



**HAL**  
open science

# Step by step design of an interior point solver in semidefinite optimization

Jean Charles Gilbert

► **To cite this version:**

Jean Charles Gilbert. Step by step design of an interior point solver in semidefinite optimization: Application to the Shor relaxation of some small OPF problems. Master. Advance Continuous Optimization II, France. 2015. cel-01252612v1

**HAL Id: cel-01252612**

**<https://inria.hal.science/cel-01252612v1>**

Submitted on 8 Jan 2016 (v1), last revised 19 Dec 2019 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Step by step design of an interior point solver in semidefinite optimization – Application to the Shor relaxation of some small OPF problems

Jean Charles GILBERT, INRIA

January 8, 2016

This report presents a project in numerical optimization dealing with the implementation of an interior point method for solving a semidefinite optimization (SDO) problem. The project was given in a course entitled *Advanced Continuous Optimization II* at the University Paris-Saclay, in 2015-1016. The solver is designed step by step during a series of 5 sessions of 4 hours each. Each session corresponds to a chapter of the report.

The correctness of the SDO solver is verified at each session on small academic problems, having diverse properties. During the last session, the developed piece of software is used to solve a few small size Shor relaxations of QCQP (quadratically constrained quadratic programming) problems, modeling various instances of the OPF (optimal power flow) problem.

The goal of the course is not to design an SDO solver that would beat the best existing solver but to help the students to understand and demystify what there is inside such a solver. Famous SDO solvers include SeDuMi, SDPT3, Gurobi, MOSEK, and GLPK.



# Table of Contents

<b>1</b>	<b>The NT direction</b> .....	5
1.1	The problem to solve .....	5
1.1.1	Notation .....	5
1.1.2	The primal and dual SDO problems .....	5
1.1.3	The central path .....	6
1.2	The NT direction .....	7
1.2.1	Overview .....	7
1.2.2	Derivation of the NT direction .....	8
1.2.3	Computation of $W$ .....	9
1.2.4	Computation of the NT direction .....	9
1.3	Implementation .....	10
1.3.1	Matrix representation .....	10
1.3.2	Calling statement .....	11
1.3.3	Matlab functions .....	11
1.3.4	Recommendations .....	12
1.3.5	Test case 1: an easy $3 \times 3$ matrix problem .....	12
1.4	Questions .....	12
<b>2</b>	<b>A predictor-corrector algorithm</b> .....	13
2.1	Algorithmic techniques .....	13
2.1.1	The closest central point .....	13
2.1.2	Neighborhood of the central path .....	13
2.1.3	A predictor-corrector algorithm .....	14
2.2	Implementation .....	15
2.2.1	The MTY predictor-corrector algorithm .....	15
2.2.2	Recommendations .....	16
2.2.3	Test cases 2: minimum matrix norm .....	16
2.3	Questions .....	17
<b>3</b>	<b>Finding an appropriate starting point</b> .....	19
3.1	Getting a feasible point close to the central path .....	19
3.2	Implementation .....	22
3.2.1	Getting close to the central path .....	22
3.2.2	Test case 3: global minimization of a univariate polynomial ...	23
3.3	Question .....	26

<b>4</b>	<b>Starting from an infeasible point</b> .....	27
4.1	Starting from an infeasible point .....	27
4.1.1	Getting linear feasibility .....	27
4.1.2	Starting from a strictly primal feasible point .....	27
4.1.3	Starting from a strictly dual feasible point .....	29
4.1.4	Starting without a strictly feasible point .....	32
4.2	Implementation .....	35
4.2.1	Implementation .....	35
4.2.2	Recommendations .....	35
4.2.3	Test case 4a .....	35
4.2.4	Test case 4b .....	35
4.2.5	Test case 4c .....	35
4.3	Questions .....	36
<b>5</b>	<b>Shor relaxation of some small OPF problems</b> .....	37
5.1	Relaxation of an OPF problem .....	37
5.1.1	QCQP formulation .....	37
5.1.2	Shor relaxation .....	38
5.2	Test cases .....	39
	<b>References</b> .....	41

# 1 The NT direction

## 1.1 The problem to solve

### 1.1.1 Notation

We denote by  $\mathcal{S}^n$  the vector space of the  $n \times n$  symmetric real matrices, equipped with the scalar product

$$\langle \cdot, \cdot \rangle : (A, B) \in \mathcal{S}^n \times \mathcal{S}^n \mapsto \langle A, B \rangle = \text{tr } AB = \sum_{i,j=1}^n A_{ij} B_{ij},$$

where  $\text{tr } A := \sum_i A_{ii}$  denotes the trace of the matrix  $A \in \mathbb{R}^{n \times n}$ . The associated norm is the so-called *Frobenius norm*; its value at  $A \in \mathcal{S}^n$  is denoted by

$$\|A\|_F = \left( \sum_{i,j=1}^n A_{ij}^2 \right)^{1/2}.$$

The vector space  $\mathcal{S}^n$  has dimension  $n(n+1)/2$ .

We denote by  $\mathcal{S}_+^n$  the cone of  $\mathcal{S}^n$  made of the positive semidefinite matrices and by  $\mathcal{S}_{++}^n$  the cone of  $\mathcal{S}^n$  made of the positive definite matrices. We adopt the short notation

$$\begin{aligned} A \succcurlyeq B &\iff A - B \in \mathcal{S}_+^n, \\ A \succ B &\iff A - B \in \mathcal{S}_{++}^n. \end{aligned}$$

Of course,  $A \preccurlyeq B$  means  $B \succcurlyeq A$  and  $A \prec B$  means  $B \succ A$ .

### 1.1.2 The primal and dual SDO problems

The primal ( $P$ ) and dual ( $D$ ) semidefinite optimization (SDO) problems read

$$\boxed{(P) \quad \begin{cases} \inf \langle C, X \rangle \\ A(X) = b \\ X \succcurlyeq 0 \end{cases}} \quad \text{and} \quad \boxed{(D) \quad \begin{cases} \sup \langle b, y \rangle \\ A^*(y) + S = C \\ S \succcurlyeq 0, \end{cases}}$$

where  $C \in \mathcal{S}^n$ ,  $A : \mathcal{S}^n \rightarrow \mathbb{R}^m$  is a linear map,  $b \in \mathbb{R}^m$ , and  $A^* : \mathbb{R}^m \rightarrow \mathcal{S}^n$  is the adjoint of  $A$ . These problems are therefore convex in the sense that their objective is

convex and their feasible sets are also convex. Constraints like those of problem  $(P)$  or  $(D)$  are called *linear matrix inequalities* (LMI).

The feasible sets of  $(P)$  and  $(D)$  are respectively denoted by

$$\mathcal{F}_P := \{X \in \mathcal{S}_+^n : A(X) = b\} \quad \text{and} \quad \mathcal{F}_D := \{(y, S) \in \mathbb{R}^m \times \mathcal{S}_+^n : A^*(y) + S = C\},$$

and their strictly feasible sets are denoted by

$$\mathcal{F}_P^s := \{X \in \mathcal{S}_{++}^n : A(X) = b\} \quad \text{and} \quad \mathcal{F}_D^s := \{(y, S) \in \mathbb{R}^m \times \mathcal{S}_{++}^n : A^*(y) + S = C\}.$$

Accordingly, a matrix  $X \in \mathcal{F}_P^s$  is said to be *strictly feasible for  $(P)$*  and a pair  $(y, S) \in \mathcal{F}_D^s$  is said to be *strictly feasible for  $(D)$* . We also introduce the following Cartesian products

$$\mathcal{F} := \mathcal{F}_P \times \mathcal{F}_D \quad \text{and} \quad \mathcal{F}^s := \mathcal{F}_P^s \times \mathcal{F}_D^s.$$

The solution sets of these problems are denoted by

$$\text{Sol}(P) \quad \text{and} \quad \text{Sol}(D)$$

and their optimal values by

$$\text{val}(P) \quad \text{and} \quad \text{val}(D).$$

The *duality gap* is the nonnegative difference  $\text{val}(P) - \text{val}(D) \geq 0$ .

In the implementation, it will be assumed that the linear map  $A$  is represented by  $m$  matrices  $A_i \in \mathcal{S}^n$ , for  $i \in [1:m]$ , through the correspondence

$$A(X) = \begin{pmatrix} \langle A_1, X \rangle \\ \vdots \\ \langle A_m, X \rangle \end{pmatrix}. \quad (1.1)$$

Such a representation is always possible by the Riesz-Fréchet theorem. It will be also assumed that the Euclidean scalar product is used on  $\mathbb{R}^m$ , in which case  $A^*(y)$  is a weighted sum of the matrices  $A_i$  (see question 1.1):

$$A^*(y) = \sum_{i=1}^m A_i y_i. \quad (1.2)$$

### 1.1.3 The central path

It is known that when  $\mathcal{F}^s \neq \emptyset$ ,  $(X, y, S)$  is a primal-dual solution to  $(P)$  and  $(D)$  if and only if

$$\begin{cases} A^*(y) + S = C, & S \succcurlyeq 0 \\ A(X) = b, & X \succcurlyeq 0 \\ \langle X, S \rangle = 0. \end{cases} \quad (1.3)$$

These may be considered as the NS optimality conditions of the convex problems  $(P)$  or  $(D)$ .

When  $A$  is surjective, the central path is the (image of the) map, which to  $\mu \in \mathbb{R}_{++}$  assigns the unique solution to the perturbed optimality conditions

$$\begin{cases} A^*(y) + S = C, & S \succ 0 \\ A(X) = b, & X \succ 0 \\ XS = \mu I. \end{cases} \quad (1.4)$$

When  $\mathcal{F}^s \neq \emptyset$ , the central path converges to a primal-dual solution, when  $\mu \downarrow 0$ . The goal of the *path-following* algorithms that we shall consider is to follow that path to reach a solution asymptotically.

## 1.2 The NT direction

### 1.2.1 Overview

Given a strictly feasible point  $z = (X, y, S) \in \mathcal{F}^s \subset \mathcal{S}^n \times \mathbb{R}^m \times \mathcal{S}^n$ , the goal of one iteration of the primal-dual path-following interior point algorithm we consider is to make a displacement  $d$  in  $\mathcal{S}^n \times \mathbb{R}^m \times \mathcal{S}^n$  towards a central point  $z(\mu)$ , for some  $\mu > 0$ , that is “closer” to the solution than the current iterate  $z$ . This displacement  $d = (d_X, d_y, d_S) \in \mathcal{S}^n \times \mathbb{R}^m \times \mathcal{S}^n$  is a Newton step on the perturbed system (1.4) without its conic conditions  $X \succ 0$  and  $S \succ 0$  (these will be enforced at each iteration), namely

$$\begin{cases} A^*(y) + S = C \\ A(X) = b \\ XS = \mu I. \end{cases} \quad (1.5)$$

When  $z \in \mathcal{F}$  is not on the central path, the matrix  $XS$  is usually not symmetric, so that there are more equations than unknowns in (1.5). To tell it otherwise, the linearization of the system (1.5) reads

$$\begin{cases} A^*(d_y) + d_S = C - A^*(y) - S \\ A(d_X) = b - A(X) \\ d_X S + X d_S = \mu I - XS. \end{cases}$$

If  $d_S$  is symmetric by the first equation, there are examples [12] showing that  $d_X$  might be nonsymmetric (because of the nonsymmetry of  $X d_S S^{-1}$  in the third equation). For this reason a symmetrization operation must be introduced in addition to the linearization of the system, in order to keep  $X + d_X$  in  $\mathcal{S}^n$ . This can be done using a large number of methods, leading to many different IP directions. The one we consider in this project is the *Nesterov-Todd* (NT) direction (called that way because of [15, 16, 21; 1997-1998], but the direction might have been proposed independently and earlier in the later published paper [20; 1999]). It is often implemented and it has the following advantages:

- it preserves the symmetry of the primal-dual variables  $X$  and  $S$ ,
- it is uniquely defined for any strictly feasible iterate  $z$ ,
- it is scale invariant,
- it is less computation demanding than many other directions, requiring . . . operations.



### 1.2.2 Derivation of the NT direction

The NT direction can be shortly presented as the result of three operations [3].

