Regularization methods for semidefinite programming*

Jérôme Malick† Janez Povh‡ Franz Rendl§
Angelika Wiegele§

September 1, 2008

Abstract

We introduce a new class of algorithms for solving linear semidefinite programming (SDP) problems. Our approach is based on classical tools from convex optimization such as quadratic regularization and augmented Lagrangian techniques. We study the theoretical properties and we show that practical implementations behave very well on some instances of SDP having a large number of constraints. We also show that the "boundary point method" from [PRW06] is an instance of this class.

Key words: semidefinite programming, regularization methods, augmented Lagrangian method, large scale semidefinite problems

1 Introduction

1.1 Motivations

Semidefinite programming (SDP) has been a very active research area in optimization for more than a decade. This activity was motivated by important applications, especially in combinatorial optimization and in control theory. We refer to the reference book [WSV00] for theory and applications.

The key object in semidefinite programming is the set of positive semidefinite matrices, denoted by $\mathcal{S}_n^+$, which constitutes a closed convex cone in $\mathcal{S}_n$, the space of n-by-n symmetric matrices. Denoting by $\langle X, Y\rangle = \text{trace}(XY)$ the standard inner product in $\mathcal{S}_n$, one writes the standard form of a (linear) semidefinite program as

$$\left\{ \begin{array}{ll} \min & \langle C, X\rangle \\ & \mathcal{A}X = b,\ X \succeq 0, \end{array} \right. \tag{1}$$

*Supported in part by the EU project Algorithmic Discrete Optimization (ADONET), MRTN-CT-2003-504438.

†CNRS, Lab. J. Kuntzmann, University of Grenoble, France
‡University of Maribor, Faculty of logistics, Slovenia
§Institut für Mathematik, Alpen-Adria-Universität Klagenfurt, Austria

where $b \in \mathbb{R}^m$, $\mathcal{A}\colon \mathcal{S}_n \to \mathbb{R}^m$ is a linear operator and $X \succeq 0$ stands for $X \in \mathcal{S}_n^+$. The problem dual to (1) is

$$\left\{ \begin{array}{ll} \max & b^\top y \\ & C - \mathcal{A}^* y = Z, \ Z \succeq 0, \end{array} \right. \tag{2}$$

where the adjoint $\mathcal{A}^*\colon \mathbb{R}^m \to \mathcal{S}_n$ satisfies

$$\forall y \in \mathbb{R}^m, \quad \forall X \in \mathcal{S}_n, \qquad y^\top \mathcal{A} X = \langle \mathcal{A}^* y, X \rangle. \tag{3}$$

The matrix $X \in \mathcal{S}_n$ is called the primal variable and the pair $(y, Z) \in \mathbb{R}^m \times \mathcal{S}_n$ forms the dual variable.

The success of semidefinite programming was also spurred by the development of efficient algorithms to solve the pair of dual problems (1),(2). A pleasant situation to develop algorithms is when strong duality holds, for instance under the classical technical Slater constraint qualification: if we assume that both primal and dual problems satisfy the Slater condition (meaning that there exists a positive definite matrix X with $\mathcal{A}X = b$ and a vector y such that $C - \mathcal{A}^* y$ is positive definite), then there is no duality gap and there exist primal and dual optimal solutions; moreover (X, y, Z) is optimal if and only if

$$\left\{ \begin{array}{llll} \mathcal{A}X = b, & C - \mathcal{A}^* y = Z, \\ X \succeq 0, & Z \succeq 0, & \langle X, Z \rangle = 0. \end{array} \right. \tag{4}$$

It is widely accepted that interior-point based approaches are among the most efficient methods for solving general SDP problems. The primal-dual interior-point path-following methods are based on solving the optimality conditions (4) with $\langle X, Z \rangle = 0$ replaced by a parameterized matrix equation $ZX = \mu I$, or some variant of this equation. As $\mu > 0$ approaches 0, optimality holds with increasing accuracy. The key operation consists in applying Newton's method to these equations, resulting in the following linear system for $\Delta X, \Delta y, \Delta Z$:

$$\begin{array}{rcl} \mathcal{A}(Z^{-1}\mathcal{A}^*(\Delta y)X) & = & b - \mu\mathcal{A}(Z^{-1}), \\ \Delta Z & = & -\mathcal{A}^*(\Delta y), \\ \Delta X & = & \mu Z^{-1} - X - Z^{-1}\Delta Z X. \end{array}$$

The specific form of the first equation, often called the Schur-Complement equation, depends on how exactly $ZX - \mu I$ is symmetrized. We refer to [Tod01] for a thorough treatment of this issue. The main effort consists in setting up and solving this Schur equation. Several public domain packages based on this approach are available. The Mittelmann website (`http://plato.asu.edu/ftp/sdplib.html`) reports benchmark computations using several implementations of this idea. Looking at the results, it becomes clear that all of these methods have their limits once the matrix dimension n goes beyond 1000 or once the number m of constraints is larger than, say, 5000. These limitations are caused by the dense linear algebra operations with matrices of order n and by setting up and solving the Schur complement equation of order m. By using iterative

methods to solve this equation, [Toh04] manages to solve certain types of SDP problems with m up to 100,000. Similarly [KS07] combines an iterative solver with a modified barrier formulation of the dual SDP and also report computational results with the code PENNON with m up to 100,000. Another approach to solve the pair of dual SDP problems (1),(2) is to use nonlinear optimization techniques. For instance, [HR00], [BM03], [Mon03] or [BV06] solve (1),(2) after rewriting them as nonlinear programs. Strong computational results on large problems with medium accuracy have been reported for these algorithms. We also mention the mirror-prox method recently proposed in [LNM07] as a (first order) method tailored for large scale structured SDP.

In this paper, we study alternatives to all these methods, using quadratic regularization of SDP problems. As linear problems, the primal problem (1) and the dual problem (2) can indeed admit several solutions which can be moreover very sensitive to small perturbations of the data C, $\mathcal{A}$ and b. A classical and fruitful idea in non-smooth or constrained optimization (which can be traced back to [BKL66], [Mar70]) is to stabilize the problems by adding quadratic terms: this will ensure existence, uniqueness and stability of solutions. Augmented Lagrangian methods [PT72], [Roc76a] are well-known important regularization techniques. Regarding SDP, an augmented Lagrangian on the primal problem (1) was considered in [BV06] and an augmented Lagrangian on the dual (2) in [Ren05] and [PRW06], introducing the so-called "boundary point method". More recently, [JR08] propose an augmented primal-dual method which is similar to the present approach, and report computational results for large-scale random SDP. The idea to apply a proximal regularization to SDP is mentioned in [Mal05].

The main contributions of this paper are the following. We introduce and study a new class of regularization methods for SDP and we show that our new methods are efficient on several classes of large-scale SDP problems (n not too large, say $n \leq 1000$, but with a large number of constraints). We also show that the boundary point method [PRW06] has a natural interpretation as one particular instantiation of our general class of methods, thus we put this method into perspective.

Here is a brief outline of the paper. In the following subsection, we finish the introduction by presenting two quadratic regularizations for SDP problems. They will eventually result in a general regularization algorithm for SDP (Algorithm 4.3, studied in Section 4). The connection of this algorithm with the boundary point method will be made explicit by Proposition 4.4 (and Proposition 3.4). Before this, Sections 2 and 3 are devoted to the study of a particular type of a nonlinear SDP problem - the "inner problem" - appearing in the two regularizations. Section 2 studies optimality condition for this type of nonlinear SDP problems, while Section 3 presents a general approach to solve them. Section 4 then uses these elements to set up the regularization algorithm. Finally in Section 5 we report computational experiments on randomly generated instances as well as instances from publicly available libraries. A simple instance of the regularization algorithm compares favorably on these instances with the best SDP solvers.

1.2 Quadratic regularizations

We focus on two quadratic regularizations: Moreau-Yosida regularization for the primal problem (1) and augmented Lagrangian method for the dual problem (2). It is known that these two regularizations are actually equivalent, as primal and dual views of the same process. We recall them briefly as we will constantly draw connections between the primal and the dual point of view.

