature of the algorithm comes from random changes of variables performed during the procedure, allowing to put the sets $\mathscr{D}_r$ in generic position.

Recall that the incidence varieties $\mathscr{V}_r$ are defined by enforcing a full column rank constraint on the dual matrix $Y(y)$. In SPECTRA this is achieved as follows [8, Section 3.1]: given a subset of $m - r$ dinstinct integers between 1 and r, we enforce the submatrix of $Y(y)$ whose rows are indexed by these integers to be equal to the identity matrix of size $m - r$. For a given value of r, there are $\binom{m}{r}$ distinct choices of row indices and hence the same number of normalized incidence varieties. For each value of r, the algorithm in SPECTRA processes iteratively these normalized incidence varieties.

Finally, let us explain briefly how SPECTRA is able to certify the correctness of the output. This explanation was not included in our paper [8], but we believe it is useful for readers interested in the implementation details. For each computed solution (x^*, y^*) belonging to a connected component of an incidence variety, SPECTRA uses exact arithmetic to decide whether $A(x^*)$ is positive semidefinite and to evaluate the rank of $A(x^*)$. If $A(x^*)$ is not positive semidefinite, then the point x^* is discarded. From Theorem 1 we know that at least one computed point x^* lies on the spectrahedron $\mathscr{S}$, and this point is of minimal rank, i.e. it solves problem (2).

We first build the following characteristic polynomial:

$$s \mapsto p(s, x) = \det(sI_m + A(x)) = s^m + p_1(x)s^{m-1} + \cdots + p_{m-1}(x)s + p_m(x),$$

where I_m is the identity matrix of size m. The coefficient $p_k(x) \in \mathbb{Q}[x]$ has degree k in x and it is the k-th elementary symmetric polynomial of the eigenvalues of $A(x)$, for $k = 1, \ldots, m$. For example, $p_1(x)$ is the trace of $A(x)$ and $p_m(x)$ is the determinant of $A(x)$. This computation is done only once.

Let $x^* \in \mathbb{R}^n$ be given. The rank defect of $A(x^*)$ is equal to the number of consecutive zeros in the sequence $p_m(x^*), p_{m-1}(x^*), \ldots$ Moreover, by Descartes' rule of signs, $A(x^*) \succeq 0$ if and only if $p_k(x^*) \geq 0$ for all $k = 1, \ldots, m$. Hence, computing exactly the rank of $A(x^*)$

and checking its positive semidefiniteness amounts to determining the signs of $p_k(x^*)$ for $k = 1, \ldots, m$.

Whereas this sign determination is a delicate issue when using floating arithmetic and approximate computation, it can be done exactly with SPECTRA, since we represent the point x^* with a rational univariate parametrization with coefficients in $\mathbb{Z}$. Indeed, suppose that x^* belongs to the finite set $\mathscr{Z}$ described as in (1) by the integer coefficient polynomials $q, q_0, q_1, \ldots, q_n$. Together with the rational intervals isolating each entry of x^*, SPECTRA computes rational intervals isolating each coefficient $p_k(x^*)$. Each isolating interval is gradually reduced, until it is so small that at the interval bounds the coefficient takes 1) distinct signs, in which case it vanishes somewhere in the interval, or 2) the same sign, in which case it does not vanish in the whole interval.

3 Getting started

SPECTRA is freely available as a library for Maple version 16 and above. It can be downloaded in the form of single binary file `SPECTRA.mla` from the following page

`homepages.laas.fr/henrion/software/spectra`

SPECTRA relies on FGB, a library for fast computation of Gröbner bases, whose Maple interface must be installed, see [6]. SPECTRA does not work without FGB.

In a Maple worksheet, from the directory containing the file `SPECTRA.mla`, please type the command

```
> with(SPECTRA);
```

to activate the main function `SolveLMI`.

3.1 Half disk

Let

$$A(x) = \begin{bmatrix} 1+x_1 & x_2 & 0 \\ x_2 & 1-x_1 & 0 \\ 0 & 0 & x_1 \end{bmatrix}$$

with $n = 2$ and $m = 3$. The corresponding spectrahedron

$$\mathscr{S} = \{x \in \mathbb{R}^2 : A(x) \succeq 0\} = \{x \in \mathbb{R}^2 : 1 - x_1^2 - x_2^2 \geq 0,\ x_1 \geq 0\}$$

is a half disk. To find a point in $\mathscr{S}$, we use SPECTRA as follows:

```
> A := Matrix([[1+x1, x2, 0], [x2, 1-x1, 0], [0, 0, x1]]):
> SolveLMI(A);
  [[x1 = [0, 0], x2 = [1, 1]]]
```

This returns the point

$$x = [0, 1] \in \mathscr{S}$$

in interval arithmetic notation, i.e.

$$x_1 \in [0, 0],\ x_2 \in [1, 1]$$

and for each component in x we obtain rational (exact) lower and upper bounds. Here the bounds coincide as the point has rational coordinates.