1. A *scaling matrix*  $W \in \mathcal{S}_{++}^n$  is introduced (a weight), which is the unique matrix in  $\mathcal{S}_{++}^n$  that satisfies the identity

$$WSW = X. \quad (1.6)$$

It is given by the formulas

$$W := X^{1/2}(X^{1/2}SX^{1/2})^{-1/2}X^{1/2} = S^{-1/2}(S^{1/2}XS^{1/2})^{1/2}S^{-1/2} \quad (1.7)$$

and it has the property that

$$Z := W^{-1/2}XW^{-1/2} = W^{1/2}SW^{1/2}.$$

2. *Symmetrization.* If  $z = (X, y, S)$  is the current iterate, one would like to find a displacement  $d = (d_X, d_y, d_S)$  such that the third equation in (1.5) is satisfied at  $z + d$ , namely

$$(X + d_X)(S + d_S) = \mu I,$$

for some  $\mu > 0$  to be defined. Left-multiplying by  $W^{-1/2}$  and right-multiplying by  $W^{1/2}$ , this equation is equivalent to

$$(Z + \tilde{d}_X)(Z + \tilde{d}_S) = \mu I,$$

where  $\tilde{d}_X = W^{-1/2}d_XW^{-1/2}$  and  $\tilde{d}_S = W^{1/2}d_SW^{1/2}$ . We now *symmetrize* the lhs of this equation by the transformation  $M \rightsquigarrow \frac{1}{2}(M + M^T)$ :

$$\frac{1}{2} \left[ (Z + \tilde{d}_X)(Z + \tilde{d}_S) + (Z + \tilde{d}_S)(Z + \tilde{d}_X) \right] = \mu I.$$

3. *Pseudo-linearization* (the term ‘‘pseudo’’ is used because the scaling matrix  $W$ , which depends on the current iterate, is not linearized). The linearization consists in dropping the products  $\tilde{d}_X\tilde{d}_S$  and  $\tilde{d}_S\tilde{d}_X$  from this equation, which then becomes

$$Z(\tilde{d}_X + \tilde{d}_S) + (\tilde{d}_X + \tilde{d}_S)Z = \mu I - Z^2.$$

This is a *Lyapunov equation* in the unknown  $\tilde{d}_X + \tilde{d}_S$ . Its unique solution is

$$\tilde{d}_X + \tilde{d}_S = \mu Z^{-1} - Z.$$

Left and right-multiplying this last equation by  $W^{1/2}$  yield the symmetric pseudo-linearization at  $z$  of the third equation in (1.5):

$$d_X + Wd_SW = \mu S^{-1} - X.$$

Therefore, when  $z \in \mathcal{F}^s$ , the NT direction is obtained by solving the following linear system in  $d = (d_X, d_y, d_S)$ :

$$A^*(d_y) + d_S = 0 \quad (1.8a)$$

$$A(d_X) = 0 \quad (1.8b)$$

$$d_X + Wd_SW = \mu S^{-1} - X, \quad (1.8c)$$

where the value of  $\mu$  still need to be specified.

### 1.2.3 Computation of $W$

The scaling matrix  $W$  given by (1.7) intervenes in the computation of the NT direction and must be computed explicitly. This can be done using two Cholesky factorizations and one singular value decomposition (SVD) [21; § 4.1].

The Cholesky factorizations are those of  $X \succ 0$  and  $S \succ 0$ :

$$\boxed{X = L_X L_X^\top} \quad \text{and} \quad \boxed{S = L_S L_S^\top},$$

where  $L_X$  and  $L_S$  are lower triangular nonsingular matrices. The SVD is the one of  $L_S^\top L_X$ :

$$\boxed{L_S^\top L_X = U \Sigma V^\top},$$

where  $U$  and  $V$  are orthogonal matrices and  $\Sigma$  is the diagonal matrix formed of the positive singular values of the nonsingular matrix  $L_S^\top L_X$ . Define

$$Q := L_X^{-1} X^{1/2},$$

which is an orthogonal matrix. It needs not be computed.

Now, there holds

$$\begin{aligned} X^{1/2} S X^{1/2} &= X^{1/2} \underbrace{L_X^{-\top} L_X^\top}_I \underbrace{L_S L_S^\top}_S \underbrace{L_X L_X^{-1}}_I X^{1/2} \\ &= Q^\top \underbrace{L_X^\top L_S}_V \underbrace{L_S^\top L_X}_U Q \\ &= Q^\top V \Sigma^2 V^\top Q. \end{aligned}$$

From this identity and the orthogonality of  $Q^\top V$ , it results

$$(X^{1/2} S X^{1/2})^{-1/2} = Q^\top V \Sigma^{-1} V^\top Q.$$

From the first identity in (1.7), we get

$$W = \underbrace{X^{1/2} Q^\top}_{L_X} V \Sigma^{-1} V^\top \underbrace{Q X^{1/2}}_{L_X^\top} = L_X V \Sigma^{-1} V^\top L_X^\top.$$

Finally

$$\boxed{W = G G^\top, \quad \text{where } G := L_X V \Sigma^{-1/2}.$$

### 1.2.4 Computation of the NT direction

The system (1.8) involves  $n(n+1) + m$  unknowns  $d = (d_X, d_y, d_S)$  and as many equations. A standard way of solving it consists in eliminating first  $d_X$  using the second and third equation, to get

$$A(W d_S W) = A(\mu S^{-1} - X)$$

and eliminating next  $d_S$  using the first equation:

$$A(WA^*(d_y)W) = A(X - \mu S^{-1}) \quad (1.9)$$

Once  $d_y$  is known,  $d_S$  can be computed by the first equation in (1.8) and then  $d_X$  by the third equation in (1.8). So let us concentrate on ways of computing  $d_y$  by (1.9). We assume that  $A$  is surjective, which implies that (1.9) is a linear system in  $d_y$  with a positive definite linear operator, hence having a unique solution.

- A first way of computing the solution  $d_y$  to (1.9) is to write this equation as the linear system in  $d_y$ :

$$\boxed{\mathcal{M}(d_y) = A(X - \mu S^{-1})}, \quad (1.10)$$

where  $\mathcal{M} : d_y \in \mathbb{R}^m \mapsto A(WA^*(d_y)W) \in \mathbb{R}^m$  is a symmetric positive definite (hence nonsingular) linear map and to form the “compact” (usually  $m \ll n^2$ )  $m \times m$  coefficient matrix  $\mathcal{M}$  by

$$\mathcal{M}_{ij} = \langle A_i, WA_jW \rangle = \langle G^T A_i G, G^T A_j G \rangle, \quad \text{for } i \text{ and } j \in [1:n], \text{ with } i \geq j, \quad (1.11)$$

and to take its Cholesky factorization  $\mathcal{M} = L_{\mathcal{M}}L_{\mathcal{M}}^T$ . Forming  $\mathcal{M}$  mainly requires  $2mn^3$  operations (for getting the products  $G^T A_i G$ , with  $i \in [1:m]$ ).

- Another way of computing  $d_y$  is to observe that (1.9) or (1.10) is the normal equation of the following linear least-squares problem in  $d_y$ :

$$\boxed{\min_{d_y \in \mathbb{R}^m} \frac{1}{2} \|\mathcal{A}(d_y) - G^{-1}(X - \mu S^{-1})G^{-T}\|_F^2}, \quad (1.12)$$

where  $\mathcal{A} : d_y \in \mathbb{R}^m \mapsto G^T A^*(d_y)G \in \mathcal{S}^n$  is a linear map. Indeed, the normal equation of this least-squares problem reads

$$\mathcal{A}^* \mathcal{A}(d_y) = \mathcal{A}^*(G^{-1}(X - \mu S^{-1})G^{-T}) = A(X - \mu S^{-1}),$$

where  $\mathcal{A}^* : M \in \mathcal{S}^n \rightarrow A(GMG^T) \in \mathbb{R}^m$ . Hence

$$\begin{aligned} \mathcal{A}^* \mathcal{A}(d_y) &= A(G\mathcal{A}(d_y)G^T) = A(WA^*(d_y)W) = \mathcal{M}(d_y), \\ \mathcal{A}^*(G^{-1}(X - \mu S^{-1})G^{-T}) &= A(X - \mu S^{-1}) \end{aligned}$$

and one recovers (1.10). The linear least-squares problem (1.12) can be solved thanks to a QR factorization of the associated  $n^2 \times m$  large matrix, also denoted  $\mathcal{A}$ , whose  $j$ th column is the matrix

$$\mathcal{A}_{:j} = G^T A_j G, \quad \text{for } j \in [1:m].$$

which was already used in (1.11). By the stability of the QR factorization, this approach should be more precise but more time consuming.

## 1.3 Implementation

### 1.3.1 Matrix representation

We suggest taking vector representations of matrices. A *symmetric*  $n \times n$  matrix will also be represented by an  $n^2$  vector (not by an  $n(n+1)/2$  one). This has the advantage

of making the computation of the scalar product  $\langle X, S \rangle$  easy (it is just  $\mathbf{x}' * \mathbf{s}$  if  $\mathbf{x}$  is the vector representing the matrix  $X$  and  $\mathbf{s}$  is the vector representing the matrix  $S$ ). Furthermore, the evaluation of  $A(X)$  is just  $\mathbf{A} * \mathbf{x}$  and the evaluation of  $A^*(y)$  is just  $\mathbf{A}' * \mathbf{y}$  if the rows of  $\mathbf{A}$  contains the matrices  $A_i$  for  $i \in [1 : m]$ . Now switching from a vector to a matrix representation and vice versa can be done with the Matlab functions `mat` and `vec` (see below), probably inexpensively if the result is not safeguarded.

### 1.3.2 Calling statement

We suggest giving to the `sdo` solver, the following Matlab function structure

```
[x,y,infos] = sdo (A,b,c,x0,y0,options)
```

where

- $\mathbf{A}$  is an  $m \times n^2$  real matrix, each row of which gives the entries of a symmetric matrix  $A_i$  of the representation of the linear operator  $A$  in (1.1);
- $\mathbf{b}$  is an  $m$  real vector, representing the right hand side of the equality constraint in (P);
- $\mathbf{c}$  is an  $n^2$  vector representing the  $n \times n$  symmetric real matrix  $C$ , which determines the linear objective in (P);
- $\mathbf{x0}$  is an  $n^2$  vector representing the  $n \times n$  symmetric real matrix  $X_0$ , which is the initial guess of the primal solution  $X$ ; it must be *strictly* feasible for the primal problem, meaning that  $A(X) = b$  and  $X \succ 0$ ;
- $\mathbf{y0}$  is an  $m$  real vector, which is the initial guess of the dual solution  $y$ ; it must be *strictly* feasible for the dual problem, meaning that  $S_0 := C - A^*(y_0) \succ 0$ ;
- `options` is a structure of possible options (one could specify whether the NT direction must be computed with a Cholesky factorization, like in (1.10), or a QR factorization, like in (1.12));
- $\mathbf{x}$  is an  $n^2$  vector representing the  $n \times n$  symmetric real matrix  $X$ , which gives the primal solution  $X$  if any;
- $\mathbf{y}$  is an  $m$  real vector, giving the dual solution  $y$  if any (the dual solution  $S$  can be recovered from  $y$  by  $S = C - A^*(y)$ );
- `infos` is a structure providing information of the run (success, failure, primal/dual infeasibility, primal/dual unboundedness, duality gap, *etc*).

### 1.3.3 Matlab functions

Here are a few useful Matlab functions.

- `L = chol(A, 'lower')` computes the Cholesky factorization of  $A = LL^T$ , where  $L$  is lower triangular (only the lower triangular part of  $A$  is used).
- `[U,S,V] = svd(A)` computes the singular value decomposition (SVD) of the real  $n \times n$  matrix  $A = USV^T$ , where  $U$  and  $V$  are orthogonal matrices and  $S$  is diagonal with nonnegative entries (the singular values of  $A$ ).
- `M = mat(v)` converts an  $n^2$  vector  $\mathbf{v}$  into an  $n \times n$  matrix  $\mathbf{M}$ , such that the columns of  $\mathbf{M}$  are stacked below each other in  $\mathbf{v}$ . The reverse operation is realized by `vec`.
- `[Q,R] = qr(A,0)` computes the QR factorization of the real  $m \times n$  matrix  $A = QR$ , where  $Q$  is orthogonal and  $R$  is upper triangular. The second zero argument

is useful when  $m > n$  (our case), in which case, only the first  $n$  column of  $Q$  and the first  $n$  rows of  $R$  are computed.

- $\mathbf{v} = \text{vec}(\mathbf{M})$  converts an  $m \times n$  matrix  $\mathbf{M}$  into an  $mn$  vector  $\mathbf{v}$ , stacking the columns of  $\mathbf{M}$  below each other in  $\mathbf{v}$ . The reverse operation is realized by `mat`.

### 1.3.4 Recommendations

There are various Matlab SDO solvers. Having a look at SeDuMi is probably a good idea, since `sdo` may be viewed as a reduced feature version of this solver.