Primal Moreau-Yosida regularization. We denote by $\|\cdot\|$ the norms associated to standard inner products for both spaces $\mathcal{S}_n$ and $\mathbb{R}^n$. We begin with considering, for given $t > 0$, the following problem

$$\left\{ \begin{array}{l} \min \quad \langle C, X\rangle + \frac{1}{2t}\|X - Y\|^2 \\ \quad \mathcal{A}X = b,\ X \succeq 0,\ Y \in \mathcal{S}_n, \end{array} \right. \tag{5}$$

which is clearly equivalent to (1) (by minimizing first with respect to Y and then with respect to X). The idea is then to solve (5) in two steps: For Y fixed, we minimize first with respect to X and the result is then minimized with respect to Y. Thus we consider the so-called Moreau-Yosida regularization $F_t \colon \mathcal{S}_n \to \mathbb{R}$ defined as the optimal value

$$F_t(Y) = \min_{X \succeq 0, \mathcal{A}X = b} \langle C, X\rangle + \frac{1}{2t}\|X - Y\|^2; \tag{6}$$

and therefore we have

$$\min_{Y \in \mathcal{S}_n} F_t(Y) = \min_{X \succeq 0, \mathcal{A}X = b} \langle C, X\rangle. \tag{7}$$

By the strong convexity of the objective function of (6), the point that achieves the minimum is unique; it is called the proximal point of Y (with parameter t) and it is denoted by $\mathrm{P}_t(Y)$. The next proposition recalls the well-known properties of the Moreau-Yosida regularization F_t (see the section XV.4 of [HUL93] for instance).

Proposition 1.1 (Properties of the regularization)**.** *The function F_t is finite everywhere, convex and differentiable. Its gradient at $Y \in \mathcal{S}_n$ is*

$$\nabla F_t(Y) = \frac{1}{t}(Y - \mathrm{P}_t(Y)). \tag{8}$$

The functions $\nabla F_t(\cdot)$ and $\mathrm{P}_t(\cdot)$ are Lipschitz continuous.

Dual regularization by augmented Lagrangian. The augmented Lagrangian technique to solve (2) (going back to [Hes69], [Pow69] and [Roc76a]) introduces the augmented Lagrangian function $\mathscr{L}_\sigma$ with parameter $\sigma > 0$:

$$\mathscr{L}_\sigma(y, Z; Y) = b^\top y - \langle Y, \mathcal{A}^* y + Z - C\rangle - \frac{\sigma}{2}\|\mathcal{A}^* y + Z - C\|^2.$$

This is just the usual Lagrangian for the problem

$$\left\{ \begin{array}{l} \max \quad b^\top y - \frac{\sigma}{2}\|\mathcal{A}^* y + Z - C\|^2 \\ \quad C - \mathcal{A}^* y = Z,\ Z \succeq 0, \end{array} \right. \tag{9}$$

that is (2) with an additional redundant quadratic term. The (bi)dual function is in this case

$$\Theta_\sigma(Y) = \max_{y,Z\succeq 0} \mathscr{L}_\sigma(y, Z; Y). \tag{10}$$

A motivation for this approach is that Θ_σ is differentiable everywhere, in contrast to the dual function associated with (2). Solving this latter problem by the augmented Lagrangian method then consists in minimizing Θ_σ.

Connections. The bridge between the primal and the dual regularizations is formalized by the following proposition. It is a known result (see [HUL93, XII.5.1.1] for the general case), and it will be a straightforward corollary of the forthcoming Proposition 2.1.

Proposition 1.2 (Outer connection)**.** *If $t = \sigma$, then $\Theta_\sigma(Y) = F_t(Y)$ for all $Y \in \mathcal{S}_n$.*

The above two approaches thus correspond to the same quadratic regularization process viewed either on the primal problem (1) or on the dual (2). The idea to solve the pair of SDP problems is to use the differentiability of F_t (or Θ_σ). This can be seen from the primal point of view: the constrained program (1) is replaced by (7), leading to the unconstrained minimization of the convex differentiable function F_t. The proximal algorithm [Roc76b] consists in applying a descent method to minimize F_t, for instance the gradient method with fixed step size t. In view of (8), this gives the simple iteration $Y_{\text{new}} = \mathrm{P}_t(Y)$.

In summary, the solution of the semidefinite program (1) by quadratic regularization requires an iterative scheme (outer algorithm) to minimize F_t or Θ_σ. Evaluating $F_t(Y)$ or $\Theta_\sigma(Y)$ is itself an optimization problem, which we call the "inner problem". From a practical point of view, we are interested in efficient methods that yield approximate solutions of the inner problem. In the following section we therefore investigate the optimality conditions of (6) and (10). These are then used to formulate algorithms for the inner problem. We will then be in position to describe the overall algorithm.

2 Inner problem: optimality conditions

In this section and the next one, we look at the problem of evaluating $F_t(Y)$ for some given Y. Since $F_t(Y)$ is itself the result of the minimization problem,

$$\begin{cases} \min \quad \langle C, X\rangle + \frac{1}{2t}\|X - Y\|^2 \\ \quad \mathcal{A}X = b,\ X \succeq 0, \end{cases} \tag{11}$$

we consider various techniques to carry out this minimization at least approximately. In this section we study Lagrangian duality of (11) and we take a closer look at the optimality conditions. The next section will be devoted to algorithms based on these elements.

We start by introducing the following notation that will be used extensively in the sequel. The projection of a matrix $A \in \mathcal{S}_n$ onto the (closed convex) cone

$\mathcal{S}_n^+$ and its polar cone $\mathcal{S}_n^-$ are denoted respectively by

$$A_+ = \operatorname*{argmin}_{X \succeq 0} \|X - A\| \qquad \text{and} \qquad A_- = \operatorname*{argmin}_{X \preceq 0} \|X - A\|.$$

Theorem III.3.2.5 of [HUL93] for instance implies that

$$A = A_+ + A_- \ . \tag{12}$$

In fact we have the decomposition explicitly. Let $A = \sum_i \lambda_i p_i p_i^T$ be the spectral decomposition of A with eigenvalues λ_i and eigenvectors p_i, which may be assumed to be pairwise orthogonal and of length one. Then it is well known that

$$A_+ = \sum_{\lambda_i > 0} \lambda_i p_i {p_i}^\top \qquad \text{and} \qquad A_- = \sum_{\lambda_i < 0} \lambda_i p_i {p_i}^\top.$$

Observe also that we have for any $A \in \mathcal{S}_n$ and $t > 0$,

$$(tA)_+ = tA_+ \qquad \text{and} \qquad (-A)_+ = -(A_-) \ . \tag{13}$$

We dualize in (11) the two constraints by introducing the Lagrangian

$$L_t(X; y, Z) = \langle C, X \rangle + \frac{1}{2t}\|X - Y\|^2 - \langle y, \mathcal{A}X - b \rangle - \langle Z, X \rangle,$$

a function of the primal variable $X \in \mathcal{S}_n$ and the dual $(y, Z) \in \mathbb{R}^m \times \mathcal{S}_n^+$. The dual function defined by

$$\theta_t(y, Z) = \min_{X \in \mathcal{S}_n} L_t(X; y, Z) \tag{14}$$

is then described as follows.

Proposition 2.1 (Inner dual function)**.** *The minimum in* (14) *is attained at*

$$X(y, Z) = t(Z + \mathcal{A}^* y - C) + Y. \tag{15}$$

The dual function θ_t is equal to

$$\theta_t(y, Z) = b^\top y - \langle Y, Z + \mathcal{A}^* y - C \rangle - \frac{t}{2}\|Z + \mathcal{A}^* y - C\|^2. \tag{16}$$

Moreover θ_t is differentiable: its gradient with respect to y is

$$\nabla_y \, \theta_t(y, Z) = b - \mathcal{A}(t(Z + \mathcal{A}^* y - C) + Y), \tag{17}$$

and its gradient with respect to Z is

$$\nabla_Z \, \theta_t(y, Z) = -(t(Z + \mathcal{A}^* y - C) + Y).$$

Proof. Let $(y, Z) \in \mathbb{R}^m \times \mathcal{S}_n^+$ fixed. The function $X \mapsto L_t(X; y, Z)$ is strongly convex and differentiable. Then it admits a unique minimum point $X(y, Z)$ satisfying

$$0 = \nabla_X L_t\big(X(y,Z); y, Z\big) = C + \frac{1}{t}(X(y,Z) - Y) - \mathcal{A}^* y - Z$$

which gives (15). Thus the dual function can be rewritten as

$$\begin{aligned}
\theta_t(y,Z) &= L_t(X(y,Z); y, Z) \\
&= b^\top y + \langle C - \mathcal{A}^* y - Z, t(Z + \mathcal{A}^* y - C) + Y\rangle + \frac{t}{2}\|Z + \mathcal{A}^* y - C\|^2 \\
&= b^\top y - \langle Y, Z + \mathcal{A}^* y - C\rangle - \frac{t}{2}\|Z + \mathcal{A}^* y - C\|^2
\end{aligned}$$

This function is differentiable with respect to (y, Z). For fixed Z, the gradient of $y \mapsto \theta_t(y, Z)$ is

$$\nabla_y \theta_t(y,Z) = b - \mathcal{A}Y - t\mathcal{A}(Z + \mathcal{A}^* y - C) = b - \mathcal{A}(t(Z + \mathcal{A}^* y - C) + Y),$$

the one of $Z \mapsto \theta_t(y, Z)$ is

$$\nabla_Z \theta_t(y,Z) = -(t(Z + \mathcal{A}^* y - C) + Y),$$

This completes the proof. □

In view of (16), the dual problem of (11) can be formulated as

$$\max_{y \in \mathbb{R}^m, Z \succeq 0} \quad b^\top y - \langle Y, Z + \mathcal{A}^* y - C\rangle - \frac{t}{2}\|Z + \mathcal{A}^* y - C\|^2. \tag{18}$$

Observe that (18) is exactly (10). Proposition 1.2 now becomes obvious, and is formalized in the following remark.