At this point, matrix $A(x)$ is guaranteed to have minimal rank over all points in $\mathscr{S}$. This rank can be obtained as follows:

```
> SolveLMI(A,{rnk});
  [[x1 = [0, 0], x2 = [1, 1], rnk = 1]]
```

3.2 Degenerate spectrahedra

Let us modify the bottom right entry in the matrix of the previous section, so that now

$$A(x) = \begin{bmatrix} 1 + x_1 & x_2 & 0 \\ x_2 & 1 - x_1 & 0 \\ 0 & 0 & x_1 - 1 \end{bmatrix}$$

and the corresponding spectrahedron $\mathscr{S} = \{x \in \mathbb{R}^2 : A(x) \succeq 0\} = \{[1, 0]\}$ reduces to a point in the plane. SPECTRA can easily deal with such a degenerate case:

```
> A := Matrix([[1+x1, x2, 0], [x2, 1-x1, 0], [0, 0, x1-1]]):
> SolveLMI(A);
  [[x1 = [1, 1], x2 = [0, 0]]]
```

Now let us modify further the bottom right entry, letting

$$A(x) = \begin{bmatrix} 1 + x_1 & x_2 & 0 \\ x_2 & 1 - x_1 & 0 \\ 0 & 0 & x_1 - 2 \end{bmatrix}$$

so that the corresponding spectrahedron is empty. SPECTRA returns the empty list, and this is a certificate of emptiness:

```
> A := Matrix([[1+x1, x2, 0], [x2, 1-x1, 0], [0, 0, x1-2]]):
> SolveLMI(A);
  []
```

Since SPECTRA is based on exact arithmetic, it is not sensitive to numerical roundoff errors or small parameter changes:

```
> A := Matrix([[1+x1, x2, 0], [x2, 1-x1, 0], [0, 0, x1-1-10^(-20)]]):
> SolveLMI(A);
  []
> A:=Matrix([[1+x1, x2, 0], [x2, 1-x1, 0], [0, 0, x1-1+10^(-20)]]):
> SolveLMI(A);
  [[x1 = [36893488147418995335 / 36893488147419103232,
          4611686018427401391 / 4611686018427387904],
    x2 = [-350142318592414077 / 24758800785707605497982484448,
          -2801138548739304423 / 19807040628566084398385987584]]
```

Displayed with 10 significant digits, the latter point reads:

$$x_1 \in [\tfrac{36893488147419103232}{36893488147419103232}, \tfrac{4611686018427401391}{4611686018427387904}] \approx 1.000000000,$$

$$x_2 \in [\tfrac{-350142318592414077}{24758800785707605497982484448}, \tfrac{-2801138548739304423}{19807040628566084398385987584}] \approx -0.1414213562 \cdot 10^{-9}.$$

The above point is an irrational solution, and the rational intervals are provided so that their floating point approximations are correct up to the number of digits specified in the Maple environment variable `Digits`, which is by default equal to 10. Use the command

```
> Digits:=100:
```

prior to calling `SolveLMI` if you want an approximation correct to 100 digits. At the price of increased computational burden, SPECTRA then provides larger integer numerators and denominators in the coordinate intervals.

4 Examples

4.1 Irrational spectrahedron

In general, each coordinate of a point computed by SPECTRA is an algebraic number, i.e. the root of a univariate polynomial with integer coefficients.

For the classical univariate matrix

$$A(x_1) = \begin{bmatrix} 1 & x_1 & 0 & 0 \\ x_1 & 2 & 0 & 0 \\ 0 & 0 & 2x_1 & 2 \\ 0 & 0 & 2 & x_1 \end{bmatrix}$$

the spectrahedron reduces to the irrational point $x_1 = \sqrt{2}$. The simple call

```
> A:=Matrix([[1, x1, 0, 0], [x1, 2, 0, 0], [0, 0, 2*x1, 2], [0, 0, 2, x1]]):
> SolveLMI(A);
  [[x1 = [26087635650665550353 / 18446744073709551616,
          13043817825332807945 / 9223372036854775808]]]
```

returns an interval enclosure valid to 10 digits. We can however obtain an exact representation of this point via a rational parametrization:

```
> SolveLMI(A, {par});
  [[x1 = [..], par = [_Z^2-2,_Z,[2]]]]
```

The output parameter `par` contains three univariate polynomials q, q_0, q_1 such that the computed point is contained in the finite set

$$\mathscr{Z} = \{q_1(z)/q_0(z) : q(z) = 0\} = \{2/z : z^2 - 2 = 0\} = \{\pm\sqrt{2}\}$$

as in (1). Here obviously the rational interval isolates the irrational point $x_1 = \sqrt{2}$.