### 1.3.5 Test case 1: an easy $3 \times 3$ matrix problem

Consider the problem in standard dual form

$$\left\{ \begin{array}{l} \sup y_1 + y_2 + y_3 \\ \begin{pmatrix} 1 - y_1 & -y_3 & -y_2 \\ -y_3 & 1 - y_2 & 0 \\ -y_2 & 0 & 1 - y_3 \end{pmatrix} \succcurlyeq 0. \end{array} \right.$$

It is easily verified that  $X_0 = I_3$  and  $(y_0, S_0) = (0, I_3)$  are strictly feasible for  $(P)$  and  $(D)$  respectively, so that the primal and dual problems have a solution without duality gap. There holds  $\langle b, y_0 \rangle = 0 < 3 = \langle C, X_0 \rangle$ , so that  $z_0 := (X_0, y_0, S_0)$  is not a primal-dual solution.

## 1.4 Questions

**1.1.** *Adjoint of A.* Show that the adjoint of the linear map  $A : \mathcal{S}^n \rightarrow \mathbb{R}^m$  defined by (1.1) is indeed given by (1.2) when  $\mathbb{R}^m$  is equipped with the Euclidean scalar product.

**1.2.** *On test case 1.* Let us define

$$\bar{\mu}(z) = \frac{\langle X, S \rangle}{n}.$$

Consider test case 1. Let  $z_1 = z_0 + d_0$ , where  $z_0 = (X_0, y_0, S_0)$  and  $d_0$  is the NT direction computed at  $z_0$  for some  $\mu = \sigma \bar{\mu}(z_0)$  with  $\sigma \in [0, 1]$ .

- 1) Is  $z_0$  on the central path?
- 2) Justify the observed value of  $\bar{\mu}(z_1)$  (check the effect of choosing  $\sigma = 1$  and  $\sigma = 0$ )?
- 3) Is  $z_1$  a primal-dual solution to the problem if  $\sigma = 0$ ?
- 4) Can you get convergence of the sequence  $\{z_k\}$  generated by  $z_{k+1} = z_k + d_k$ , where  $d_k$  is the NT direction at  $z_k$  for some  $\mu_k = \sigma \bar{\mu}(z_k)$  and some  $\sigma$  constant in  $]0, 1[$ ? (Probably not, but this will be done in another session.)
- 5) Can you compute/guess the solution? (Probably not, but this will be done in another session.)

## 2 A predictor-corrector algorithm

### 2.1 Algorithmic techniques

#### 2.1.1 The closest central point

In a path-following algorithm, the first question that arises deals with the determination of the central point  $z(\mu)$  to which the current iterate  $z := (X, y, S)$  is the closest. Indeed, in such an algorithm, the next iterate aims a point on the central path  $\mathcal{C}$  and gets close to it by a Newton step, provided this central point is not too far from  $z$ . If we want the iteration to make a progress towards the solution it is necessary that this aimed central point be closer to the solution than the central point that is the closest to  $z$ , hence the necessity to determine the latter.

Finding the closest central point is not a well defined concept, actually, since the central path is not a convex set, making the projection on  $\mathcal{C}$  not well defined. To overcome the difficulty, we look instead to the closest central point in a transformed space in which the central path becomes a half-line. The transformation is

$$\tau : z = (X, y, S) \in \mathcal{F} \mapsto XS \in \mathbb{R}^{n \times n}.$$

This transformation is certainly not a bijection, but it turns out to be useful, which is enough to make it appropriate. Since  $\mathcal{C} := \{z \in \mathcal{F} : \underline{XS} \in \mathbb{R}_{++}I\}$ , it follows that  $\tau(\mathcal{C}) = \mathbb{R}_{++}I$ . Then, by the convexity and closedness of  $\tau(\mathcal{C})$ , it makes sense to look for the point in  $\tau(\mathcal{C})$  that is the closest to  $\tau(z)$ . That problem simply reads

$$\min_{\mu \geq 0} \|XS - \mu I\|_F^2. \quad (2.1)$$

It has a unique solution, which is

$$\bar{\mu}(z) := \frac{\langle X, S \rangle}{n}. \quad (2.2)$$

The *closest central point* is therefore “considered” to be the one on the central path with the parameter  $\mu = \bar{\mu}(z)$ . This is not irrelevant, since when  $z \in \mathcal{C}$ , it is clear that the closest central point to  $z$  in the preceding sense is  $z$  itself; reassuring.

#### 2.1.2 Neighborhood of the central path

Experimentation has convinced the algorithmicians that it is not good to let the iterates going too far from the central path  $\mathcal{C}$  or, more precisely, too close to the

boundary of the feasible set  $\mathcal{F}$ . When  $z$  is close to that boundary, the stepsize along the Newton direction may be very small, so much as to prevent any progress to the solution: the iterates get stuck on the boundary of  $\mathcal{F}$ . For this reason, the iterates are maintained in some neighborhood of the central path, often the one that we now define. Recall that the central path is an analytic concept (not a geometric one, depending only on the feasible set), so that this is a second best solution.

Observe first that for a point  $z \in \mathcal{F}^s$ :

$$z \in \mathcal{C} \iff XS = \bar{\mu}(z)I \iff Z^2 = \bar{\mu}(z)I \iff \bar{\mu}(z)^{1/2}Z^{-1} = \bar{\mu}(z)^{-1/2}Z,$$

where  $Z := W^{-1/2}XW^{-1/2} = W^{1/2}SW^{1/2}$  and  $W$  is the unique matrix in  $\mathcal{S}_{++}^n$  satisfying  $WSW = X$  (see session 1). Then one defines a “distance” to the central path by<sup>1</sup> ([5] and [3; § 7.1]):

$$\delta : z \in \mathcal{F}^s \mapsto \delta(z) := \frac{1}{2} \left\| \bar{\mu}(z)^{1/2}Z^{-1} - \bar{\mu}(z)^{-1/2}Z \right\|_F \in \mathbb{R}. \quad (2.3)$$

Accordingly, for  $\theta \in [0, 1]$ , one introduces the following neighborhood of the central path

$$\mathcal{V}(\theta) := \{z \in \mathcal{F}^s : \delta(z) \leq \theta\}.$$

We note that

$$\delta(z) = \frac{1}{2} \left\| G^T \left( \bar{\mu}(z)^{1/2}X^{-1} - \bar{\mu}(z)^{-1/2}S \right) G \right\|_F, \quad (2.4)$$

so that the distance can be computed efficiently.

### 2.1.3 A predictor-corrector algorithm

The predictor-corrector algorithms are the most often implemented interior-point methods. We essentially give the description of the *Mizuno-Todd-Ye predictor-corrector method* [14, 3], whose iteration is composed of two phases.

- The *predictor phase* starts with  $z \in \mathcal{V}(1/3)$  and consists in making a displacement along the NT direction with  $\mu = 0$  (known as the *affine-scaling direction*), say  $d^a$ . The stepsize  $\alpha > 0$  along that direction is given by the formula (no need of linesearch)

$$\alpha = \frac{2}{1 + [1 + 13\|\tilde{d}_X^a \tilde{d}_S^a + \tilde{d}_S^a \tilde{d}_X^a\|_F / (2\bar{\mu}(x))]^{1/2}}, \quad (2.5)$$

where

$$\tilde{d}_X^a = W^{-1/2}d_X^a W^{-1/2} \quad \text{and} \quad \tilde{d}_S^a = W^{1/2}d_S^a W^{1/2},$$

in which  $W$  is the scaling matrix used to define the NT direction. It can be shown that

$$\|\tilde{d}_X^a \tilde{d}_S^a + \tilde{d}_S^a \tilde{d}_X^a\|_F = \|G^{-1}d_X^a d_S^a G + G^T d_S^a d_X^a G^{-T}\|_F, \quad (2.6)$$

where  $G$  has been defined to compute the NT direction (there holds  $W = GG^T$ ), so that the computation of this norm can be done efficiently (in  $8n^3$  operations).

The stepsize (2.5) is small enough to ensure that the intermediate iterate  $z' := z + \alpha d^a$  is strictly feasible and large enough to ensure polynomiality of the algorithm (see below).

<sup>1</sup> No square on the norm!

- The *corrector phase*, which follows the predictor step, aims at putting the next iterate in  $\mathcal{V}(1/3)$ , since usually  $z' \notin \mathcal{V}(1/3)$ . The phase consists in getting closer to the central path by making a displacement along the NT direction with  $\mu = \bar{\mu}(z')$  (known as the *centering direction*), say  $d^c$ , and a unit stepsize. It turns out that a single Newton step of that kind is sufficient to have  $z_+ := z' + d^c$  in  $\mathcal{V}(1/3)$ .

**Algorithm 2.1.1 (Mizuno-Todd-Ye predictor-corrector)** A tolerance  $\varepsilon > 0$  on  $\bar{\mu}(z)$  is given to determine when stopping the iterations. The iteration from  $z$  to  $z_+$  starts at some  $z \in \mathcal{V}(1/3)$ .

1. *Stopping test.* If  $\bar{\mu}(z) \leq \varepsilon$ , stop.
2. *Predictor phase.* Compute the NT direction at  $z$  with  $\mu = 0$ , denote it  $d^a$  (affine-scaling direction), and the stepsize  $\alpha$  given by (2.5). Set
 
$$z' := z + \alpha d^a.$$
3. *Corrector phase.* Compute the NT direction at  $z'$  with  $\mu = \bar{\mu}(z')$ , denote it  $d^c$  (centering direction). Take as next iterate
 
$$z_+ := z' + d^c.$$

It can be shown [3] that  $\alpha$  given by (2.5) satisfies

$$\alpha \geq \frac{2}{1 + [1 + 13n/2]^{1/2}},$$

which implies the polynomiality of the algorithm, with the iteration complexity

$$K := \left\lceil \left( 1 + \sqrt{1 + \frac{13n}{2}} \right) \log \frac{\bar{\mu}(z_0)}{\varepsilon} \right\rceil.$$

This means that if the algorithm starts with  $z_0 \in \mathcal{V}(1/3)$ , it ends up with  $z_K$  satisfying  $\bar{\mu}(z_K) \leq \varepsilon$ , where the number  $K$  of iterations is the one given above.

## 2.2 Implementation

### 2.2.1 The MTY predictor-corrector algorithm

Implement the MTY predictor-corrector algorithm and use it to find a solution to the problem in test case 1 (below).

Here is a possible output:

```

-----
SDO solver (version 0.1, December 2015)
-----
Description of the problem
Size of the matrices      3
    
```



Nb of constraints 3				
it	c'*x	gap	cp dist	stepsize
1	3.0000000e+00	1.000e+00	0.0e+00	5.7967e-01
2	2.1192603e+00	4.203e-01	4.4e-03	6.9701e-01
3	1.5919211e+00	1.274e-01	2.0e-03	8.5276e-01
4	1.3805195e+00	1.875e-02	7.3e-04	9.7125e-01
5	1.3442240e+00	5.392e-04	1.2e-04	9.9900e-01
6	1.3431468e+00	5.392e-07	1.8e-06	9.9900e-01
7	1.3431458e+00	5.392e-10	7.7e-08	9.9900e-01
8	1.3431458e+00	5.391e-13	3.4e-05	9.9900e-01

Solution found up to the required precision (1e-10)

The abbreviations are:

- **it**: iteration counter,
- **c'\*x**: primal cost,
- **gap**:  $\bar{\mu}(z)$ ,
- **cp dist**: distance  $\delta(z)$  to the central path, computed by (2.4),
- **stepsize**: stepsize  $\alpha$  along the affine scaling direction  $d^a$ , computed by (2.5)–(2.6).

## 2.2.2 Recommendations

- It is important to check that the following property holds in the Mizuno-Todd-Ye predictor-corrector algorithm:

$$\bar{\mu}(z_+) = \bar{\mu}(z') = (1 - \alpha)\bar{\mu}(z). \quad (2.7)$$

- The value of  $\alpha$  given by (2.5) may be very close to one, or even equal to one. This usually yields matrices  $X$  and/or  $S$  that are no longer positive semidefinite. In this case the algorithm gets stuck in a Cholesky factorization. Limiting this computed value to

$$\text{options.max\_stepsize} = 0.999$$

or so, may help the algorithm to go further and compute a more precise solution.

## 2.2.3 Test cases 2: minimum matrix norm

We consider the *minimum matrix norm* problem which reads

$$\min_{z \in \mathbb{R}^p} \|\mathcal{B}(z)\|_2, \quad (2.8)$$

where  $\mathcal{B}(z) := B_0 + \sum_{i=1}^p z_i B_i$ , the matrices  $B_i \in \mathbb{R}^{q \times r}$  and  $\|\cdot\|_2$  denotes the  $\ell_2$ -norm. This problem is convex, but nonsmooth. It can be expressed as an SDO problem as follows.