Remark 2.2 (Proof of Proposition 1.2)**.** *Proposition 2.1 and [HUL93, XII.2.3.6] imply that there is no duality gap. Thus for $t = \sigma$ we have $F_t(Y) = \Theta_\sigma(Y)$ by equations* (6) *and* (10)*.* □

We also get the expression of the proximal point and of the gradient of the Moreau regularization F_t in terms of solutions of (18).

Corollary 2.3 (Gradient of F_t)**.** *Let $(\bar{y}, \bar{Z})$ be a solution of* (18)*. Then*

$$P_t(Y) = t(\bar{Z} + \mathcal{A}^* \bar{y} - C) + Y \qquad \textit{and} \qquad \nabla F_t(Y) = -(\bar{Z} + \mathcal{A}^* \bar{y} - C).$$

Proof. Given a solution $(\bar{y}, \bar{Z})$ of (18), the general duality theory (see for example [HUL93, XII.2.3.6]) yields that $X(\bar{y}, \bar{Z}) = t(\bar{Z} + \mathcal{A}^* \bar{y} - C) + Y$ is the unique solution of the primal inner problem (11), that is $P_t(Y)$ by definition. Moreover, (8) gives the desired expression for the gradient. □

The following technical lemma specifies the role of the primal Slater assumption.

Lemma 2.4 (Coercivity)**.** *Assume that there exist $\bar{X} \succ 0$ such that $\mathcal{A}\bar{X} = b$ and that $\mathcal{A}$ is surjective. Then θ_t is coercive, in other terms $\theta_t(y, Z) \to -\infty$ when $\|(y, Z)\| \to +\infty,\ Z \succeq 0$.*

Proof. By (14) defining θ_t, we have for all $(X, y, Z) \in \mathcal{S}_n^+ \times \mathbb{R}^m \times \mathcal{S}_n^+$

$$\theta_t(y, Z) \leq \langle C, X\rangle + \frac{1}{2t}\|X - Y\|^2 - \langle y, \mathcal{A}X - b\rangle - \langle Z, X\rangle. \tag{19}$$

By surjectivity of $\mathcal{A}$, there exist $r > 0$ and $R > 0$ such that for all $\gamma \in \mathbb{R}^m$ with $\|\gamma\| < 2r$, there exists $X_\gamma \succ 0$ with $\|X_\gamma - \bar{X}\| \leq R$ satisfying $\mathcal{A}X_\gamma - b = \gamma$. Then set, for $y \in \mathbb{R}^m$,

$$\bar{\gamma} = r\frac{y}{\|y\|},$$

and plug the associated $X_{\bar{\gamma}}$ into (19) to get

$$\begin{aligned}\theta_t(y, Z) &\leq \langle C, X_{\bar{\gamma}}\rangle + \frac{1}{2t}\|X_{\bar{\gamma}} - Y\|^2 - \langle y, \bar{\gamma}\rangle - \langle Z, X_\gamma\rangle \\ &= \langle C, X_{\bar{\gamma}}\rangle + \frac{1}{2t}\|X_{\bar{\gamma}} - Y\|^2 - r\|y\| - \langle Z, X_{\bar{\gamma}}\rangle.\end{aligned}$$

Observe that the minimum of $\langle Z, X_\gamma\rangle$ is attained over the compact set defined by $\|\gamma\| \leq r$, $\{Z \succeq 0$ and $\|Z\| = 1\}$. Call the minimum M and the points achieving the minimum $\tilde{Z} \succeq 0$ and $\tilde{X} \succ 0$, so that $M > 0$. Then we can derive the bounds, for all (y, Z),

$$\langle Z, X_{\bar{\gamma}}\rangle = \left\langle \frac{Z}{\|Z\|}, X_{\bar{\gamma}}\right\rangle \|Z\| \geq M\|Z\|$$

and

$$\theta_t(y, Z) \leq \langle C, X_{\bar{\gamma}}\rangle + \frac{1}{2t}\|X_{\bar{\gamma}} - Y\|^2 - r\|y\| - M\|Z\|$$

To conclude, note that the quantity $\langle C, X_{\bar{\gamma}}\rangle + \frac{1}{2t}\|X_{\bar{\gamma}} - Y\|^2$ is bounded, since $X_{\bar{\gamma}}$ stays on a ball centered in $\bar{X}$. So we see that $\theta_t(y, Z) \to -\infty$ when $\|y\| \to +\infty$ or $\|Z\| \to +\infty$. □

Remark 2.5 (A simple example showing that Lemma 2.4 is wrong without a Slater point)**.** *Let J be the matrix of all ones, and consider the problem*

$$\begin{cases} \min \quad \langle C, X\rangle \\ \quad \langle J, X\rangle = 0,\ \langle I, X\rangle = 1,\ X \succeq 0, \end{cases}$$

and its dual

$$\begin{cases} \max \quad y_2 \\ \quad C - y_1 J - y_2 I = Z,\ Z \succeq 0. \end{cases}$$

We first observe that the primal problem has no Slater point. To see this, take a feasible X, and write

$$\lambda_{\min}(X) = \min_{z \neq 0} \frac{z^T X z}{z^T z} \leq \frac{e^T X e}{e^T e} = \frac{\langle J, X\rangle}{n} = 0.$$

Together with $X \succeq 0$, it follows $\lambda_{\min}(X) = 0$, hence X is singular, and thus there is no Slater point.

Now we show that the dual function is not coercive. By (16)*, there holds*

$$\theta_t(y, Z) = y_2 - \langle Y, Z + y_1 J + y_2 I - C\rangle - \frac{t}{2}\|Z + y_1 J + y_2 I - C\|^2.$$

Choosing $Z = -y_1 J$ and $y_2 = 0$ with $y_1 < 0$ and $y_1 \to -\infty$, we have (y, Z) with $\|(y, Z)\| \to +\infty$. However, substituting in θ_t, we obtain

$$\theta_t(y, Z) = \langle Y, C\rangle - \frac{t}{2}\|C\|^2,$$

which is constant. □

We end this subsection summarizing the optimality conditions for (11) and (18) under the primal Slater assumption.

Proposition 2.6 (Optimality conditions)**.** *If there exists a positive definite matrix $\bar{X}$ satisfying $\mathcal{A}\bar{X} = b$, then for any $Y \in \mathcal{S}_n$ there exist primal and dual optimal solutions (X, y, Z) for* (11),(18)*, and there is no duality gap. In this case, the following statements are equivalent:*

(i) (X, y, Z) *is optimal for* (11),(18)*.*

(ii) (X, y, Z) *satisfies*

$$\begin{cases} \mathcal{A}X = b, \quad X = t(Z + \mathcal{A}^*y - C) + Y, \\ X \succeq 0, \quad Z \succeq 0, \quad \langle X, Z\rangle = 0. \end{cases} \tag{20}$$

(iii) (X, y, Z) *satisfies*

$$\begin{cases} X = t(Y/t + \mathcal{A}^*y - C)_+ \\ Z = -(Y/t + \mathcal{A}^*y - C)_- \\ \mathcal{A}\mathcal{A}^*y + \mathcal{A}(Z - C) = (b - \mathcal{A}Y)/t. \end{cases} \tag{21}$$

Proof. The Slater assumption implies, first, that the intersection of $\mathcal{S}_n^+$ and $\mathcal{A}X = b$ is non-empty, and therefore that there exists a solution to (11), and second, that there exist dual solutions (thanks to Lemma 2.4).