4.2 Algebraic degree

The algebraic degree of semidefinite programming was studied in [15]. Let us consider the spectrahedron of Example 4 in this reference, for which

$$A(x) = \begin{bmatrix} 1+x_3 & x_1+x_2 & x_2 & x_2+x_3 \\ x_1+x_2 & 1-x_1 & x_2-x_3 & x_2 \\ x_2 & x_2-x_3 & 1+x_2 & x_1+x_3 \\ x_2+x_3 & x_2 & x_1+x_3 & 1-x_3 \end{bmatrix}$$

The following point can be easily found with SPECTRA, and it has rank 2, which is guaranteed to be the minimal rank achieved in the spectrahedron:

```
> A:=Matrix([[1+x3, x1+x2, x2, x2+x3], [x1+x2, 1-x1, x2-x3, x2],
             [x2, x2-x3, 1+x2, x1+x3], [x2+x3, x2, x1+x3, 1-x3]]):
> SolveLMI(A, {rnk});
  [[x1 = [29909558235590963953/36893488147419103232,
          29909558235593946897/36893488147419103232],
    x2 = [-18555206088021567643/36893488147419103232,
          -9277603044010395249/18446744073709551616],
    x3 = [-12556837519724045701/36893488147419103232,
          -12556837519723709525/36893488147419103232],
    rnk = 2]]
```

With the following instruction we can indeed certify that there is no point of rank 1 or less:

```
> SolveLMI(A, {}, [1]);
  []
```

The command

```
> SolveLMI(A, {par});
```

returns the following rational univariate parametrization (1) of the above rank 2 point:

$$
\begin{aligned}
q(z) &= 16144z^{10} + 35160z^9 + 14536z^8 - 17690z^7 - 16278z^6 - 2001z^5 + 1556z^4 + 454z^3 + 23z^2 - 4z - 1\\
q_0(z) &= 161440z^9 + 316440z^8 + 116288z^7 - 123830z^6 - 97668z^5 - 10005z^4 + 6224z^3 + 1362z^2 + 46z - 4\\
q_1(z) &= 97248z^9 + 146144z^8 - 18192z^7 - 134826z^6 - 63302z^5 + 4048z^4 + 6758z^3 + 846z^2 - 49z - 14\\
q_2(z) &= 34456z^9 + 37516z^8 - 8734z^7 - 22150z^6 - 8223z^5 - 3978z^4 - 1324z^3 + 104z^2 + 103z + 13\\
q_3(z) &= -35160z^9 - 29072z^8 + 53070z^7 + 65112z^6 + 10005z^5 - 9336z^4 - 3178z^3 - 184z^2 + 36z + 10
\end{aligned}
$$

The degree of the polynomial q in this parametrization can be obtained with the command

```
> SolveLMI(A, {deg});
```

We can obtain more points in the spectrahedron as follows:

```
> SolveLMI(A, {all, rnk, deg}, [2]);
```

This returns 4 feasible solutions of rank $r = 2$, all parametrized by the above degree 10 polynomial. Notice that this degree matches with the algebraic degree of a generic semidefinite programming problem with parameters $(m, n, r) = (4, 3, 2)$, which is 10 according to [15, Table 2].

4.3 Reproducibility

Consider the matrix

$$A(x) = \begin{bmatrix} 1 + x_1 & x_2 \\ x_2 & 1 - x_1 \end{bmatrix}$$

modeling the unit disk. Two consecutive calls to `SolveLMI` return two distinct points:

```
> A:=Matrix([[1+x1,x2],[x2,1-x1]]):
> SolveLMI(A);
 [[x1 = [-2120105604406202 7875/36893488147419103232, -662533001376936933/1152921504606846976],
   x2 = [-754836360701898 8253/9223372036854775808, -188709090175474 2967/230584300921369 3952]]]
> SolveLMI(A);
   [[x1 = [-10862500438565607907/59029581035870 5651712, -217250008771311772 15/118059162071741 1303424],
     x2 = [-576363141759828805/576460752303423488, -9221810268157244495/9223372036854775808]]]
```

After another call, or on your own computer, these intervals should still differ as SPECTRA makes random changes of coordinates to ensure that the geometric objects computed are in general position. This kind of behavior is expected when there are infinitely many points of minimal rank in the spectrahedron.