Problem (2.8) can be rewritten  $\min_{t,z} \{t : \|\mathcal{B}(z)\|_2^2 \leq t^2\}$ . Using the Schur complement technique, it can be verified that this last problem can also be written as the following SDO problem in dual form [22]:

$$\begin{cases} \max_{(t,z) \in \mathbb{R} \times \mathbb{R}^p} & -t \\ \begin{pmatrix} tI_q & \mathcal{B}(z) \\ \mathcal{B}(z)^\top & tI_r \end{pmatrix} \succeq 0. \end{cases} \quad (2.9)$$

This problem has dimension  $n = q + r$  and  $m = p + 1$ . One can also easily find a strictly feasible primal-dual point (question 2.3).

Solve problem (2.9) with, say  $p = 2$ ,  $q = 2$ ,  $r = 2$ , and random data for the matrices  $B_i \in \mathbb{R}^{q \times r}$ , with  $i \in [0:2]$ .

## 2.3 Questions

- 2.1.** *Formula of  $\bar{\mu}(z)$ .* Show that the solution to problem (2.1) is indeed given by (2.2).
- 2.2.** *Evolution of  $\bar{\mu}(z)$  in the MTY algorithm.* Show that, in the Mizuno-Todd-Ye predictor-corrector algorithm, algorithm 2.1.1, (2.7) holds.
- 2.3.** *Strictly feasible point for test case 2.* Show that, in standard notation,  $(X_0, y_0, S_0)$  with

$$X_0 = \frac{1}{q+r} I_{q+r}, \quad y_0 = (t_0, z_0) = (\|B_0\|_2 + 1, 0), \quad \text{and} \quad S_0 = \begin{pmatrix} t_0 I_q & B_0 \\ B_0^T & t_0 I_r \end{pmatrix}$$

is a strictly feasible primal-dual point for problem (2.9).



## 3 Finding an appropriate starting point

### 3.1 Getting a feasible point close to the central path

The predictor-corrector algorithm that was presented in section 1.3 of session 2 assumes that the first iterate is in some neighborhood  $\mathcal{V}(\theta)$  of the central path ( $\theta = 1/3$ ). In this section, we consider a method to get such a point, provided a strictly feasible primal-dual point  $z_0 = (X_0, y_0, S_0) \in \mathcal{F}^s$  is known. The proposed algorithm can get a point in  $\mathcal{V}(\theta)$  for any prescribed  $\theta > 0$ , in particular  $\theta = 1/3$ .

The idea is simple and consists in solving *approximately* the primal-dual barrier problem [3; §7.1]

$$\min_{z \in \mathcal{F}^s} \left( \varphi_\mu(z) := \langle X, S \rangle + \mu \text{ld}(XS) \right), \quad (3.1)$$

where  $\mu > 0$  is a fixed parameter and  $\text{ld} : \mathbb{R}^{n \times n} \rightarrow \bar{\mathbb{R}}$  is defined at  $M \in \mathbb{R}^{n \times n}$  by

$$\text{ld}(M) = \begin{cases} -\log \det M & \text{if } \det M > 0 \\ +\infty & \text{otherwise.} \end{cases}$$

To achieve this goal primal-dual iterates are generated using NT directions with its parameter  $\mu$  fixed to the one taken in the definition of  $\varphi_\mu$ . A natural value for  $\mu$  is  $\bar{\mu}(z_0)$ .

This strategy is grounded on the following observations and properties.

It makes sense to minimize  $\varphi_\mu$  to get close to the central path, since, when  $A$  is surjective, the unique minimizer of  $\varphi_\mu$  is the central point  $z_\mu$ . This is shown in the next proposition. To see this, we need to compute the gradient of  $\varphi_\mu$ . Recall that for a matrix  $M \in \mathbb{R}^{n \times n}$  (not necessary symmetric, like  $XS$ ) verifying  $\det M > 0$ , there holds

$$\nabla \text{ld}(M) = -M^{-\top}.$$

An easy computation then shows that

$$\nabla \varphi_\mu(z) = \begin{pmatrix} S - \mu X^{-1} \\ 0 \\ X - \mu S^{-1} \end{pmatrix} \quad \text{and} \quad \nabla^2 \varphi_\mu(z)d = \begin{pmatrix} \mu X^{-1} d_X X^{-1} + d_S \\ 0 \\ d_X + \mu S^{-1} d_X s^{-1} \end{pmatrix}. \quad (3.2)$$

**Proposition 3.1.1 (solution of the primal-dual barrier)** *If  $\mathcal{F}^s \neq \emptyset$  and  $A$  is surjective. Then  $\varphi_\mu$  is strictly convex on  $\mathcal{F}^s$  and problem (3.1) has the central point  $z_\mu$  as unique solution.*

PROOF. Let  $z \in \mathcal{F}^s$  and  $z + d \in \mathcal{F}^s$ . Then  $A^*(d_y) + d_S = 0$  and  $A(d_X) = 0$ , from which one deduces  $\langle d_X, d_S \rangle = 0$ . To prove that  $\varphi_\mu$  is strictly convex on the feasible set of (3.1), we just have to show that  $\langle \nabla^2 \varphi_\mu(x)d, d \rangle > 0$  when  $d \neq 0$ . Now there holds

$$\begin{aligned} \langle \nabla^2 \varphi_\mu(x)d, d \rangle &= \mu \langle X^{-1}d_X X^{-1}, d_X \rangle + 2\langle d_X, d_S \rangle + \mu \langle S^{-1}d_X S^{-1}, d_S \rangle \\ &= \mu \|X^{-1/2}d_X X^{-1/2}\|_F^2 + \mu \|S^{-1/2}d_S S^{-1/2}\|_F^2, \end{aligned}$$

which is positive when  $d_X \neq 0$  or  $d_S \neq 0$ , which is the case when  $d \neq 0$  and  $A$  is surjective.

Now, the Lagrangian of problem (3.1) can be written

$$(z, \Lambda, \lambda) \mapsto \langle X, S \rangle + \mu \text{ld}(XS) + \langle \Lambda, A^*(y) + S - C \rangle + \langle \lambda, A(X) - b \rangle,$$

where  $\Lambda \in \mathcal{S}^n$  and  $\lambda \in \mathbb{R}^m$ . If  $z$  minimizes  $\varphi_\mu$  on  $\mathcal{F}^s$ , there are multipliers  $(\Lambda, \lambda)$  such that the following optimality conditions holds (the constraints are qualified by the affinity)

$$\begin{cases} S - \mu X^{-1} + A^*(\lambda) = 0 \\ X - \mu S^{-1} + \Lambda = 0 \\ A(\Lambda) = 0 \\ A^*(y) + S = C \\ A(X) = b. \end{cases}$$

Using the first two identities, one gets

$$0 = \langle S - \mu X^{-1} + A^*(\lambda), \Lambda \rangle = \langle S - \mu X^{-1}, \Lambda \rangle = \langle S - \mu X^{-1}, \mu S^{-1} - X \rangle.$$

Now the matrices  $W$  and  $Z$  of session 1 allow us to write  $X = W^{1/2}Z W^{1/2}$  and  $S = W^{-1/2}Z W^{-1/2}$ , so that the last identity yields

$$0 = \langle W^{-1/2}(Z - \mu Z^{-1})W^{-1/2}, W^{1/2}(\mu Z^{-1} - Z)W^{1/2} \rangle = -\|Z - \mu Z^{-1}\|_F^2.$$

Therefore  $Z = \mu Z^{-1}$  or  $X = \mu S^{-1}$  or  $XS = \mu I$ , meaning that  $z = z_\mu$ .  $\square$

The previous proposition has for consequence that generating iterates for solving (3.1) will eventually provide a point in  $\mathcal{V}(\theta)$  for some prescribed  $\theta > 0$ .

The next result shows that it makes sense to force the decrease of  $\varphi_\mu$  along the NT directions. It also provides in (3.4) the value of the derivative of  $\varphi_\mu$  along the NT direction. The function  $\delta$  is the ‘‘distance’’ to the central path defined by formula (1.3) in session 2:

$$\delta : z \in \mathcal{F}^s \mapsto \delta(z) := \frac{1}{2} \left\| \bar{\mu}(z)^{1/2} Z^{-1} - \bar{\mu}(z)^{-1/2} Z \right\|_F \in \mathbb{R}. \quad (3.3)$$

**Proposition 3.1.2 (decrease of the primal-dual barrier along the NT direction)** *Let  $d$  be the NT direction at a point  $z \in \mathcal{F}^s$  for  $\mu := \bar{\mu}(z)$ . Then*

$$\langle \nabla \varphi_\mu(z), d \rangle = -4\mu \delta(z)^2, \quad (3.4)$$

where  $\delta$  is defined by (3.3). Furthermore the following properties are equivalent:

- (i)  $z \neq z_\mu$ ,
- (ii)  $d \neq 0$ ,
- (iii)  $\langle \nabla \varphi_\mu(z), d \rangle < 0$ .

PROOF. Using (3.2), one gets

$$\langle \nabla \varphi_\mu(z), d \rangle = \langle S - \mu X^{-1}, d_X \rangle + \langle X - \mu S^{-1}, d_S \rangle.$$

Using the scaling matrix  $W \succ 0$  and the scaled directions  $\tilde{d}_X = W^{-1/2} d_X W^{-1/2}$  and  $\tilde{d}_S = W^{1/2} d_S W^{1/2}$ , one gets

$$\langle \nabla \varphi_\mu(z), d \rangle = \langle S - \mu X^{-1}, W^{1/2} \tilde{d}_X W^{1/2} \rangle + \langle X - \mu S^{-1}, W^{-1/2} \tilde{d}_S W^{-1/2} \rangle.$$

It is known that  $Z := W^{-1/2} X W^{-1/2} = W^{1/2} S W^{1/2}$  and that  $\tilde{d}_X + \tilde{d}_S = \mu Z^{-1} - Z$ , so that

$$\begin{aligned} \langle \nabla \varphi_\mu(z), d \rangle &= \langle Z - \mu Z^{-1}, \tilde{d}_X + \tilde{d}_S \rangle \\ &= -\|Z - \mu Z^{-1}\|_F^2 \\ &= -\mu \|\mu^{-1/2} Z - \mu^{1/2} Z^{-1}\|_F^2 \\ &= -4\mu \delta(z)^2, \end{aligned}$$

which is (3.4). The equivalences follow easily. □

**Proposition 3.1.3 (distance to the central path after a NT step)** *Let  $d$  be the NT direction at a point  $z \in \mathcal{F}^s$  for  $\mu := \bar{\mu}(z)$ . Then the distance  $\delta$  to the central path satisfies the inequality*

$$\delta(z + d) \leq \frac{\delta(z)^2}{\sqrt{2(1 - \delta(z)^2)}}. \quad (3.5)$$

PROOF. See [3; lemma 7.4]. □

A consequence of propositions 3.1.1 and 3.1.2 is that, to get closer to the central path, it makes sense to minimize the function  $\varphi_\mu$ , with  $\mu = \bar{\mu}(z)$ , along the NT direction, using a linesearch technique. Furthermore, from (3.5) in proposition 3.1.3,

$$\delta(z) \leq \sqrt{\frac{2\tau^2}{1+2\tau^2}} \implies \delta(z+d) \leq \tau\delta(z),$$

so that, choosing  $\tau < 1$ , the distance decreases linearly without the need to make linesearch, as soon as  $\delta(z) \leq \sqrt{2\tau^2/(1+2\tau^2)}$  (less than  $\sqrt{2/3}$ ). Inequality (3.5) also tells us that the distance  $\delta(z)$  converges to zero quadratically.

This discussion leads to the following algorithm to get a point in the neighborhood  $\mathcal{V}(1/3)$  of the central path, starting from a point  $z \in \mathcal{F}^s$ .

**Algorithm 3.1.4 (getting a point in  $\mathcal{V}(1/3)$ )** The algorithm uses two parameters: a desired linear speed of convergence  $\tau < 1$  close to 1 and a linesearch parameter  $\omega > 0$  close to zero. For example  $\tau \simeq 0.99$  and  $\omega \simeq 10^{-4}$ .

Let  $z \in \mathcal{F}^s$ . The iteration computes  $z_+$  closer to the central path, ensuring  $z \in \mathcal{V}(1/3)$  in a finite number of such iterations.