Observe now that *(ii)* are the KKT conditions of (11), which gives the equivalence between *(i)* and *(ii)* since the problems are convex. The equivalence between *(ii)* and *(iii)* comes essentially from (12) (precisely from [HUL93, Theorem III.3.2.5]), which ensures that

$$X/t - Z = Y/t + \mathcal{A}^*y - C, \quad X \succeq 0, \quad Z \succeq 0, \quad \langle X, Z\rangle = 0$$

is equivalent to

$$X/t = (Y/t + \mathcal{A}^*y - C)_+, \quad -Z = (Y/t + \mathcal{A}^*y - C)_-.$$

Using the equality $X/t = Y/t + \mathcal{A}^*y + Z - C$, we can also replace the variable X in $\mathcal{A}X = b$ to obtain exactly (21). □

3 Inner problem: algorithms

We have just seen that the optimality conditions for the inner problem have a rich structure. We are going to exploit it and consider several approaches to get approximate solutions of the inner problem.

3.1 Using the optimality conditions

A simple method to solve the inner problem (18) exploits the fact (look at the optimality conditions of (21)) that for fixed Z, the vector y can be determined from a linear system of equations, and for fixed y, the matrix Z is obtained by projection. This is the idea used in the boundary point method [Ren05], [PRW06] whose corresponding inner problem is solved by the following two-step process.

Algorithm 3.1 (Two-step iterative method for inner problem)**.**
Given $t > 0$ and $Y \in \mathcal{S}_n$.
Choose $y \in \mathbb{R}^m$ and set $Z = -(Y/t + \mathcal{A}^* y - C)_-$.
Repeat until $\|\mathcal{A}X - b\|$ is small:
Step 1: Compute the solution y of $\mathcal{A}\mathcal{A}^* y = \mathcal{A}(C - Z) + (b - \mathcal{A}Y)/t$.
Step 2: Update $Z = -(Y/t + \mathcal{A}^* y - C)_-$ and $X = t(Y/t + \mathcal{A}^* y - C)_+$.

By construction, each iteration of this two-step process guarantees that

$$X \succeq 0, \quad Z \succeq 0, \quad \langle X, Z \rangle = 0.$$

This explains the name "boundary point method". Observe also that Step 1 of Algorithm 3.1 amounts to solving an $m \times m$ linear system of the form $\mathcal{A}\mathcal{A}^* y =$ rhs, which can be expensive if m is large. However, in contrast to interior point methods, the matrix $\mathcal{A}\mathcal{A}^*$ is constant throughout the algorithm. So it can be decomposed at the beginning of the process once and for all. Moreover, the matrix structure can be exploited to speed up calculation.

To see that the stopping condition makes sense, we observe that after Step 2 of the algorithm is executed, the only possibly violated optimality condition is $\mathcal{A}\mathcal{A}^* y + \mathcal{A}(Z - C) = (b - \mathcal{A}Y)/t$, using (21). After Step 2, X and Z satisfy $X = t(Z + \mathcal{A}^* y - C) + Y$. This condition holding, it is clear that $\mathcal{A}X = b$ if and only if $\mathcal{A}\mathcal{A}^* y + \mathcal{A}(Z - C) = (b - \mathcal{A}Y)/t$. We will come back to convergence issues in more detail in Section 4.2.

3.2 Using the dual function

We consider again the dual inner problem (18). We observe that the minimization with respect to Z, with y held constant, leads to a projection onto $\mathcal{S}_n^+$ and results in the dual function

$$\widetilde{\theta}_t(y) = \max_{Z \succeq 0} \theta_t(y, Z)$$

depending on y only. The differentiability properties of this function are now summarized.

Proposition 3.2 (Dual function)**.** *The function $\widetilde{\theta}_t$ is a differentiable function that can be expressed (up to a constant) as*

$$\widetilde{\theta}_t(y) = b^\top y - \frac{t}{2} \left\| (\mathcal{A}^* y - C + Y/t)_+ \right\|^2 \tag{22}$$

and whose gradient is $\nabla\widetilde{\theta}_t(y) = b - t\mathcal{A}(\mathcal{A}^* y - C + Y/t)_+$.

Proof. With the help of (16), express $\widetilde{\theta}_t(y)$ as the minimum

$$\widetilde{\theta}_t(y) = -\min_{Z \succeq 0} -\theta_t(y, Z) = b^\top y - \min_{Z \succeq 0} \langle Y, Z + \mathcal{A}^* y - C\rangle + \frac{t}{2}\|Z + \mathcal{A}^* y - C\|^2.$$

Rewrite this objective function as

$$\langle Y, Z + \mathcal{A}^* y - C\rangle + \frac{t}{2}\|Z + \mathcal{A}^* y - C\|^2 = \frac{t}{2} \left\| Z - (C - Y/t - \mathcal{A}^* y) \right\|^2 - \frac{1}{2t} \left\| Y \right\|^2 .$$

Observe that the minimum is attained by the projection of $C - Y/t - \mathcal{A}^* y$ onto $\mathcal{S}_n^+$ that is by the matrix

$$Z(y) = (C - Y/t - \mathcal{A}^* y)_+ = -(\mathcal{A}^* y - C + Y/t)_- . \tag{23}$$

Observe also that the function $Z\colon y \mapsto (C - Y/t - \mathcal{A}^* y)_+$ is continuous (since the projection $X \mapsto X_+$ is Lipschitz continuous). Equation (12) enables to write

$$\widetilde{\theta}_t(y) = b^\top y - \frac{t}{2} \left\| (Y/t - C + \mathcal{A}^* y)_+ \right\|^2 + \frac{1}{2t} \left\| Y \right\|^2 . \tag{24}$$

We now want to use a theorem of differentiability of a min function (as [HUL93, VI4.4.5] for example) to compute the derivative of $\widetilde{\theta}_t$. So we need to ensure the compactness of the index on which the maximum is taken. Let $\bar{y} \in \mathbb{R}^m$ and V a compact neighborhood of $\bar{y}$ in $\mathbb{R}^m$. By continuity of $Z(\cdot)$ defined by (23), the set $U = Z(V)$ is compact and we can write

$$\widetilde{\theta}_t(y) = \max_{Z \in U \cap \mathcal{S}_n^+} \theta_t(y, Z)$$

for all $y \in V$. Since the maximum is taken on a compact set, [HUL93, VI.4.4.5] gives that $\widetilde{\theta}_t$ is differentiable at $\bar{y}$ with

$$\nabla\widetilde{\theta}_t(\bar{y}) = \nabla_y \theta_t(\bar{y}, Z(\bar{y}))$$

for which we have an expression (recall (17)). Since this can be done around any $\bar{y} \in \mathbb{R}^m$, we conclude that $\widetilde{\theta}$ is differentiable on $\mathbb{R}^m$. Using (23), we compute, for $y \in \mathbb{R}^m$

$$\begin{aligned} \nabla\widetilde{\theta}_t(y) &= b - t\mathcal{A}(Z(y) + \mathcal{A}^* y - C + Y/t) \\ &= b - t\mathcal{A}\big((\mathcal{A}^* y - C + Y/t)_+\big) \end{aligned}$$

and the proof is complete. □

The dual problem (18) can thus be cast as the following concave differentiable problem

$$\max_{y\in\mathbb{R}^m} \quad b^\top y - \frac{t}{2}\left\|(\mathcal{A}^* y - C + Y/t)_+\right\|^2, \tag{25}$$

up to a constant, which is explicitly $\|Y\|^2/2t$ (see (24)). So we can use this formulation to solve the inner problem (11) through its dual (25) (when there is no duality gap, see Proposition 2.6). The key is that the objective function $\widetilde{\theta}_t$ is concave, differentiable with an explicit expression of its gradient (Proposition 3.2). Thus we can use any classical algorithm of unconstrained convex programming to solve (25). In particular we can use

- gradient-like strategies (e.g. gradient, conjugate gradient or Nesterov method)
- Newton-like strategies (e.g. quasi-Newton or inexact generalized Newton).

For generality and simplicity, we consider the following variable metric method to solve (25), which generalizes many of the above classical strategies.