To generate reproducible outputs, the instruction `randomize` can be used to seed the random number generator used by Maple:

```
> randomize(31415926):
> SolveLMI(A);
  [[x1 = [-3520473351342110 4993/36893488147419103232, -352047335134210 00447/368934881474191 03232],
    x2 = [-2758579864857623899/9223372036854775808, -5517159729715231413/18446744073709551616]]]
> randomize(31415926):
> SolveLMI(A);
  [[x1 = [-3520473351342110 4993/36893488147419103232, -352047335134210 00447/368934881474191 03232],
    x2 = [-2758579864857623899/9223372036854775808, -5517159729715231413/18446744073709551616]]]
```

4.4 Convex quartic

Let

$$A(x) = \begin{bmatrix} 1+x_1 & x_2 & 0 & 0 \\ x_2 & 1-x_1 & x_2 & 0 \\ 0 & x_2 & 2+x_1 & x_2 \\ 0 & 0 & x_2 & 2-x_1 \end{bmatrix}.$$

The spectrahedron $\mathscr{S} = \{x \in \mathbb{R}^2 : A(x) \succeq 0\}$ is the orange region whose boundary is the internal oval of the smooth quartic determinantal curve $\{x \in \mathbb{R}^2 : \det A(x) = 0\}$ represented in black on Figure 2. With the following instructions

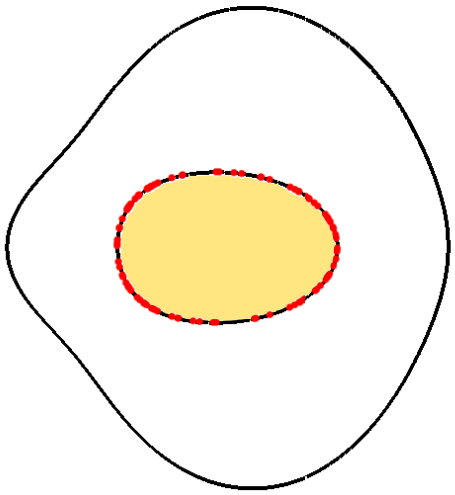

Figure 2: Quartic curve (black) with sample points (red) on the boundary of its spectrahedron (orange).

```
> A:=Matrix([[1+x1,x2,0,0],[x2,1-x1,x2,0],[0,x2,2+x1,x2],[0,0,x2,2-x1]]):
> SolveLMI(A,{},[3]);
> SolveLMI(A,{},[3]);
> ...
```

we compute several points on the boundary of $\mathscr{S}$, they are plotted in red on Figure 2. Note the third input argument `[3]` which specifies to `SolveLMI` the expected rank of the computed point. Since the determinantal curve is smooth, we know that the rank of $A(x)$ equals 3 on the whole curve, and in particular on the boundary of $\mathscr{S}$. Since the rank is specified, SPECTRA does not have to process iteratively the incidence varieties corresponding to points of smaller ranks, thereby reducing the computational burden to find at least one point in the spectrahedron.

Each of these points is represented by a rational univariate parametrization of degree 12, obtained with the instruction

```
> SolveLMI(A,{par},[3]);
```

For example, for the point $(x_1, x_2) \approx (-0.9689884394, -0.2434013983)$ the polynomial q in the rational parametrization (1) is

$$\begin{aligned} q(z) = {} & 5506034827600\, z^{12} - 4608031295324\, z^{10} - 192908794368\, z^9 + 25693318717857\, z^8 + \\ & + 4774492660608\, z^7 - 17188212283956\, z^6 - 23438418515712\, z^5 + 64967482316484\, z^4 - \\ & - 11285164470528\, z^3 - 11887630039728\, z^2 + 296990121024. \end{aligned}$$

Recall that the algebraic degree of a point x^* in $\mathscr{S}$ is the degree of the minimal algebraic extension of the ground field (here the rational numbers) required to represent x^*. The algebraic degree depends on the size of the pencil A but also on the rank r of $A(x^*)$. With $(m, n, r) = (4, 2, 3)$ and generic data, the algebraic degree is 12, cf. [15, Table 2], which indeed coincides with the degree of the exact representation of x^* computed by SPECTRA.

4.5 Polynomial sums of squares

Deciding whether a multivariate real polynomial is non-negative is difficult in general. A sufficient condition, or certificate for non-negativity, is that the polynomial can be expressed as a sum of squares (SOS) of other polynomials. Finding a polynomial SOS decomposition amounts to finding a point in a specific spectrahedron called Gram spectrahedron, see e.g. [5] and references therein.