1. *Stopping test.* If  $\delta(z) \leq 1/3$ , stop.
2. *NT direction.* Compute the NT direction at  $z$  with  $\mu := \bar{\mu}(z)$ , denote it  $d^c$ .
3. *Next iterate.* If  $\delta(z) \leq \sqrt{2\tau^2/(1+2\tau^2)}$ , then compute the next iterate by

$$z_+ := z + d^c.$$

Otherwise, find a stepsize  $\alpha \in ]0, 1]$  such that

$$\varphi_\mu(z + \alpha d^c) \leq \varphi_\mu(z) - 2\omega\mu\alpha\delta(z)^2 \quad (3.6)$$

and set  $z_+ := z + \alpha d^c$ .

The previous algorithm hides a serious difficulty, which may occur during the linesearch when (3.6), which evaluates the quality of the trial stepsize  $\alpha$ . By definition, the function  $\varphi(z) = +\infty$  when  $X \not\prec 0$  or  $S \not\prec 0$ . This fact may not be seen by simply evaluating  $\det(XS)$ , which may be positive even when  $X \not\prec 0$  or  $S \not\prec 0$ . Therefore, the correct evaluation of  $\varphi_\mu$  requires to check that  $X \succ 0$  or  $S \succ 0$  (either by the computation of the minimum eigenvalues of  $X$  and  $S$  or by doing their Cholesky factorizations).

## 3.2 Implementation

### 3.2.1 Getting close to the central path

Implement algorithm 3.1.4 and try it on test case 2, with  $p = 50$ ,  $q = 50$ ,  $r = 50$ , and random data for the matrices  $B_i \in \mathbb{R}^{q \times r}$ , with  $i \in [0:p]$ . The data of this test case can be recovered by

$$[A, b, c, x0, y0] = \text{t2}(\text{dim});$$

where  $\text{dim}$  is the common dimension of  $p = q = r$  (here 50).

One could get an output close to the one below (actually, it depends on the random number generator, which may depend on the Matlab installation or version), in which the last column (“type”) gives the type of step realized:

- **c**: corrector step (normally a few steps at the beginning), in which case the **stepsize** is the one during the correction phase (linesearch),
- **pc**: predictor-corrector step, in which case the **stepsize** is the one during the prediction phase.

```
-----
SDO solver (version 0.1, December 2015)
-----
Description of the problem
Size of the matrices 100
Nb of constraints    51
-----
it   c'*x          gap    cp dist  stepsize type
1   0.0000000e+00  2.622e-01  2.5e+00  1.0000e+00 c
2  -1.6770204e-01  2.622e-01  2.3e-01  4.6241e-01 pc
3  -2.8823879e-01  1.410e-01  1.8e-03  3.6592e-01 pc
4  -4.5434024e-01  8.939e-02  1.2e-03  3.0170e-01 pc
5  -6.499992e-01  6.242e-02  1.9e-03  2.5513e-01 pc
6  -8.7034447e-01  4.650e-02  2.9e-03  2.2025e-01 pc
7  -1.1091389e+00  3.626e-02  4.4e-03  1.9557e-01 pc
8  -1.3599149e+00  2.917e-02  6.0e-03  1.8223e-01 pc
9  -1.6185093e+00  2.385e-02  6.6e-03  1.8167e-01 pc
10 -1.8841906e+00  1.952e-02  6.1e-03  1.9323e-01 pc
11 -2.1554869e+00  1.575e-02  5.7e-03  2.1642e-01 pc
12 -2.4278616e+00  1.234e-02  5.0e-03  2.5220e-01 pc
13 -2.6938177e+00  9.227e-03  4.3e-03  2.9890e-01 pc
14 -2.9396556e+00  6.469e-03  4.0e-03  3.5163e-01 pc
15 -3.1479103e+00  4.194e-03  3.7e-03  4.0628e-01 pc
16 -3.3066539e+00  2.490e-03  3.4e-03  4.6228e-01 pc
17 -3.4151253e+00  1.339e-03  3.0e-03  5.2168e-01 pc
18 -3.4814453e+00  6.405e-04  2.6e-03  5.8922e-01 pc
19 -3.5174530e+00  2.631e-04  2.3e-03  6.7596e-01 pc
20 -3.5344772e+00  8.525e-05  1.5e-03  7.9810e-01 pc
21 -3.5410028e+00  1.721e-05  5.7e-04  9.3334e-01 pc
22 -3.5425449e+00  1.147e-06  1.1e-04  9.9470e-01 pc
23 -3.5426544e+00  6.086e-09  3.9e-05  9.9900e-01 pc
-----
Solution found up to the required precision (1e-10)
. c'*x = -3.5426550e+00
. gap = 6.0855161e-12
-----
```

### 3.2.2 Test case 3: global minimization of a univariate polynomial

We consider the problem of finding the *global minimum of a univariate real polynomial*  $p \in \mathbb{R}[x]$ , in the variable  $x$ :

$$\min_{x \in \mathbb{R}} \left( p(x) := \sum_{\alpha \in [0:d]} p_{\alpha} x^{\alpha} \right), \quad (3.7)$$

where  $p_{\alpha} \in \mathbb{R}$  and  $d \in \mathbb{N}$  is the degree  $\deg p$  of the polynomial  $p$ . Problem (3.7) can be rewritten as an SDP problem. First, we write it as a problem with an infinite number of constraints:

$$\begin{cases} \max_{s \in \mathbb{R}} s \\ p(x) \geq s, \quad \forall x \in \mathbb{R}. \end{cases}$$



This infinite number of constraints “disappears” if we trivially express them in terms of membership to the cone  $\mathcal{P}$  of univariate nonnegative polynomials:

$$\begin{cases} \max_{s \in \mathbb{R}} s \\ p - s \in \mathcal{P}. \end{cases} \quad (3.8)$$

It turns out that membership to  $\mathcal{P}$  has an LMI representation (this is no longer true for multivariate polynomials, but there are bypasses [17, 8, 10, 1, 2, 11]). That fact is based on the following two properties: a univariate nonnegative polynomial can be written as a *sum of squares* (SOS) of polynomials (proposition 3.2.1, which is no longer true for multivariate polynomials) and an SOS polynomial can be represented thanks to a positive semidefinite matrix (proposition 3.2.2, which is still true for multivariate polynomials).

**Proposition 3.2.1 (nonnegative univariate polynomial)** *A univariate polynomial  $p \in \mathbb{R}[x]$  is nonnegative on  $\mathbb{R}$  if and only if it is of even degree, say  $2m$ , and reads  $p = q^2 + r^2$ , with  $q, r \in \mathbb{R}[x]$ ,  $\deg q = m$ , and  $\deg r \leq m - 1$ .*

PROOF. [18; VI 44] The given conditions are clearly sufficient. Let us show that they are necessary.

Being nonnegative on  $\mathbb{R}$  the polynomial is necessary of even degree, say  $2m$ , and the leading coefficient is positive. There is therefore no loss of generality in supposing that this coefficient is 1. Then the polynomial can be decomposed in  $m$  factors of the form

$$(x - a)^2 + b^2.$$

It is indeed the form of  $(x - r)(x - \bar{r})$  when  $r$  and  $\bar{r}$  complex conjugate roots  $a \pm ib$ . On the other hand, any real root has an even multiplicity (otherwise the polynomial would have positive and negative values around the root) and each double root is of the form above with  $b = 0$ .

The  $m$  factors of the form  $q^2 + r^2$  are then multiplied successively, by using the following formula

$$(q_j^2 + r_j^2)(q^2 + r^2) = (q_j q + r_j r)^2 + (q_j r - r_j q)^2 =: q_{j+1}^2 + r_{j+1}^2.$$

By induction, we see that  $\deg q_j = j$  and  $\deg r_j \leq j - 1$ . It is indeed the case for  $j = 1$  since  $\deg q_1 = 1$  and  $\deg r_1 \leq 0$ . Now, be  $r$  vanishing or not,  $\deg q_{j+1} = \deg q_j + 1 = j + 1$ . Finally, if  $r = 0$ ,  $\deg r_{j+1} = \deg r_j + 1 \leq j$  and, if  $r \neq 0$ ,  $\deg r_{j+1} \leq \max(\deg q_j, \deg r_j + 1) \leq j$ .  $\square$

**Proposition 3.2.2 (caractérisation of SOS polynomials)** *A polynomial  $p \in \mathbb{R}[x]$  of degree  $\leq 2m$  is a sum of  $r$  squares of polynomials if and only if there exists a matrix  $S \succcurlyeq 0$  of order  $m + 1$  and of rank  $\leq r$  such that  $p(x) = v_m(x)^\top S v_m(x)$ , where  $v_m(x)$  is the vector of monomials  $(1 \ x \ x^2 \ \dots \ x^m)^\top$ .*

PROOF. If  $p$  is the sum of  $r$  squares of polynomials of degree  $\leq m$ , one can find vectors  $s_i \in \mathbb{R}^{m+1}$  such that

$$p(x) = \sum_{i=1}^r (s_i^\top v_m(x))^2 = \sum_{i=1}^r v_m(x)^\top s_i s_i^\top v_m(x) = v_m(x)^\top S v_m(x),$$

where  $S := \sum_{i=1}^r s_i s_i^\top \succeq 0$  is of rank  $\leq r$ .

Conversely, suppose that  $p(x) = v_m(x)^\top S v_m(x)$ , with  $S \succeq 0$  of rank  $\leq r$ . The spectral decomposition of  $S = \sum_{i=1}^r s_i s_i^\top$  allows us to write

$$p(x) = \sum_{i=1}^r v_m(x)^\top s_i s_i^\top v_m(x) = \sum_{i=1}^r (s_i^\top v_m(x))^2,$$

showing that  $p$  is the sum of the squares of at most  $r$  polynomials.  $\square$

Using the above propositions and setting  $s = -t$ , problem 3.8 can be written

$$\begin{cases} \min_{(S,t) \in \mathcal{S}^{m+1} \times \mathbb{R}} t \\ p(x) + t = v_m(x)^\top S v_m(x), \quad \forall x \in \mathbb{R} \\ S \succeq 0. \end{cases} \quad (3.9)$$

Knowing  $p$ , the first constraint in (3.9) is an affine constraint on the unknown  $t \in \mathbb{R}$  and the unknown real coefficients of  $S$ , if we impose equality between the coefficients of the same monomials in both sides of the equality. Therefore, problem (3.9) is almost in the primal form of an SDO problem. The difference is the free variable  $t$ , which is only constrained by the affine constraint in (3.9), not by a nontrivial cone.

At least two techniques could be considered to solve this problem, which is not in the standard form of the primal SDO problem.

1. Redefine the basic problem to solve

$$\begin{cases} \inf_{(x,X) \in \mathbb{R}^p \times \mathcal{S}^n} \langle c, x \rangle + \langle C, X \rangle \\ A(x, X) = b \\ X \succeq 0, \end{cases}$$

with the “free” variables  $x \in \mathbb{R}^p$  and a linear map  $A : \mathbb{R}^p \times \mathcal{S}^n \rightarrow \mathbb{R}^m$ .

2. Write  $t = u - v$  and impose  $u \geq 0, v \geq 0$  by considering the problem in standard primal SDP form with  $X = \text{Diag}(S, u, v) \in \mathcal{S}^{m+3}$ :

$$\begin{cases} \min_{X \in \mathcal{S}^{m+3}} \langle C, X \rangle \\ A(X) = b \\ X \succeq 0, \end{cases} \quad (3.10)$$

where  $C = E^{m+2, m+2} - E^{m+3, m+3}$  (we have denoted by  $E^{ij}$  the matrix having 1 in position  $(i, j)$  and 0 elsewhere), the affine constraint  $A(X) = b$  contains those in the affine constraints in (3.9) and constraints imposing  $X_{1:m+1, m+2} = 0$  and  $X_{1:m+2, m+3} = 0$ .

We suggest using the second technique, which has the advantage of allowing us to use the so far developed code without modification.

### 3.3 Question

- 3.1.** *Getting the global minimizer of a univariate polynomial.* Write the dual of problem (3.9) and show that this one yields the global minimizer of  $p$ , provided this one is unique.

## 4 Starting from an infeasible point

### 4.1 Starting from an infeasible point

In all this section, we assume that an initial  $X_0 \in \mathcal{S}^n$  is known, which satisfies the equality constraint

$$A(X_0) = b. \quad (4.1)$$

A way of getting such a starting iterate is given in section 4.1.1. We then discuss the so-called *big M methods* for solving the SDO problem from a point  $z_0 = (X_0, y_0, S_0)$  satisfying (4.1) but not the conditions  $X_0 \succ 0$  and  $S_0 \succ 0$  that would make  $z_0$  strictly feasible. The contents of this section is adapted from [22; §6].