Algorithm 3.3 (Dual variable metric method for the inner problem)**.**
Given $t > 0$ and $Y \in \mathcal{S}_n$. Choose $y \in \mathbb{R}^m$.
Repeat until $\|b - \mathcal{A}X\|$ is small:
Compute $X = t(\mathcal{A}^* y - C + Y/t)_+$.
Set $g = b - \mathcal{A}X$.
Update $y \leftarrow y + \tau W g$ with appropriate W and τ.

The stopping condition is as before, but now it is motivated by the fact that $\nabla\widetilde{\theta}_t(y) = b - t\mathcal{A}(\mathcal{A}^* y - C + Y/t)_+ = b - \mathcal{A}X$. In fact the connections between Algorithm 3.3 and the previous inner method (Algorithm 3.1) are strong; they are precisely stated by the following proposition.

Proposition 3.4 (Inner connection)**.** *If $\mathcal{A}$ is surjective, Algorithm 3.1 generates the same sequence as Algorithm 3.3 when both algorithms start from the same dual iterate y, and when W and τ are kept constant for all iterations, equal to*

$$W = [\mathcal{A}\mathcal{A}^*]^{-1} \qquad \text{and} \qquad \tau = 1/t.$$

Proof. The surjectivity of $\mathcal{A}$ implies that $\mathcal{A}\mathcal{A}^*$ is invertible and then that the sequence $(X_k, y_k, Z_k)_k$ generated by Algorithm 3.1 is well defined as well as the sequence $(\widetilde{X}_k, \widetilde{y}_k)_k$ generated by Algorithm 3.3 when $W = [\mathcal{A}\mathcal{A}^*]^{-1}$ and $\tau = 1/t$. Let us prove by induction that $y_k = \widetilde{y}_k$ and $X_k = \widetilde{X}_k$ for all $k \geq 0$.

We assume that the two algorithms start by the same dual iterate $y_0 = \widetilde{y}_0$. It holds that $X_0 = \widetilde{X}_0 = t(Y/t + \mathcal{A}^* y_0 - C)_+$ by construction. Assume now that we have $y_k = \widetilde{y}_k$ and $X_k = \widetilde{X}_k$. To prove that $\widetilde{y}_{k+1} = y_{k+1}$, we check if $\widetilde{y}_{k+1}$ defined by

$$\widetilde{y}_{k+1} = \widetilde{y}_k + \frac{1}{t}[\mathcal{A}\mathcal{A}^*]^{-1}(b - \mathcal{A}\widetilde{X}_k)$$

satisfies the equation defining y_{k+1}, that is, if

$$\Delta_k = \mathcal{A}\mathcal{A}^* \widetilde{y}_{k+1} - \mathcal{A}(C - Z_k) - (b - \mathcal{A}Y)/t$$

is null. By construction of $\widetilde{y}_{k+1}$, we have

$$\begin{aligned} \Delta_k &= \mathcal{A}\mathcal{A}^* \Big(\widetilde{y}_k + [\mathcal{A}\mathcal{A}^*]^{-1}(b - \mathcal{A}\widetilde{X}_k)/t \Big) - \mathcal{A}(C - Z_k) - (b - \mathcal{A}Y)/t \\ &= \mathcal{A}\mathcal{A}^* \widetilde{y}_k + (b - \mathcal{A}\widetilde{X}_k)/t - (b - \mathcal{A}Y)/t - \mathcal{A}(C - Z_k) \\ &= -\mathcal{A}\big(\widetilde{X}_k/t - Z_k - (Y/t + \mathcal{A}^* \widetilde{y}_k - C) \big). \end{aligned}$$

Since $\widetilde{X}_k = X_k$ and $\widetilde{y}_k = y_k$, we get $\Delta_k = 0$ (by construction of X_k and Z_k in Step 2 of Algorithm 3.1). Hence, $y_{k+1} = \widetilde{y}_{k+1}$, and it yields

$$\widetilde{X}_{k+1} = t(\mathcal{A}^* y_{k+1} - C + Y/t)_+ = X_{k+1},$$

and the proof is completed by induction. Note also that the two algorithms have the same stopping rule. □

A direct calculation shows that the primal inner problem (11) can be cast as

$$\left\{ \begin{array}{ll} \min & \frac{1}{2t} \|X - (Y - tC)\|^2 \\ & \mathcal{A}X = b, \\ & X \succeq 0. \end{array} \right. \tag{26}$$

This problem is a so-called semidefinite least-squares problem: we want to compute the nearest matrix to $(Y - tC)$ belonging to $\mathcal{C}$, the intersection of $\mathcal{S}_n^+$ with the affine subspace $\mathcal{A}X = b$. In others words, we want to project the matrix $Y - tC$ onto the intersection. The problem received recently a great interest (see [Mal04] for the general case, and [Hig02], [QS06] for the important special case of correlation matrices).

4 Outer algorithm

In the previous section we have investigated how the inner problem, that is evaluating $F_t(Y)$ or $\Theta_\sigma(Y)$ for some given Y, can be done approximately. These methods are the backbone of the overall algorithm, which we are going to describe in this section.

4.1 Conceptual outer algorithm

We start with the primal point of view. The Moreau-Yosida quadratic regularization of (1) leads us to a conceptual proximal algorithm, which can be written as follows.

Algorithm 4.1 (Conceptual proximal algorithm)**.**
Initialization: Choose $t > 0$ *and* $Y \in \mathcal{S}_n$.
Repeat until $\frac{1}{t}\|Y - \mathrm{P}_t(Y)\|$ *is small:*
Step 1: Solve the inner problem (11) *to get* $X = \mathrm{P}_t(Y)$.
Step 2: Set $Y = X$ *and update* t.

The stopping condition is based on the gradient (8); but obviously, this "algorithm" is only conceptual since it requires $\mathrm{P}_t(Y)$. Moving now to the dual point of view, we propose to apply the augmented Lagrangian method to solve (2) leading to the following algorithm.

Algorithm 4.2 (Conceptual "boundary point")**.**
Initialization: Choose $\sigma > 0$ *and* $Y \in \mathcal{S}_n$.
Repeat until $\|Z + \mathcal{A}^* y - C\|$ *is small:*
Step 1: Solve the inner problem (18) *to get* (y, Z).
Step 2: Compute $X = Y + \sigma(Z + \mathcal{A}^* y - C)$*, set* $Y = X$ *and update* σ.

If the two inner steps (Step 1 just above and Step 1 of Algorithm 4.1) are solved exactly, then the expression of the gradient of the regularization F_t (recall Corollary 2.3) would show that the previous algorithm produces the same iterates as Algorithm 4.1. In other words, the conceptual boundary point method is equivalent to the conceptual proximal algorithm. Proposition 4.4 below shows that this correspondence property also holds when the inner problems are solved approximately. Note that we implicitly assume that the two regularization parameters are equal ($t = \sigma$); for clarity, we use only t for the rest of the paper.

Implementable versions of the above algorithms require the computation of Step 1. In view of the previous sections, we use the general Algorithm 3.3 inside of Algorithms 4.1 and 4.2 to solve Step 1 (inner problem), and we introduce a tolerance $\varepsilon^{\text{inner}}$ for the inner error and another tolerance $\varepsilon^{\text{outer}}$ for the outer stopping condition. We thus obtain the following regularization algorithm for SDP.

Algorithm 4.3 (Regularization algorithm for SDP)**.**
Initialization: Choose initial iterates Y, y, *and* $\varepsilon^{\text{inner}}, \varepsilon^{\text{outer}}$.
Repeat until $\|Z + \mathcal{A}^* y - C\| \leq \varepsilon^{\text{outer}}$*:*
Repeat until $\|b - \mathcal{A}X\| \leq \varepsilon^{\text{inner}}$*:*
Compute $X = t(\mathcal{A}^* y - C + Y/t)_+$ *and* $Z = -(\mathcal{A}^* y - C + Y/t)_-$.
Update $y \leftarrow y + \tau W g$ *with appropriate* τ *and* W.
end (inner repeat)
$Y \leftarrow X$.
end (outer repeat)

We note that Algorithm 4.3 has the following particular feature. It is "orthogonal" to interior point methods in the sense that it works to enforce the primal and dual feasibilities while the complementarity and semidefiniteness conditions are guaranteed throughout. In contrast, interior-point methods maintain primal and dual feasibility and semidefiniteness and try to reach complementarity. We now establish the connection to the boundary point method from [PRW06].