Consider the homogeneous ternary quartic

$$f(u) = u_1^4 + u_1u_2^3 + u_2^4 - 3u_1^2u_2u_3 - 4u_1u_2^2u_3 + 2u_1^2u_3^2 + u_1u_3^3 + u_2u_3^3 + u_3^4.$$

The polynomial f belongs to a series of examples provided by C. Scheiderer in [18] to answer (in the negative) the following question by B. Sturmfels: let f be a polynomial with rational coefficients which is an SOS of polynomials with real coefficients; is it an SOS of polynomials with rational coefficients? Scheiderer's counterexamples prove that, generally speaking, there is no hope in obtaining nonnegativity certificates over the rationals. However, certificates exist in some algebraic extension of the field of rational numbers.

In the graded reverse lexicographic ordered monomial basis, the Gram matrix of f is the matrix

$$A(x) = \begin{bmatrix} 1 & 0 & x_1 & 0 & -3/2 - x_2 & x_3 \\ 0 & -2x_1 & 1/2 & x_2 & -2 - x_4 & -x_5 \\ x_1 & 1/2 & 1 & x_4 & 0 & x_6 \\ 0 & x_2 & x_4 & -2x_3 + 2 & x_5 & 1/2 \\ -3/2 - x_2 & -2 - x_4 & 0 & x_5 & -2x_6 & 1/2 \\ x_3 & -x_5 & x_6 & 1/2 & 1/2 & 1 \end{bmatrix}$$

depending linearly on 6 real parameters. The Gram spectrahedron $\mathscr{S} = \{x \in \mathbb{R}^6 : A(x) \succeq 0\}$ parametrizes the set of all SOS decompositions of f. We deduce by the discussion above that $\mathscr{S}$ does not contain rational points. In particular, its interior is empty.

Let us use SPECTRA to compute points in $\mathscr{S}$ and hence to get positivity certificates for f:

```
> A := Matrix([[1,0,x1,0,-3/2-x2,x3], [0,-2*x1,1/2,x2,-2-x4,-x5], [x1,1/2,1,x4,0,x6],
              [0,x2,x4,-2*x3+2,x5,1/2], [-3/2-x2,-2-x4,0,x5,-2*x6,1/2], [x3,-x5,x6,1/2,1/2,1]]):
> SolveLMI(A, {rnk, deg, par});
[[[x1 = [..], x2 = [..], x3 =  [..], x4 = [..], x5 = [..], x6 = [..]],
    rnk = 2, deg = 3,
    par = [8*z^3+8*z+1, 24*z^2-8, [16*z+3, -24*z^2+8, 8*z^2+6*z+8, -16*z^2+6*z+16, -16*z-3, 16*z+3]]]
```

We obtain an irrational point $x \in \mathscr{S}$ whose coordinates are algebraic numbers of degree 3, and which belongs to the finite set

$$\left\{ \left(\frac{16z+3}{24z^2-8}, \frac{-24z^2+8}{24z^2-8}, \frac{8z^2+6z+8}{24z^2-8}, \frac{-16z^2+6z+16}{24z^2-8}, \frac{-16z-3}{24z^2-8}, \frac{16z+3}{24z^2-8} \right) : 8z^3-8z-1=0 \right\}$$

At this point, the Gram matrix A has rank 2, and hence f is an SOS of 2 polynomials.

Let us compute more non-negativity certificates of rank 2:

```
> SolveLMI(A,{rnk,deg,par,all},[2]);
```

In addition to the point already obtained above, we get another point. The user can compare this output with [18, Ex. 2.8]: it turns out that these are the only 2 points of rank 2. Other points in the Gram spectrahedron have rank 4 and they are convex combinations of these 2.

4.6 Application to computer arithmetic

In the paper [3] the authors need to check the following inequality in order to derive bounds for roundoff errors of floating-point computation performed on complex numbers $a_0 + ib_0$ and $a_1 + ib_1$:

$$\frac{32}{7}\left(a_0^2+b_0^2\right)\left(a_1^2+b_1^2\right) - \left(2a_0a_1 - b_0b_1\right)^2 - \left(2a_0b_1 + 2a_1b_0\right)^2 \geq 0. \tag{3}$$

Relaxing positivity to sums of squares, as in the previous section, allows to derive a SOS-certificate using SPECTRA. The dense Gram matrix associated to the polynomial in (3) has size 10×10 and it is linear in 21 variables. The current implementation of SPECTRA allows to solve the associated LMI in 5 hours. However, exploiting the sparsity of the polynomial in (3) allows to reduce the LMI description to a 4×4 linear matrix, and the computation to a few seconds.