#### 4.1.1 Getting linear feasibility

If a matrix  $X_0$  satisfying (4.1) is not given by the user of the solver, such a matrix can easily be computed as follows. The first step is to check, using linear algebra techniques, whether  $b \in \mathcal{R}(A)$ . If this is not the case, the primal problem is infeasible and there is no reason to go further. Otherwise, (4.1) has a solution, which can be obtained by computing  $M \in \mathbb{R}^{n \times n}$  satisfying the constraint and symmetrizing it:

$$A(M) = b \quad \text{and} \quad X_0 = \frac{1}{2}(M + M^\top). \quad (4.2)$$

#### 4.1.2 Starting from a strictly primal feasible point

We assume in this section that a strictly feasible primal point  $X_0 \in \mathcal{F}_p^s$  is known, meaning that

$$A(X_0) = b \quad \text{and} \quad X_0 \succ 0,$$

and determine a modified SDO problem such that it is easy to maintain  $X_0$  strictly feasible and to determine a strictly feasible dual pair  $(y_0, S_0)$ .

Consider the following modified dual problem in  $(y, t, S) \in \mathbb{R}^m \times \mathbb{R} \times \mathcal{S}^n$ :

$$\begin{cases} \sup \langle b, y \rangle - Mt \\ A^*(y) - tI + S = C \\ S \succcurlyeq 0 \\ t \geq 0, \end{cases} \quad (4.3)$$

in which one tries to vanish the nonnegative scalar  $t$  by taking  $M$  “sufficiently” large, since if  $t = 0$ , the original dual problem  $(D)$  is recovered. We claim that

- this problem can be cast in the standard dual SDO problem, for which a strictly feasible point is easy to determine,
- its primal-dual solutions are the same as the original problem, provided  $M$  is chosen “sufficiently large” and primal solution set  $\text{Sol}(P)$  is bounded.

Let us see this.

One can cast problem (4.3) in the standard dual SDO problem in the variable  $(\tilde{y}, \tilde{S}) \in \mathbb{R}^{m+1} \times \mathcal{S}^{n+1}$ :

$$\begin{cases} \sup \langle \tilde{b}, \tilde{y} \rangle \\ \tilde{A}^*(\tilde{y}) + \tilde{S} = \tilde{C} \\ \tilde{S} \succcurlyeq 0. \end{cases} \quad (4.4)$$

This problem is identical to problem (4.3) provided the vector  $\tilde{y}$ , the vector  $\tilde{b}$ , the adjoint linear map  $\tilde{A}^* : \mathbb{R}^m \times \mathbb{R} \rightarrow \mathcal{S}^{n+1}$ , and the matrix  $\tilde{C} \in \mathcal{S}^{n+1}$  are defined by

$$\tilde{y} = \begin{pmatrix} y \\ t \end{pmatrix}, \quad \tilde{b} = \begin{pmatrix} b \\ -M \end{pmatrix}, \quad \tilde{A}^*(\tilde{y}) = \begin{pmatrix} A^*(y) - tI & 0 \\ 0 & -t \end{pmatrix}, \quad \text{and} \quad \tilde{C} = \begin{pmatrix} C & 0 \\ 0 & 0 \end{pmatrix}.$$

This implies that  $\tilde{S} = \text{Diag}(S, s) \in \mathcal{S}^{n+1}$  is block diagonal and that there is the additional constraint  $-t + s = 0$ . It is easy to get a strictly feasible point for this model by taking

$$\boxed{y_0 \text{ arbitrary,} \quad t_0 = s_0 > [\lambda_{\min}(C - A^*(y_0))]^-, \quad \text{and} \quad S_0 := C - A^*(y_0) + t_0 I,}$$

where  $\alpha^- = \max(0, -\alpha)$ .

We know that (4.4) is the dual of

$$\begin{cases} \inf \langle \tilde{C}, \tilde{X} \rangle \\ \tilde{A}(\tilde{X}) = \tilde{b} \\ \tilde{X} \succcurlyeq 0. \end{cases} \quad (4.5)$$

Now,  $\tilde{A} : \mathcal{S}^{n+1} \rightarrow \mathbb{R}^m \times \mathbb{R}$  is defined at  $\tilde{X}$  by

$$\tilde{A}(\tilde{X}) = \begin{pmatrix} A(X) \\ -\langle I, X \rangle - x \end{pmatrix}, \quad \text{where} \quad \tilde{X} = \begin{pmatrix} X & v \\ v^\top & x \end{pmatrix}.$$

Therefore, problem (4.5) reads

$$\begin{cases} \inf \langle C, X \rangle \\ A(X) = b \\ \langle I, X \rangle + x = M \\ X \succcurlyeq 0 \\ x \geq 0. \end{cases} \quad (4.6)$$

A strictly feasible point for (4.5)-(4.6) can be

$$\boxed{\text{the known } X_0, \quad v_0 = 0, \quad \text{and} \quad x_0 = M - \langle I, X_0 \rangle,}$$

provided  $M > \langle I, X_0 \rangle$ .

The proposed approach consists in solving (4.6)-(4.3), or equivalently (4.5)-(4.4) in standard form, from a strictly primal feasible-dual pair hoping that at the primal-dual solution  $(X, x, y, t, S)$  the scalar  $t$  vanishes, in which case problem  $(P)$  is actually solved. If  $t \neq 0$ , one increases  $M$  and solve problem (4.6)-(4.3) again. The process is stopped when  $M$  is considered as too large, which may occur if  $(D)$  is infeasible.

**Algorithm 4.1.1 (getting a solution from a strictly primal feasible point)** Let  $X \in \mathcal{F}_P^s$  and let  $M > \langle I, X \rangle$ . One iteration of the algorithm is as follows.

1. *Solve (4.6)-(4.3)* from a primal-dual strictly feasible point to get  $(X, x, y, t, S)$ .
2. *Successful stopping test.* If  $t = 0$  or, equivalently,  $x > 0$ , stop and declare that  $(X, y, S)$  is a primal-dual solution to  $(P)$ .
3. *Increase  $M$ .*
4. *Failure stopping test.* If  $M$  is too large, stop and declare that it is likely that either the dual is infeasible or the primal has an unbounded set of solutions.

Note that in none of the situations detected in step 4 of the algorithm, an interior point method can find a solution.

### 4.1.3 Starting from a strictly dual feasible point

We assume in this section that a strictly feasible dual pair  $(y_0, S_0) \in \mathcal{F}_D^s$  is known, meaning that

$$A^*(y_0) + S_0 = C \quad \text{and} \quad S_0 \succ 0,$$

and determine a modified SDO problem such that it is easy to maintain  $(y_0, S_0)$  strictly feasible and to determine a strictly feasible primal point  $X_0$ .

Consider the following modified primal problem in  $(X, x_2) \in \mathcal{S}^n \times \mathbb{R}$ :

$$\left\{ \begin{array}{l} \inf \langle C, X \rangle + x_2(M_2 - \langle C, I \rangle) \\ A(X - x_2 I) = b \\ X \succcurlyeq 0 \\ x_2 \geq 0 \end{array} \right. \quad \text{or} \quad \left\{ \begin{array}{l} \inf \langle C, X \rangle + M_2 x_2 \\ A(X) = b \\ X + x_2 I \succcurlyeq 0 \\ x_2 \geq 0, \end{array} \right. \quad (4.7)$$

in which one tries to vanish the nonnegative scalar  $x_2$  by taking the positive number  $M_2$  “sufficiently” large, since if  $x_2 = 0$ , the original primal problem  $(P)$  is recovered. By considering the problem in the right hand side in (4.7), one sees that it is desired and enough to have  $M_2 > 0$  (not  $M_2 > \langle C, I \rangle$  as the problem in the left hand side may suggest). We claim that

- this problem can be cast in the standard primal SDO problem, for which a strictly feasible point is easy to define,
- its primal-dual solutions are the same as the original problem, provided  $M_2$  is chosen “sufficiently large” and dual solution set  $\text{Sol}(D)$  is bounded.

Let us see this.

One can cast the problem in the left hand side of (4.7) in the standard primal SDO problem in the variable  $\tilde{X} \in \mathcal{S}^{n+1}$ :

$$\begin{cases} \inf \langle \tilde{C}, \tilde{X} \rangle \\ \tilde{A}(\tilde{X}) = \tilde{b} \\ \tilde{X} \succcurlyeq 0. \end{cases} \quad (4.8)$$

This problem is identical to problem (4.7) provided the matrix  $\tilde{C}$ , the linear map  $\tilde{A} : \mathcal{S}^{n+1} \rightarrow \mathbb{R}^{m+n}$ , and the vector  $\tilde{b}$  are defined by

$$\tilde{C} = \begin{pmatrix} C & 0 \\ 0 & M_2 - \langle C, I \rangle \end{pmatrix}, \quad \tilde{A}(\tilde{X}) = \begin{pmatrix} A(X - x_2 I) \\ v \end{pmatrix}, \quad \text{and} \quad \tilde{b} = \begin{pmatrix} b \\ 0 \end{pmatrix},$$

where we have denoted

$$\tilde{X} = \begin{pmatrix} X & v \\ v^\top & x_2 \end{pmatrix}.$$

Since  $v$  must vanish, the condition  $\tilde{X} \succcurlyeq 0$  in (4.8) is therefore equivalent to the constraints  $X \succcurlyeq 0$  and  $x_2 \geq 0$  in (4.7). A strictly feasible point for (4.8) is given by

$$\tilde{X}_0 := \begin{pmatrix} X_0 + x_{2,0} I & 0 \\ 0 & x_{2,0} \end{pmatrix},$$

where

$$\begin{aligned} X_0 \in \mathcal{S}^n \text{ is an arbitrary solution to } A(X_0) = b, \\ x_{2,0} > [\lambda_{\min}(X_0)]^-. \end{aligned}$$

We know that (4.8) has for dual

$$\begin{cases} \sup \langle \tilde{b}, \tilde{y} \rangle \\ \tilde{A}^*(\tilde{y}) + \tilde{S} = \tilde{C} \\ \tilde{S} \succcurlyeq 0. \end{cases} \quad (4.9)$$

Now,  $\tilde{A}^* : \mathbb{R}^{m+n} \rightarrow \mathcal{S}^{n+1}$  is defined at  $\tilde{X}$  by

$$\tilde{A}^*(\tilde{y}) = \begin{pmatrix} A^*(y) & \frac{1}{2}z \\ \frac{1}{2}z^\top & -\langle I, A^*(y) \rangle \end{pmatrix}, \quad \text{where} \quad \tilde{y} = \begin{pmatrix} y \\ z \end{pmatrix}.$$

Writing

$$\tilde{S} := \begin{pmatrix} S & w \\ w^\top & s_2 \end{pmatrix},$$

we see that problem (4.9) reads

$$\begin{cases} \sup \langle b, y \rangle \\ A^*(y) + S = C \\ \frac{1}{2}z + w = 0 \\ \langle I, C - A^*(y) \rangle + s_2 = M_2 \\ \tilde{S} \succcurlyeq 0. \end{cases}$$

The constraint  $\tilde{S} \succcurlyeq 0$  includes  $S \succcurlyeq 0$  and  $s_2 \geq 0$  and is equivalent to these two constraints if  $w$  is set to zero, which is permitted by the second constraint since  $z$  is arbitrary. The problem is then simplified by taking  $z = w = 0$ . Eliminating also  $s_2$ , the dual problem (4.9) becomes

$$\begin{cases} \sup \langle b, y \rangle \\ A^*(y) + S = C \\ S \succcurlyeq 0 \\ \langle I, S \rangle \leq M_2. \end{cases} \quad (4.10)$$

It can be shown that the variable  $x_2$  in (4.7) is a dual variable associated with the constraint  $\langle I, S \rangle \leq M_2$  in (4.10). A strictly feasible dual pair for the modified dual problems (4.9) is deduced from the known strictly feasible dual pair  $(y_o, S_0)$  for  $(D)$  by

$$\tilde{y}_0 = \begin{pmatrix} y_o \\ 0 \end{pmatrix} \in \mathbb{R}^m \times \mathbb{R}^n \quad \text{and} \quad \tilde{S}_0 = \begin{pmatrix} S_0 & 0 \\ 0 & s_{2,0} \end{pmatrix}$$

in which one sets successively

$$\begin{aligned} (y_o, S_0) \in \mathbb{R}^m \times \mathcal{S}_{++}^n \text{ is given,} \\ M_2 > \langle I, S_0 \rangle, \\ s_{2,0} := M_2 - \langle I, S_0 \rangle, \end{aligned}$$

where  $\alpha^- = \max(0, -\alpha)$ . Note that this setting ensures that  $M_2 > 0$  (since  $S_0 \succ 0$ ), as desired.