Proposition 4.4 (Outer connections)**.** *If $\mathcal{A}$ is surjective, W and τ are fixed at*

$$W = [\mathcal{A}\mathcal{A}^*]^{-1} \qquad \text{and} \qquad \tau = 1/t,$$

then Algorithm 4.3 generates the same sequence as the boundary point method (see Table 2 of [PRW06]) when starting from the same initial iterates $Y_0 = 0$ and y_0 such that $\mathcal{A}^* y_0 - C \succeq 0$ and with the same stopping threshold $\varepsilon^{\text{inner}}$ and $\varepsilon^{\text{outer}}$.

Proof. The result follows easily from Proposition 3.4 by induction. Note first that Y_0 and y_0 give $Z_0 = 0$, and therefore that the initializations coincide. (For the boundary point method we refer to Table 2 of [PRW06].) Proposition 3.4 then shows that the two inner algorithms generate the same iterates, and they have the same stopping condition if $\varepsilon^{\text{inner}}$ and $\varepsilon^{\text{outer}}$ correspond to the thresholds of Table 2 of [PRW06]. Observe finally that the two outer iterations are also identical: the expressions of X and Z (Step 1 in Algorithm 4.3) give the update of the boundary point (Step 2 in Algorithm 4.2). □

4.2 Elements of convergence

The convergence behaviour of Algorithm 4.3 is for the moment much less understood than the convergence properties of interior point methods for example. This lack of theory also opens the way for many degrees of freedom in tuning parameters for the algorithm. Section 5 discusses briefly this question and presents the practical algorithm, used for experimentation.

For the sake of completeness we include here a first convergence theorem with an elementary proof. The theorem's assumptions and proof are inspired from classical references on the proximal and augmented Lagrangian methods ([PT72], [Roc76a], [Ber82], you may see also [BV06]).

Theorem 4.5 (Convergence)**.** *Denote by $(Y_p, y_p, Z_p)_p$ the sequence of (outer) iterates generated by Algorithm 4.3 and by $\varepsilon_p = \varepsilon_p^{\text{inner}}$ the associated inner error. Make three assumptions in Algorithm 4.3:*

- *t is constant,*
- *$\sum_p \varepsilon_p < \infty$,*
- *$(y_p)_p$ is bounded.*

If primal and dual problems satisfy the Slater assumption, the sequence $(Y_p,\, y_p,\, Z_p)_p$ *is asymptotically feasible:*

$$Z_p + \mathcal{A}^* y_p - C \to 0 \qquad \textit{and} \qquad \mathcal{A}Y_p - b \to 0.$$

Thus any accumulation point of the sequence gives a primal-dual solution.

Proof. Assume that the primal and dual problems satisfy the Slater assumption, and consider $(\bar{Y}, \bar{y}, \bar{Z})$ be an optimal solution, that is satisfying (4). The assumption that the series of ε_p converges yields that $\mathcal{A}Y_p - b \to 0$ is satisfied. We want to prove that the dual residue $G_p = -(Z_p + \mathcal{A}^* y_p - C)$ vanishes as well. Recall we have $Y_p = Y_{p-1} - tG_p$. So we develop

$$\begin{aligned}
\|Y_p - \bar{Y}\|^2 &= \|Y_{p-1} - \bar{Y}\|^2 - \|Y_{p-1} - \bar{Y}\|^2 + \|Y_p - \bar{Y}\|^2 \\
&= \|Y_{p-1} - \bar{Y}\|^2 - \|Y_p + tG_p - \bar{Y}\|^2 + \|Y_p - \bar{Y}\|^2 \\
&= \|Y_{p-1} - \bar{Y}\|^2 - 2\langle Y_p - \bar{Y}, tG_p\rangle - \|tG_p\|^2 \\
&= \|Y_{p-1} - \bar{Y}\|^2 - t^2\|G_p\|^2 - 2t\langle Y_p - \bar{Y}, G_p\rangle
\end{aligned}$$

Let us focus on the last term $\langle Y_p - \bar{Y}, G_p\rangle$. Using the optimality conditions (4), we write

$$G_p = -(Z_p + \mathcal{A}^* y_p - C) = -(Z_p - \bar{Z}) + \mathcal{A}^*(y_p - \bar{y}).$$

and then

$$\begin{aligned}
\langle Y_p - \bar{Y}, G_p\rangle &= -\langle Y_p - \bar{Y}, Z_p - \bar{Z} + \mathcal{A}^*(y_p - \bar{y})\rangle \\
&= -\langle Y_p - \bar{Y}, \mathcal{A}^*(y_p - \bar{y})\rangle - \langle Y_p, Z_p\rangle - \langle \bar{Y}, \bar{Z}\rangle + \langle Y_p, \bar{Z}\rangle + \langle \bar{Y}, Z_p\rangle \\
&\geq -\langle Y_p - \bar{Y}, \mathcal{A}^*(y_p - \bar{y})\rangle \\
&= -\langle \mathcal{A}(Y_p - \bar{Y}), y_p - \bar{y}\rangle \\
&\geq -\varepsilon_p\|y_p - \bar{y}\|,
\end{aligned}$$

the first inequality coming from $\langle \bar{Y}, \bar{Z}\rangle = \langle Y_p, Z_p\rangle = 0$ and $\bar{Y},\ \bar{Z},\ Y_p,\ Z_p \succeq 0$, and the second from the Cauchy-Schwarz Inequality. So we get

$$t^2\|G_p\|^2 \leq -\|Y_p - \bar{Y}\|^2 + \|Y_{p-1} - \bar{Y}\|^2 + 2t\varepsilon_p\|y_p - \bar{y}\|.$$

Summing these inequalities for the first N outer iterations, we finally get

$$t^2 \sum_{p=1}^{N} \|G_p\|^2 \leq -\|Y_N - \bar{Y}\|^2 + \|Y_0 - \bar{Y}\|^2 + 2t \sum_{p=1}^{N} \varepsilon_p\|y_p - \bar{y}\|.$$

Call M a bound of the sequence $\|y_p - \bar{y}\|$ and write

$$\sum_{p=1}^{N} \|G_p\|^2 \leq \frac{1}{t^2}\|Y_0 - \bar{Y}\|^2 + \frac{2M}{t} \sum_{p=1}^{N} \varepsilon_p.$$

By assumption, $\sum_p \varepsilon_p$ is finite, then so is $\sum_p \|G_p\|^2$, hence $G_p \to 0$ and the proof is complete. □

4.3 Min-Max Interpretation and stopping rules

We end the presentation of the overall algorithm by drawing connections with a method to solve min-max problems. Since we have an interpretation of the inner algorithm via duality, it makes sense indeed to cast the primal SDP problem as a min-max problem. With the help of (7) and (25) we can write (1) as

$$\min_{Y \in \mathcal{S}_n} \max_{y \in \mathbb{R}^m} \ \varphi(Y, y)$$

with

$$\varphi(Y, y) = b^\top y - \frac{t}{2} \left\| (\mathcal{A}^* y - C + Y/t)_+ \right\|^2 + \frac{1}{2t} \|Y\|^2. \tag{27}$$

Our approach can thus be interpreted as solving the primal and dual SDP problems by computing a saddle-point of φ.

With this point of view, the choice of stopping conditions appears to be even more crucial, because the inner and outer loops are antagonistic, as the first minimizes and the second maximizes. In this context, there are two opposite strategies. First, solving the inner maximization with respect to y with high accuracy (using a differentiable optimization algorithm) and then updating Y would amount to follow the conceptual proximal algorithm as closely as possible. Alternatively, we can do one gradient-like iteration with respect to each variable successively: this is essentially the idea of the "Arrow-Hurwicz" approach (see for instance [AHU59]), an algorithm that goes back to the 1950's, and that has been applied for example to the saddle-point problems that appear in finite element discretization of Stokes-type and elasticity equations and in mixed finite element discretization.

We tried both strategies and have observed that in practice the second option gives much better numerical results: for the numerical experiments in the next section, we thus always fix the inner iteration to one. Besides, an adequate stopping rule in such a situation still has to be studied theoretically; this is a general question, beyond the scope of this paper. Note also that since there is no distinction between the inner and the outer iteration in this approach, we use a pragmatic stopping condition on both the relative primal and dual error, see the next section.

5 Numerical illustrations

5.1 Simple regularization algorithm

Up to now we have seen that a practical implementation of a regularization approach for SDP can be derived from both the primal and the dual view points, resulting in the same algorithmic framework. The choice of the regularization parameter t (or σ), the tolerance levels for the inner problem, and the strategy for W have a strong impact on the actual performance. We tested various prototype algorithms, and decide to emphasize the following simple instance of the general algorithm which often yields to the overall best practical performance on our test problems.