5 Performance

5.1 Exponential bit-size spectrahedron

For a given $n \in \mathbb{N}$, consider the spectrahedron

$$\mathscr{S}_n = \left\{ x \in \mathbb{R}^n : \begin{bmatrix} 1 & 2 \\ 2 & x_1 \end{bmatrix} \succeq 0, \begin{bmatrix} 1 & x_1 \\ x_1 & x_2 \end{bmatrix} \succeq 0, \cdots, \begin{bmatrix} 1 & x_{n-1} \\ x_{n-1} & x_n \end{bmatrix} \succeq 0 \right\}.$$

For every $x^* \in \mathscr{S}_n$ it holds $x_n^* \geq (x_{n-1}^*)^2 \geq \cdots \geq (x_1^*)^{2^{n-1}} \geq 2^{2^n}$, which shows that exponentially many bits are required to represent a point. It is elementary to check that each of the above n matrices of size 2 can have rank 1, and hence that we can compute a point of rank n as follows:

```
> with(LinearAlgebra):
> A:=DiagonalMatrix([<<1,2>|<2,x1>>,<<1,x1>|<x1,x2>>,<<1,x2>|<x2,x3>>,..]):
> SolveLMI(A,{},[n]);
```

Recall from Section 2 that SPECTRA examines iteratively a family of $\binom{m}{r} = \binom{2n}{n}$ incidence varieties, a number growing exponentially with n. For example there are $12870 = \binom{16}{8}$ incidence varieties to test to solve our problem for $n = 8$. Hence we could expect SPECTRA to perform poorly on this example. However, on our standard desktop PC equipped with Intel i7 processor at 2.5GHz and 16GB RAM, we were able to handle spectrahedra of size $2n = 10$ in 29 seconds, and of size $2n = 12$ in 505 seconds.

5.2 Random spectrahedra

Finally, we report on randomly generated examples. The rational entries of A are generated as quotients of integers drawn uniformly in the interval $[-100, 100]$. Here is the script we used to generate a random symmetric pencil given the number n of variables and the size m:

```
> var:=[seq(cat('x',i),i=1..n)]:
> A:=Matrix(m,m):
> for i from 1 to m do
     for j from i to m do
       A[i,j]:=randpoly(var, degree=1, dense, coeffs=rand(-100..100)):
       A[j,i]:=A[i,j];
     od:
   od:
```

For each instance, given the expected rank r, we execute the command

```
> SolveLMI(A,{},[r]);
```

For $m = 2$, $r = 1$ and values of n ranging from 30 to 100, we obtain the timings reported on Figure 3. This corresponds to spectrahedra whose boundaries belong to determinantal varieties of increasing dimension. Moreover, the singularity locus of the determinant has positive dimension, it is a linear subspace of co-dimension 3. We observe a polynomial dependence of the computational time as a function of the number of variables, with exponent around 3.

When $m = 3$, $r = m - 1 = 2$ and values of n ranging from 30 to 80, we obtain the timings reported on Figure 4, depending polynomially on n with an exponent around 4.

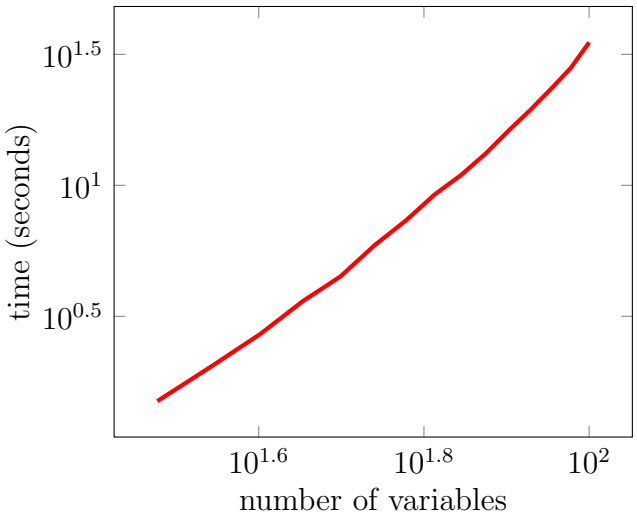

Figure 3: Timings for random instances of size $m = 2$ and rank $r = 1$, as a function of the number of variables n.