The proposed approach consists in solving (4.7)-(4.10) from a strictly primal feasible-dual pair (instead of the original problem  $(P)$ ) hoping that at the solution  $(X, x_2, y, S)$  the scalar  $x_2$  vanishes, in which case problem  $(P)$  is actually solved. If  $x_2 \neq 0$ , one increases  $M_2$  and solve problem (4.7) again. The process is stopped when  $M_2$  is considered as too large, which may occur if  $(P)$  is infeasible.

**Algorithm 4.1.2 (getting a solution from a strictly dual feasible point)** Let  $(y, S) \in \mathcal{F}_D^s$  and let  $M_2 > \langle I, S \rangle$ . One iteration of the algorithm is as follows.

1. *Solve (4.7)-(4.10) from a primal-dual strictly feasible point to get  $(X, x_2, y, S)$ .*
2. *Successful stopping test.* If  $x_2 = 0$  or, equivalently,  $\langle S, I \rangle < M_2$ , stop and declare that  $(X, y, S)$  is a primal-dual solution to  $(P)$ .
3. *Increase  $M_2$ .*
4. *Failure stopping test.* If  $M_2$  is too large, stop and declare that it is likely that either the primal is infeasible or the dual has an unbounded set of solutions.

Note that in none of the situations detected in step 4 of the algorithm, an interior point method can find a solution.



#### 4.1.4 Starting without a strictly feasible point

If no strictly feasible primal and/or dual point is known, then one considers the modified primal problem, obtained by combining the ideas of sections 4.1.2 and 4.1.3:

$$\left\{ \begin{array}{l} \inf \langle C, X \rangle + x_2(M_2 - \langle C, I \rangle) \\ A(X - x_2I) = b \\ \langle I, X \rangle + x_1 = M_1 \\ X \succcurlyeq 0 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{array} \right. \quad \text{or} \quad \left\{ \begin{array}{l} \inf \langle C, X \rangle + M_2x_2 \\ A(X) = b \\ \langle I, X \rangle + x_1 + nx_2 = M_1 \\ X + x_2I \succcurlyeq 0 \\ x_1 \geq 0, \\ x_2 \geq 0, \end{array} \right. \quad (4.11)$$

in which both  $M_1$  and  $M_2$  are taken “sufficiently” large (initial values are given below). By considering the problem in the right hand side in (4.11), one sees that it is desired and enough to have  $M_2 > 0$  (not  $M_2 > \langle C, I \rangle$  as the problem in the left hand side may suggest).

The problem can be cast in the standard primal SDO problem in the variable  $\tilde{X} \in \mathcal{S}^{n+2}$ :

$$\left\{ \begin{array}{l} \inf \langle \tilde{C}, \tilde{X} \rangle \\ \tilde{A}(\tilde{X}) = \tilde{b} \\ \tilde{X} \succcurlyeq 0 \end{array} \right. . \quad (4.12)$$

The equivalence between (4.11) and (4.12) can be viewed by decomposing the new variable  $\tilde{X}$  into

$$\tilde{X} = \left( \begin{array}{cc|c} X & v_1 & v_2 \\ \bar{v}_1^\top & x_1 & \\ \hline & \bar{v}_2^\top & x_2 \end{array} \right)$$

and by defining the matrix  $\tilde{C}$ , the linear map  $\tilde{A} : \mathcal{S}^{n+2} \rightarrow \mathbb{R}^{m+4n+3}$ , and the vector  $\tilde{b} \in \mathbb{R}^{m+4n+3}$  by

$$\tilde{C} = \left( \begin{array}{cc|c} C & 0 & 0 \\ 0 & 0 & \\ \hline & & M_2 - \langle C, I \rangle \end{array} \right), \quad \tilde{A}(\tilde{X}) = \begin{pmatrix} A(X - x_2I) \\ -\langle I, X \rangle - x_1 \\ v_1 \\ \bar{v}_1 \\ v_2 \\ \bar{v}_2 \end{pmatrix}, \quad \text{and} \quad \tilde{b} = \begin{pmatrix} b \\ -M_1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

A strictly feasible point for the modified primal problems (4.11) or (4.12) is

$$\tilde{X}_0 := \left( \begin{array}{cc|c} X_0 + x_{2,0}I & 0 & 0 \\ 0 & x_{1,0} & \\ \hline & & x_{2,0} \end{array} \right)$$

where

$$\begin{aligned} X_0 \in \mathcal{S}^n \text{ is an arbitrary solution to } A(X_0) &= b, \\ x_{2,0} &> [\lambda_{\min}(X_0)]^-, \\ M_1 &> \langle I, X_0 \rangle + nx_{2,0}, \\ x_{1,0} &:= M_1 - \langle I, X_0 \rangle - nx_{2,0}, \end{aligned}$$

where  $\alpha^- = \max(0, -\alpha)$ . Note that  $\langle I, X_0 \rangle = \text{tr}(X_0) = \sum_i \lambda_i(X_0)$  and  $x_{2,0} > -\lambda_{\min}(X_0)$ , so that  $M_1 > 0$  and the factor of  $t$  in the objective of (4.14) is positive, a desired property.

We know that the dual of (4.12) reads

$$\begin{cases} \sup \langle \tilde{b}, \tilde{y} \rangle \\ \tilde{A}^*(\tilde{y}) + \tilde{S} = \tilde{C} \\ \tilde{S} \succcurlyeq 0. \end{cases} \quad (4.13)$$

where  $\tilde{A}^* : \mathbb{R}^{m+4n+3} \rightarrow \mathcal{S}^{n+2}$  is defined at

$$\tilde{y} = \begin{pmatrix} y \\ t \\ z_1 \\ \tilde{z}_1 \\ z_2 \\ \tilde{z}_2 \end{pmatrix} \in \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n+1} \times \mathbb{R}^{n+1}$$

by

$$\tilde{A}^*(\tilde{y}) = \left( \begin{array}{cc|c} A^*(y) - tI & z_1 & z_2 \\ \hline \tilde{z}_1^\top & -t & \\ \hline \tilde{z}_2^\top & & -\langle I, A^*(y) \rangle \end{array} \right).$$

Now decomposing  $\tilde{S}$  like  $\tilde{X}$ , namely

$$\tilde{S} = \left( \begin{array}{cc|c} S & w_1 & w_2 \\ \hline w_1^\top & s_1 & \\ \hline w_2^\top & & s_2 \end{array} \right),$$

problem (4.13) becomes

$$\begin{cases} \sup \langle b, y \rangle - M_1 t \\ A^*(y) - tI + S = C \\ z_1 = \tilde{z}_2 = -w_1 \\ -t + s_1 = 0 \\ z_2 = \tilde{z}_2 = -w_2 \\ -\langle I, A^*(y) \rangle + s_2 = M_2 - \langle C, I \rangle \\ \tilde{S} \succcurlyeq 0. \end{cases}$$

The constraint  $\tilde{S} \succcurlyeq 0$  includes  $S \succcurlyeq 0$ ,  $s_1 \geq 0$ , and  $s_2 \geq 0$  and amounts to these constraints if  $w_1$  and  $w_2$  are set to zero, which is allowed since  $z_1$ ,  $\tilde{z}_1$ ,  $z_2$ , and  $\tilde{z}_2$  are not constrained. The problem is then simplified by taking  $w_1 = 0$  and  $w_2 = 0$ . Eliminating also  $s_1$  and  $s_2$ , the dual problem becomes the following problem in  $(y, t, S)$ :

$$\begin{cases} \sup \langle b, y \rangle - M_1 t \\ A^*(y) - tI + S = C \\ \langle I, S \rangle - nt \leq M_2 \\ S \succcurlyeq 0 \\ t \geq 0. \end{cases} \quad (4.14)$$

A strictly feasible point for the modified dual problems (4.13) or (4.14) is

$$\tilde{y}_0 = \begin{pmatrix} y_0 \\ t_0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \in \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n+1} \times \mathbb{R}^{n+1} \quad \text{and} \quad \tilde{S}_0 = \left( \begin{array}{cc|c} S_0 & 0 & 0 \\ 0 & s_{1,0} & \\ \hline & 0 & s_{2,0} \end{array} \right),$$

in which one sets successively

$$\begin{array}{l} y_0 \text{ arbitrary,} \\ t_0 = s_{1,0} > [\lambda_{\min}(C - A^*(y_0))]^-, \\ S_0 := C - A^*(y_0) + t_0 I, \\ M_2 > [\langle I, S_0 \rangle - nt_0]^+, \\ s_{2,0} := M_2 - \langle I, S_0 \rangle + nt_0, \end{array} \quad (4.15)$$

where  $\alpha^- = \max(0, -\alpha)$ . The setting of  $M_2$  above also ensures its positivity.

The goal of the following algorithm is to solve a sequence of primal dual problems (4.12)-(4.13) or (4.11)-(4.14) by increasing  $M_1$  and  $M_2$  at each loop in order to have  $t = x_2 = 0$ , which guarantees that the original primal-dual SDO problem has been solved. The form of the problems (4.11) and (4.14) clearly indicates that one must increase  $M_1$  if  $t > 0$  and one must increase  $M_2$  when  $x_2 > 0$ .

**Algorithm 4.1.3 (getting a solution from a linear feasible point)** One loop of the algorithm is as follows.

1. *Solve (4.12)-(4.13) or (4.11)-(4.14) from a primal-dual strictly feasible point to get  $(X, x_1, x_2, y, t, S, s_1, s_2)$ .*
2. *Successful stopping test.* If  $x_2 = 0$  and  $t = 0$ , stop and declare that  $(X, y, S)$  is a primal-dual solution to  $(P)$ .
3. *Increase  $M_1$  if  $t > 0$ .*
4. *Increase  $M_2$  if  $x_2 > 0$ .*
5. *Failure on dual infeasibility.* If  $M_1$  is too large, stop and declare that it is likely that *either the dual is infeasible or the primal has an unbounded set of solutions.*
6. *Failure on primal infeasibility.* If  $M_2$  is too large, stop and declare that it is likely that *either the primal is infeasible or the dual has an unbounded set of solutions.*

Note that in none of the situations detected in steps 5 and 6 of the algorithm, an interior point method can find a solution.

## 4.2 Implementation

### 4.2.1 Implementation

Implement algorithm 4.1.3 and solve/diagnose the test cases of sections 4.2.3, 4.2.4, and 4.2.5.

### 4.2.2 Recommendations

- To develop the solver, a good idea is to test it on the easy test cases 1 and 2, in which the initial values of  $X$  and  $y$  has been discarded. Once the solver works correctly on these test cases, you may want to test it on the more special problems of section ??.
- The linear operator  $\tilde{A}$  does not depend on  $M_1$  and  $M_2$  and can be set outside the loop described in algorithm 4.1.3.
- Check whether the linear constraints  $\tilde{A}(\tilde{X}_0) = \tilde{b}$  and  $\tilde{A}^*(\tilde{y}) + \tilde{S} = \tilde{C}$  are satisfied at the beginning of *each cycle*, after the setting of  $\tilde{A}$ ,  $\tilde{X}_0$ , and  $\tilde{b}$ .

### 4.2.3 Test case 4a

If possible, solve with algorithm 4.1.3 the problem

$$\left\{ \begin{array}{l} \inf x_1 \\ \begin{pmatrix} 0 & x_1 & 0 \\ x_1 & x_2 & 0 \\ 0 & 0 & x_1 + 1 \end{pmatrix} \succcurlyeq 0. \end{array} \right.$$

What is your diagnosis?

### 4.2.4 Test case 4b

If possible, solve with algorithm 4.1.3 the primal SDO problem written in standard form with the following data:

$$C = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad A_1 = \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix}, \quad \text{and} \quad b = \begin{pmatrix} -1 \\ 0 \end{pmatrix}.$$

What is your diagnosis?

### 4.2.5 Test case 4c

If possible, solve with algorithm 4.1.3 the primal SDO problem written in standard form with the following data:

$$C = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \text{and} \quad b = \begin{pmatrix} 0 \\ 2 \end{pmatrix}.$$

What is your diagnosis?

### 4.3 Questions

- 4.1. *Getting a linear feasible primal point.* Justify the correctness of the procedure (4.2) to get a matrix  $X_0 \in \mathcal{S}^n$  satisfying (4.1).
- 4.2. *Failure diagnosis.* Explain the *failure diagnosis* of steps 5 and 6 of algorithm 4.1.3.

## 5 Shor relaxation of some small OPF problems

### 5.1 Relaxation of an OPF problem

One of the emblematic instance of the so-called Optimal Power Flow (OPF) problem, in alternating current, consists in minimizing the Joule heating losses in an electricity transmission network, while satisfying the electricity demand, by determining appropriately the powers of the generators installed at given buses (or nodes of the network). This optimization problem is NP hard, implying that there is (presently) no algorithm to find the (global) solution in polynomial time (i.e., in a reasonable time when the network size is like the one in most countries or union of countries), while there are several good reasons for being only interested in the global solutions.