- **Errors.** We use a relative error measure following [Mit03]. Thus the relative primal and dual infeasibilities are:

$$\delta_p := \frac{\|\mathcal{A}(X) - b\|}{1 + \|b\|}, \quad \delta_d := \frac{\|C - Z - \mathcal{A}^* y\|}{1 + \|C\|}, \quad \delta := \max\{\delta_p, \delta_d\}.$$

  Recall that all other optimality conditions hold by construction of the algorithm.

- **Inner iterations.** In our experiments we noticed that the overall performance is best if we execute the inner loop only once. As explained in Section 4.3, this is a plausible alternative in our context. With one inner iteration, the choice of W (and τ) is natural: we take $W = (\mathcal{A}\mathcal{A}^*)^{-1}$ and $\tau = 1/t$.

- **Normalization.** In order to simplify the choice for the internal parameters, we assume without loss of generality that the data are scaled so that both b and C have norm one.

- **Initializing t.** We select t large enough, so that the relative dual infeasibility is smaller than the relative primal infeasibility. If b and C have norm one, then a value of t in the range $0.1 \leq t \leq 10$ turned out to be a robust choice.

- **Updating t.** We use the following simple update strategy for t. As a general rule we try to change t as little as possible. Therefore we leave t untouched for a fixed number it of iterations. (Our choice is $it = 10$.) Every it-th iteration we check whether $\delta_p \leq \delta_d$. If this is not the case, we reduce t by a constant multiplicative factor, which we have set to 0.9. Otherwise we leave t unchanged.

- **Stopping condition.** We stop the algorithm once the overall error δ is below a desired tolerance ε, which we set to

$$\varepsilon = 10^{-7}.$$

  We also stop after a maximum of 300 iterations in case we do not reach the required level of accuracy.

These specifications lead to the following algorithm we use for numerical testing. Its MATLAB source code is available online at [Web].

Algorithm 5.1.
Initialization: Choose $t > 0$, $Y \in \mathcal{S}_n$ and ε. Set $Z = 0$.
Repeat until $\delta < \varepsilon$:
Compute the solution y of $\mathcal{A}\mathcal{A}^* y = \mathcal{A}(C - Z) + (b - \mathcal{A}Y)/t$.
Compute $Z = -(Y/t + \mathcal{A}^* y - C)_-$ and $X = t(Y/t + \mathcal{A}^* y - C)_+$.
Update $Y \leftarrow X$.
Compute δ.
Update t.

Note that the computational effort in each iteration is essentially determined by the eigenvalue decomposition of the matrix $Y/t + \mathcal{A}^* y - C$ of order n, needed to compute the projections X and Z, and by solving the linear system of order m with matrix $\mathcal{A}\mathcal{A}^*$. However, computing $\mathcal{A}\mathcal{A}^*$ and its Cholesky factorization is done only once at the beginning of the algorithm. Hence when n is not too big, the cost of each iteration is moderate, so we can target problems with m very large.

The following section provides some computational results with this algorithm. All computations are done under Linux on a Pentium 4 running with 3.2 GHz and 2.5 GB of RAM.

5.2 Pseudo-random SDP

Our algorithm should be considered as an alternative to interior-point methods in situations where the number m of equations prohibits direct methods to solve the Schur complement equations. In order to test our method systematically, we first consider 'unstructured' SDPs. The currently available libraries of SDP instances do not contain enough instances of this type with sizes suitable for our purposes. Therefore we consider randomly generated problems. Given the parameters n, m, p and *seed*, we generate pseudo-random instances as follows.

First generator. The main focus lies on $\mathcal{A}$ which should be such that the Cholesky factor of $\mathcal{A}\mathcal{A}^*$ can be computed with reasonable effort; this means we must control the sparsity of the matrix. The linear operator $\mathcal{A}$ is defined through m symmetric matrices A_i with $(\mathcal{A}X)_i = \langle A_i, X\rangle$. The matrix $\mathcal{A}\mathcal{A}^*$ therefore has entries $(\mathcal{A}\mathcal{A}^*)_{ij} = \langle A_i, A_j\rangle$. This means that the m matrices A_i forming $\mathcal{A}$ should be sparse, and the rows of $\mathcal{A}$ should be reordered to reduce the fill-in in the Cholesky factor. In order to control the sparsity of $\mathcal{A}$, we generate the matrices A_i defining $\mathcal{A}$ to be nonzero only on a submatrix of order p, whose support is selected randomly. Then we reorder the rows of A using the MATLAB command `symamd` to reduce the fill-in. So when n and p are fixed, it increases with m, and when m and p are fixed, it decreases with n.

Having $\mathcal{A}$, we generate a positive definite X and set $b = \mathcal{A}X$. Similarly, we select a positive definite Z and a vector y and set $C = Z + \mathcal{A}^* y$, so we have strong duality. The generator is also available on [Web]. To make the data reproducible, we initialize the random number generator with the parameter *seed*.

In Table 1, we provide a collection of instances. Aside from the parameters $n, m, p, seed$ used to generate the instances, we include:

- the time (in seconds) to compute the Cholesky factor of $\mathcal{A}\mathcal{A}^*$,
- the time (in seconds) to reach the required relative accuracy $\varepsilon = 10^{-7}$,
- the optimal value of the SDP problems.

Observe that the computing time for the Cholesky factorization cannot be neglected in general, and that it varies a lot according to the construction and sparsity of the matrix $\mathcal{A}$ (see comments above).

Second generator. When both n and m get larger, the bottleneck of our approach is the computation of the Cholesky factor of $\mathcal{A}\mathcal{A}^*$. In order to experiment also with larger instances, we have set up another generator for random instances which generates $\mathcal{A}$ through the QR decomposition of $\mathcal{A}^* = QR$. Here we select orthogonal columns in Q and select an upper triangular matrix R at random, with a prespecified density (=proportion of nonzero entries in the upper triangle). Again this generator is available online at [Web].

In Table 2 we collect some representative results. We ran the algorithm for 300 iterations and report the accuracy level reached. It was always below 10^{-6}. We manage to solve these instances with reasonable computational effort. We emphasize however, that these results are only achievable because the Cholesky factorization $\mathcal{A}\mathcal{A}^* = R^T R$ is given as part of the input. Computing the Cholesky factor of the smallest instance (seed=4004010) failed on our machine due to lack of memory.

5.3 Comparison with other methods

There are only a few methods available which are capable of dealing with large-scale SDP. The website `http://plato.asu.edu/bench.html` maintained by Hans Mittelmann gives an overview of the current state of the art in software to solve large SDP problems.

In Table 3, we compare our method with the low-rank approach SDPLR of [BM03]. Since this is essentially a first order nonlinear optimization approach, it does not easily reach the same level of accuracy as our method. We have set a time limit of 3600 seconds and report the final relative accuracy. We also compare with the code of Kim Toh, described in [Toh04]. The results were provided to us by Kim Toh; the timings for this experiment were obtained on a Pentium 4 with 3.0 GHz, a machine that is slightly slower than ours. We first note that SDPLR is clearly outperformed by both other methods. We also see that as m gets larger, our method is substantially faster than the code of Toh.

At the time of final revisions of this paper, the preprint [ZST08] became publicly available, showing that a refined instantiation of our regularization algorithm leads to more powerful computational results.

5.4 The Lovász theta number

In [PRW06], the boundary point method was applied to the SDP problem underlying the Lovász theta number of a graph. We come back now on this type of SDP with the present approach and we compute challenging instances.