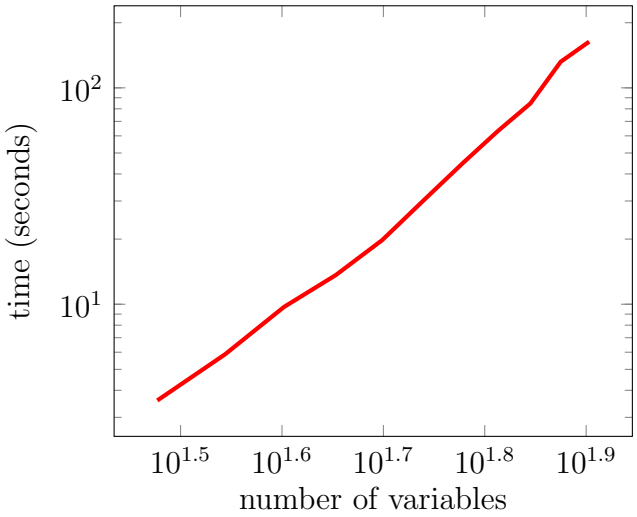

Figure 4: Timings for random instances of size $m = 3$ and rank $r = 2$, as a function of the number of variables n.

6 Input and output syntax

6.1 Input

The calling sequence of function `SolveLMI` is as follows:

```
> SolveLMI(A, options, ranks);
```

where

- A is a symmetric matrix of size m with rational coefficients, depending affinely on n variables;

- `options` (optional) is a set that can contain the following keywords:

  `all` : compute as many solutions as possible, which can be computationally demanding; when this option is not specified, the algorithm is stopped as soon as one solution is computed, which is typically much faster;

  `rnk` : return the rank of A at every computed solution;

  `par` : return the rational univariate parametrization of every computed solution;

  `deg` : return the algebraic degree of every computed solution;

- `ranks` (optional) is a list of nonnegative integers corresponding to expected ranks of computed solutions. The default value is $[0, 1, \ldots, m-1]$. The algorithm is run for each value r in `ranks` by solving the quadratic system of equations

$$A(x)Y(y) = 0$$

  for a vector x and a matrix $Y(y)$ with m rows and $m - r$ columns whose entries are stored in a vector y. It may happen that the rank of $A(x)$ at a computed solution x is strictly less than r.

6.2 Output

Let us denote by $x_1, x_2, \ldots, x_n$ the variables on which matrix A depends affinely. They are gathered in a vector $x \in \mathbb{R}^n$. When the input argument `options` is empty, the output returned by `SolveLMI` is

- eithter the empty list `[]` in which case $\mathscr{S}$ is empty, or
- a rational enclosure of a single point $x \in \mathscr{S}$, in the form

```
> SolveLMI(A)
  [[x1 = [a1, b1], x2 = [a2, b2], ..., xn = [an, bn]]]
```

  where a_i, b_i are rational numbers, displayed as ratios of integers. This means that each coordinate x_i belongs to the interval $[a_i, b_i]$ ensuring a floating point approximation of x valid to a number of digits equal to the Maple environment variable `Digits`. When $a_i = b_i$ this implies that x_i is a rational number.

When `options` contains the keyword `all`, more points can be returned, in the form of a list

```
> out := SolveLMI(A, {all})
  [[x1 = [a11, b11], x2 = [a12, b12], ..., xn = [a1n, b1n]],
   [x1 = [a21, b21], x2 = [a22, b22], ..., xn = [a2n, b2n]],
   ...]
```

such that `nops(out)` is the number of computed points, `out[1]` is the first point, `out[2]` is the second point, etc.

When `options` contains the keyword `rnk`, the rank of A at x is returned:

```
> SolveLMI(A, {rnk})
  [[x1 = [a1, b1], x2 = [a2, b2], ..., xn = [an, bn], rnk = r]]
```

These keywords and the following ones can be freely combined:

```
> SolveLMI(A, {all, rnk})
  [[x1 = [a11, b11], x2 = [a12, b12], ..., xn = [a1n, b1n], rnk = r1],
   [x1 = [a21, b21], x2 = [a22, b22], ..., xn = [a2n, b2n], rnk = r2],
   ...]
```

When `options` contains the keyword `par`, a rational univariate parametrization of x is returned:

```
> SolveLMI(A, {par})
  [[x1=[a1,b1], x2=[a2,b2], ..., xn=[an,bn], par=[q,q0,[q1,q2,...,qn]]]
```

This parametrization is such that $q, q_0, q_1, q_2, \ldots, q_n$ are univariate polynomials with integer coefficients such that x belongs to the finite set

$$\left\{ \left(\frac{q_1(z)}{q_0(z)}, \frac{q_2(z)}{q_0(z)}, \cdots, \frac{q_n(z)}{q_0(z)} \right) : q(z) = 0,\ z \in \mathbb{C} \right\}.$$

The intervals $[a_i, b_i]$ are provide to isolate the computed point from this set of points.