#### 5.1.1 QCQP formulation

Structurally, one can view (or reduce) the problem to a nonconvex quadratic optimization problem with nonconvex quadratic constraints (sometimes abbreviated by the acronym QCQP for Quadratically Constrained Quadratic Programming). Recalling that any polynomial optimization problem can be written that way [19, 4, 13], one understands the potential difficulty of such a problem.

A QCQP representation of an OPF instance can be obtained by the Matlab function `qcqp_opf` [6], which should be available soon in **Matpower** (a Matlab package for simulating power systems). This function writes a specified OPF instance as a QCQP problem in complex numbers:

$$\begin{cases} \inf_z z^H C z \\ z^H A_k z = a_k, & \text{for } k \in [1 : m], \\ z^H B_k z \leq b_k, & \text{for } k \in [1 : p], \end{cases} \quad (5.1)$$

where  $z \in \mathbb{C}^n$  is a complex vector,  $C$ ,  $A_k$ , and  $B_k \in \mathbb{C}^{n \times n}$  are Hermitian matrices<sup>1</sup>,  $a$  and  $b$  are real vectors, and the exponent  $\cdot^H$  is used to denote the transposed conjugate. In this representation,  $n$  is the number of buses (or nodes of the network).

To solve (5.1) (more precisely a relaxation of it) by an SDO solver in real numbers, the first task is to rewrite the problem in real numbers. For  $M \in \mathbb{C}^{n \times n}$ , we write  $M = \Re(M) + i\Im(M)$ , where  $\Re(M)$  and  $\Im(M) \in \mathbb{R}^{n \times n}$ , and  $i \in \mathbb{C}$  the pure imaginary number ( $i^2 = -1$ ). Similarly, a vector  $z \in \mathbb{C}^n$  is decomposed in  $z = \Re(z) + i\Im(z)$ , with  $\Re(z)$  and  $\Im(z) \in \mathbb{R}^n$ . It is easy to see that

---

<sup>1</sup> Recall that  $M \in \mathbb{C}^{n \times n}$  is Hermitian if  $M^H = M$ .

$$M \text{ is Hermitian} \iff \begin{cases} \Re(M) \text{ is symmetric,} \\ \Im(M) \text{ is skew symmetric.} \end{cases}$$

With this notation and a Hermitian matrix  $M$ , there holds

$$z^H M z = \tilde{z}^T \tilde{M} \tilde{z},$$

where

$$\tilde{z} := \begin{pmatrix} \Re(z) \\ \Im(z) \end{pmatrix} \in \mathbb{R}^{2n} \quad \text{and} \quad \tilde{M} := \begin{pmatrix} \Re(M) & -\Im(M) \\ \Im(M) & \Re(M) \end{pmatrix} \in \mathbb{R}^{(2n) \times (2n)}.$$

Note that  $\tilde{M}$  is symmetric. Then, problem (5.1) becomes the QCQP in real numbers

$$\begin{cases} \inf_{\tilde{z}} \tilde{z}^T \tilde{C} \tilde{z} \\ \tilde{z}^T \tilde{A}_k \tilde{z} = a_k, & \text{for } k \in [1:m], \\ \tilde{z}^T \tilde{B}_k \tilde{z} \leq b_k, & \text{for } k \in [1:p]. \end{cases} \quad (5.2)$$

The dimension of this problem is twice the one of (5.1).

### 5.1.2 Shor relaxation

The second task is to define an SDO relaxation of (5.2). Denoting by  $\langle \cdot, \cdot \rangle$  the scalar product on the space of symmetric matrices, problem (5.2) reads

$$\begin{cases} \inf_{\tilde{z}} \langle \tilde{C}, \tilde{z} \tilde{z}^T \rangle \\ \langle \tilde{A}_k, \tilde{z} \tilde{z}^T \rangle = a_k, & \text{for } k \in [1:m], \\ \langle \tilde{B}_k, \tilde{z} \tilde{z}^T \rangle \leq b_k, & \text{for } k \in [1:p] \end{cases}$$

or

$$\begin{cases} \inf_{\tilde{X}} \langle \tilde{C}, \tilde{X} \rangle \\ \langle \tilde{A}_k, \tilde{X} \rangle = a_k, & \text{for } k \in [1:m], \\ \langle \tilde{B}_k, \tilde{X} \rangle \leq b_k, & \text{for } k \in [1:p] \\ \tilde{X} \succcurlyeq 0 \\ \text{rank}(\tilde{X}) \leq 1. \end{cases}$$

The rank constraint is very annoying (it takes integer values, hence is discontinuous). The Shor relaxation of the problem drops that constraint and therefore reads

$$\begin{cases} \inf_{\tilde{X}} \langle \tilde{C}, \tilde{X} \rangle \\ \langle \tilde{A}_k, \tilde{X} \rangle = a_k, & \text{for } k \in [1:m], \\ \langle \tilde{B}_k, \tilde{X} \rangle \leq b_k, & \text{for } k \in [1:p] \\ \tilde{X} \succcurlyeq 0. \end{cases}$$

The obtained relaxation is a convex problem, since the objective is convex and the feasible set is also convex (it is the intersection of the cone of positive semidefinite matrices and a convex polyhedron).

We still have to write the relaxed problem in the standard primal SDO form, which looks like the closest to the present one. This can be realized by enlarging  $\tilde{X}$  in

$$X := \begin{pmatrix} \tilde{X} & 0 \\ 0 & D \end{pmatrix},$$

where  $D$  is a diagonal matrix made of nonnegative slack variables:

$$\begin{cases} \inf_{(\tilde{X}, D)} \langle \tilde{C}, \tilde{X} \rangle \\ \langle \tilde{A}_k, \tilde{X} \rangle = a_k, & \text{for } k \in [1:m], \\ \langle \tilde{B}_k, \tilde{X} \rangle + D_{kk} = b_k, & \text{for } k \in [1:p] \\ \text{Diag}(\tilde{X}, D) \succeq 0. \end{cases} \quad (5.3)$$

This problem can be written in the standard primal SDO form, in which the affine equality constraint is formed of those in (5.3) and the one that imposes vanishment of the elements of  $X$  above and in the left hand side of  $D$ .

## 5.2 Test cases

The real SDO relaxation of the problem, namely the standard primal SDO form deduced from (5.3), is available by

```
[A,b,c,x0,y0] = t5(testcase);
```

where `testcase` is a string giving the name of a **Matpower** test case (see the table below for a short list),  $A$ ,  $b$ , and  $c$  stand for the data  $A$ ,  $b$ , and  $C$  of the standard primal SDO problem, while  $x0$  and  $y0$  are empty variables (no initial primal-dual point is suggested).

Table 5.1 lists the **Matpower** test cases that have been implemented and gives some

Problem name	$n$	$m$	$p$	$v_{\text{mips}}$	$v_{\text{Shor}}$
caseWB2	2	2	8	–	8.9082
caseWB5	5	6	18	13.780	9.9095
case9	9	12	30	373.83	373.83

**Table 5.1.** Some **Matpower** test cases.

of their features:

- the name of the problem (to put between simple quotes as argument `testcase` of the function `t5` above),
- the dimensions  $n$ ,  $m$ ,  $p$  of (5.1) (the number  $n$  is also the number of buses of the network),
- the optimal value  $v_{\text{mips}}$  obtained by the Matlab local interior point solver `mips` (“–” means that the solver fails),
- and the optimal value  $v_{\text{Shor}}$  obtained by the Shor relaxation.

The optimal value of the problem is necessarily in the interval  $[v_{\text{Shor}}, v_{\text{mips}}]$ . Therefore the optimal value is obtained by the local solver `mips` in `case9`, but the situation is uncertain in `caseWB5`.



The Shor relaxation of a QCQP can be viewed as the Lasserre [8] (or moment-sos) relaxation of degree one [9]. For the OPF problem, when the degree one relaxation of its QCQP formulation is inexact (i.e., it does not provide the solution to the original QCQP), it is often the case that the degree 2 moment-sos relaxation is exact [7].

## References

- [1] M.F. Anjos, J.B. Lasserre (2012). *Introduction to semidefinite, conic and polynomial optimization*, volume 166 of *International Series in Operations Research & Management Science*, chapter 1. Springer. 24
- [2] G. Blekherman, P.A. Parrilo, R.R. Thomas (2013). *Semidefinite Optimization and Convex Algebraic Geometry*. MOS-SIAM Series on Optimization. SIAM and MPS, Philadelphia. [doi]. 24
- [3] E. de Klerk (2002). *Aspects of Semidefinite Programming - Interior Point Algorithms and Selected Applications*. Kluwer Academic Publishers, Dordrecht. [doi]. 8, 14, 15, 19, 21
- [4] C. Ferrier (1998). Hilbert’s 17th problem and best dual bounds in quadratic minimization. *Cybernetics and Systems Analysis*, 34, 696–709. [doi]. 37
- [5] J. Jiang (1998). A long step primal-dual path following method for semidefinite programming. *Operations Research Letters*, 23(1-2), 53–62. 14
- [6] C. Jozs, S. Fliscounakis, J. Maeght, P. Panciatici (2015). The `Matlab` function `qcqp_opf`. Piece of software, Réseau de Transport d’Electricité, France. Personal communication. 37
- [7] C. Jozs, J. Maeght, P. Panciatici, J.Ch. Gilbert (2015). Application of the moment-SOS approach to global optimization of the OPF problem. *IEEE Transactions on Power Systems*, 30(1), 463–470. [doi]. 40
- [8] J.B. Lasserre (2001). Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11, 796–817. [doi]. 24, 40
- [9] J.B. Lasserre (2001). Convergent LMI relaxations for nonconvex quadratic programs. Technical report. Graduate seminar at MIT, fall 2001. [http://www.mit.edu/~6.454/www\\_fall\\_2001/cmccaram/lasserre\\_1.pdf](http://www.mit.edu/~6.454/www_fall_2001/cmccaram/lasserre_1.pdf). 40
- [10] J.B. Lasserre (2010). *Moments Positive Polynomials and Their Applications*. Imperial College Press Optimization Series 1. Imperial College Press. 24
- [11] J.B. Lasserre (2015). *An Introduction to Polynomial and Semi-Algebraic Optimization*. Cambridge Texts in Applied Mathematics. Cambridge University Press. 24
- [12] C.-J. Lin, R. Saigal (1995). A predictor-corrector method for semi-definite linear programming. Technical Report 95-20, Department of Industrial & Operations Engineering, The University of Michigan, Ann Arbor, Michigan 48109-2117, USA. 7
- [13] R. Madani, G. Fazelnia, J. Lavaei (2014). Rank-2 matrix solution for semidefinite relaxations of arbitrary polynomial optimization problems. Technical report. 37
- [14] S. Mizuno, M.J. Todd, Y. Ye (1993). On adaptive-step primal-dual interior-point algorithms for linear programming. *Mathematics of Operations Research*, 18(4), 964–981. [doi]. 14
- [15] Y.E. Nesterov, M.J. Todd (1997). Self-scaled barriers and interior-point methods for convex programming. *Mathematics of Operations Research*, 22(1), 1–42. [doi]. 7
- [16] Y.E. Nesterov, M.J. Todd (1998). Primal-dual interior-point methods for self-scaled cones. *SIAM Journal on Optimization*, 8(2), 324–364. [doi]. 7
- [17] P.A. Parrilo (2000). On a decomposition for multivariable forms via LMI methods. In *Proceedings of the American Control Conference*. 24

- [18] G. Pólya, G. Szegő (1976). *Problems and Theorems in Analysis II*. Springer-Verlag, Berlin. [24](#)
- [19] N.Z. Shor (1987). Class of global minimum bounds of polynomial functions. *Kibernetika*, 6, 9–11. [English translation: *Cybernetics*, 23 (1987) 731-734]. [37](#)
- [20] J.F. Sturm, S. Zhang (1999). Symmetric primal-dual path-following algorithms for semidefinite programming. *Applied Numerical Mathematics*, 29(3), 301–315. [\[doi\]](#). [7](#)
- [21] M.J. Todd, K.-C. Toh, R.H. Tütüncü (1998). On the Nesterov-Todd direction in semidefinite programming. *SIAM Journal on Optimization*, 8, 769–796. [\[doi\]](#). [7](#), [9](#)
- [22] L. Vandenberghe, S. Boyd (1996). Semidefinite programming. *SIAM Review*, 38, 49–95. [\[doi\]](#). [16](#), [27](#)