When `options` contains the keyword `deg`, the degree of the polynomial q in the rational univariate parametrization of each computed point x is also returned:

```
> SolveLMI(A, {deg})
  [[x1 = [a1, b1], x2 = [a2, b2], ..., xn = [an, bn], deg = d]]
```

References

[1] G. Blekherman, P. A. Parrilo, R. R. Thomas, editors. Semidefinite Optimization and Convex Algebraic Geometry. MOS-SIAM Series on Optimization 13, 2012.

[2] B. Borchers. SDPLIB 1.2, a library of semidefinite programming test problems. Optimization Methods and Software 11:683-690, 1999.

[3] R. Brent, C. Percival, and P. Zimmermann. Error bounds on complex floating-point multiplication. Mathematics of Computation 76(259):1469–1481, 2007.

[4] K. Cafuta, I. Klep, J. Povh. Rational sums of hermitian squares of free noncommutative polynomials. Ars Mathematica Contemporanea, 9:243–259, 2015.

[5] L. Chua, D. Plaumann, R. Sinn, C. Vinzant. Gram spectrahedra. Arxiv Preprint: `arXiv:1608.00234`, July 2016.

[6] J.-C. Faugère. FGb: a library for computing Gröbner bases. In K. Fukuda, J. van der Hoeven, M. Joswig, N. Takayama, editors, Mathematical Software - ICMS 2010, Lecture Notes in Computer Science, vol. 6327, pp. 84–87, Springer, Berlin, 2010. Software available at `www-polsys.lip6.fr/~jcf/FGb`

[7] J.W. Helton, M.C. de Oliveira, M. Stankus, R.L. Miller. NCAlgebra. Software available at `http://math.ucsd.edu/~ncalg`

[8] D. Henrion, S. Naldi, M. Safey El Din. Exact algorithms for linear matrix inequalities. Siam J. on Optimization. 26(4):2512–2539, 2017.

[9] E. L. Kaltofen, B. Li, Z. Yang, L. Zhi. Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients. J. Symbolic Comput. 47(1):1–15, 2012.

[10] I. Klep, M. Schweighofer. Sums of Hermitian squares and the BMV conjecture. Journal of Statistical Physics, 133(4):739–760, 2008.

[11] J. B. Lasserre. Moments, Positive Polynomials and Their Applications. Imperial College Press, London, UK, 2010.

[12] V. Magron, M. Farid. Certified lower bounds of roundoff errors using semidefinite programming. arXiv Preprint arXiv:1611.01318, 2016.

[13] M. Nakata. A numerical evaluation of highly accurate multiple-precision arithmetic version of semidefinite programming solver: SDPA-GMP, -QD and -DD. Proceedings of the IEEE Multi-Conference on Systems and Control, 2010.

[14] S. Naldi. Exact algorithms for determinantal varieties and semidefinite programming. PhD thesis, Univ. Toulouse and Univ. Pierre et Marie Curie Paris, September 2015. Available at `tel.archives-ouvertes.fr/tel-01212502`.

[15] J. Nie, K. Ranestad, B. Sturmfels. The algebraic degree of semidefinite programming. Math. Prog. 122(2):379–405, 2010.

[16] H. Peyrl, P. A. Parrilo. Computing sum of squares decompositions with rational coefficients. Theoretical Comput. Sci. 409:269–281, 2008

[17] M. Ramana, A.J. Goldman. Some geometric results in semidefinite programming. Journal of Global Optimization, 7(1):33–50, 1995.

[18] C. Scheiderer. Sums of squares of polynomials with rational coefficients. J. Eur. Math. Soc. 18(7):1495–1513, 2016.

[19] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optimization Methods and Software, 11/12(1–4):625–653, 1999.

[20] M. J. Todd. Semidefinite optimization. Acta Numerica, 10:515–560, 2001.

[21] C. Vinzant. What is... a spectrahedron ? Notices of the AMS, 61(5):492–494, 2014.

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

[23] M. Yamashita, K. Fujisawa, M. Fukuda, K. Kobayashi, K. Nakta, M. Nakata. Latest developments in the SDPA Family for solving large-scale SDPs. In Handbook on Semidefinite, Cone and Polynomial Optimization: Theory, Algorithms, Software and Applications. In M.F. Anjos and J.B. Lasserre (Editors). Springer, NY,USA. Chapter 24, pp. 687–714, 2011.