Given a graph G (and its complementary $\overline{G}$) with the set of vertices $V(G)$ and the set of edges $E(G)$, $n = |V(G)|$, the Lovász theta number $\vartheta(G)$ of G is defined as the optimal value of the following SDP problem

$$\vartheta(G) := \left\{ \begin{array}{ll} \max & \langle J, X \rangle \\ & X_{ij} = 0, \quad \forall ij \in E(G), \\ & \operatorname{trace} X = 1, \quad X \succeq 0. \end{array} \right.$$

n	m	p	seed	Cholesky time	Algorithm time	obj-value
300	20000	3	3002030	37	63	761.352
300	25000	3	3002530	217	127	73.838
300	10000	4	3001040	83	97	165.974
400	30000	3	4003030	35	118	1072.139
400	40000	3	4004030	672	202	805.768
400	15000	4	4001540	222	195	-655.000
500	30000	3	5003030	1	201	1107.626
500	40000	3	5004030	19	198	816.610
500	50000	3	5005030	277	254	364.945
500	20000	4	5002040	276	323	328.004
600	40000	3	6004030	1	395	306.617
600	50000	3	6005030	7	372	-386.413
600	60000	3	6006030	93	345	641.736
600	20000	4	6002040	60	485	1045.269
700	50000	3	7005030	1	591	313.202
700	70000	3	7007030	27	507	-369.559
700	90000	3	7009030	749	601	-26.755
800	70000	3	8007030	1	793	2331.395
800	100000	3	80010030	219	805	2259.288
800	110000	3	80011030	739	836	1857.920
900	100000	3	90010030	7	1047	954.222
900	140000	3	90014030	1672	1340	2319.830
1000	100000	3	100010030	1	1600	3096.361
1000	150000	3	100015030	420	1851	1052.887

Table 1: Randomly generated SDPs. Columns 1 to 4 provide the parameters to generate the data. The columns "time" give the time (in seconds) to compute the Cholesky factor, and the time of Algorithm 5.1. In the last column we add the optimal value computed. The stopping criteria is error δ being below 10^{-7}.

n	m	$dens$	seed	time	error	obj-value
400	40000	$10/m$	4004010	200	1e-6	8018.86
500	60000	$10/m$	5006010	382	1e-6	10201.85
600	80000	$10/m$	8008010	714	1e-6	12465.07
700	100000	$10/m$	70010010	1176	.5e-6	13063.73
800	130000	$10/m$	80013010	1550	1e-6	13707.81
900	150000	$10/m$	90015010	2800	.5e-6	15964.01

Table 2: Randomly generated SDPs with known Cholesky factor of $\mathcal{A}\mathcal{A}^*$. We stop the algorithm after 300 iterations. We provide in columns 1 to 4 the parameters to generate the data, and in the last three columns the time (in seconds) to solve the problem, the error and the optimal value.

	SDPLR		Toh		our
seed	time	error	time	error	time
3002030	686	1.1e-5	204	1.5e-4	63
3002530	1079	1.1e-5	958	2.5e-7	127
3001040	123	1.2e-5	159	3.7e-8	97
4003030	1880	7.3e-6	1000	0.6e-7	118
4004030	3055	5.0e-6	425	0.2e-5	202
4001540	299	1.0e-5	372	0.4e-7	195
5003030	2165	7.8e-6	1309	0.5e-7	201
5004030	3600	8.8e-6	1668	0.4e-7	198
5005030	3600	1.6e-5	1207	0.1e-5	254
5002040	347	1.1e-5	644	0.9e-7	323
6004030	3499	5.8e-6	1658	0.7e-7	395
6005030	3600	3.8e-5	2009	0.5e-5	372
6006030	3600	4.1e-5	1630	0.5e-6	345
6002040	600	4.3e-5	838	0.9e-7	485
7005030	3600	3.4e-5	2696	0.8e-7	591
7007030	3600	6.3e-5	4065	0.3e-7	507
8007030	3600	4.0e-5	2951	1.0e-7	793

Table 3: Comparison on randomly generated SDPs. We compare SDPLR [BM03] and the code of Toh [Toh04] against our code on data sets from Table 1.

graph name	n	$\lvert E(\overline{G})\rvert$	$\vartheta(\overline{G})$	$\omega(G)$	$\lvert E(G)\rvert$	$\vartheta(G)$
keller5	776	74710	31.000	27	225990	31.000
keller6	3361	1026582	63.000	≥ 59	4619898	63.000
san1000	1000	249000	15.000	15	250500	67.000
san400-07.3	400	23940	22.000	22	55860	19.000
brock400-1	400	20077	39.702	27	59723	10.388
brock800-1	800	112095	42.222	23	207505	19.233
p-hat500-1	500	93181	13.074	9	31569	58.036
p-hat1000-3	1000	127754	84.80	≥ 68	371746	18.23
p-hat1500-3	1500	277006	115.44	≥ 94	847244	21.52

Table 4: DIMACS graphs and theta numbers

where J is again the matrix of all ones. Lovász [Lov79] showed that the polynomially computable number $\vartheta(\overline{G})$ separates the clique number $\omega(G)$ from the chromatic number $\chi(G)$, i.e., it holds

$$\omega(G) \leq \vartheta(\overline{G}) \leq \chi(G).$$

Both numbers $\omega(G)$ and $\chi(G)$ are NP-complete to compute and in fact even hard to approximate.

Let us take some graphs from the DIMACS collection [JT96] (available at `ftp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/clique`). The clique number for most of these instances is known, we compute their theta number. We do not consider instances having special structure, like Hamming or Johnson graphs, because these allow computational simplifications leading to purely linear programs, see [Sch79]. In [DR07] it is observed that $\vartheta(G)$ can be computed efficiently by interior point methods in case that either $|E(G)|$ or $|E(\overline{G})|$ is not too large. Hence we consider only instances where $|V(G)| > 300$ and both $|E(G)|$ and $|E(\overline{G})| > 10000$.

For the collection of instances of Table 4, ϑ has not yet been able to be computed before. We provide for them approximations of values of both $\vartheta(G)$ and $\vartheta(\overline{G})$. The number of outer iterations ranges from 200 to 2000. The computation times are for the smallest graphs ($n = 400$) some minutes, for the graphs with sizes from $n = 500$ to $n = 1500$ several hours, and for the graph keller6 ($n = 3361$) one has to allow days in order to reach the desired tolerance.

It turned out that our method worked fine in most of the cases. We noticed nevertheless an extremely slow convergence in case of the Sanchis graphs (e.g., san1000, san400-07.3). One reason for this may lie in the fact that the resulting SDPs have optimal solutions with rank-one.

5.5 Conclusions and perspectives on experiments

We propose a class of regularization methods for solving linear SDP problems, having in mind SDP problems with a large number of constraints. In the previous sections, we have presented and studied in theory these methods and in this

last section, we presented numerical experiments which show that in practice our approach compares favourably on several classes of SDPs with the most efficient currently available SDP solvers. Our approach has nevertheless not yet reached the level of refinement necessary to be competitive on a very large range of problems. The main issues are the choice of the classical optimization algorithm for the inner problem (in other words, with our notation, the choice of W and τ), and moreover the stopping rule for this inner algorithm. As usual with regularization methods, an important issue (still rather mysterious) is the proper choice of the regularization parameter, and the way to update it. So the clarification and generalization of this paper open the way for improvements: there is large room for further research in this area, in both theory and practice. Moreover, in order to be widely applicable, it is necessary to set up the method so that it can handle several blocks of semidefinite matrices, as well as inequality constraints. All this is also the topic of further research and development.

Acknowledgement

We thank Claude Lemaréchal for many fruitful discussions on the approach discussed in this paper. We also thank Kim Toh and Sam Burer for making their codes available to us. Moreover, Kim Toh was kind enough to experiment with our random number generators and to run some of the instances independently on his machine. Finally, we appreciate the constructive comments of two anonymous referees, leading to the present version of the paper.

References

[AHU59] K. Arrow, L. Hurwicz, and H. Uzawa. *Studies in Linear and Nonlinear Programming*. Stanford University Press, 1959.

[Ber82] D.P. Bertsekas. *Constrained Optimization and Lagrange multiplier methods*. Academic Press, 1982.

[BKL66] R. Bellman, R. Kalaba, and J. Lockett. *Numerical Inversion of the Laplace Transform*. Elsevier, 1966.

[BM03] S. Burer and R.D.C Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming (Series B)*, 95:329–357, 2003.

[BV06] S. Burer and D. Vandenbussche. Solving lift-and-project relaxations of binary integer programs. *SIAM Journal on Optimization*, 16(3):726–750 (electronic), 2006.

[DR07] I. Dukanovic and F. Rendl. Semidefinite programming relaxations for graph coloring and maximal clique problems. *Mathematical Programming*, 109:345–365, 2007.

[Hes69] M.R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4:303–320, 1969.

[Hig02] N. Higham. Computing a nearest symmetric correlation matrix - a problem from finance. *IMA Journal of Numerical Analysis*, 22(3):329–343, 2002.

[HR00] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10(3):673–696, 2000.

[HUL93] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer Verlag, Heidelberg, 1993. Two volumes.

[JR08] F. Jarre and F. Rendl. An augmented primal-dual method for linear conic problems. *SIAM Journal on Optimization*, 2008. To appear.

[JT96] D.J. Johnson and M.A. Trick, editors. *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, Workshop, October 11-13, 1993*. American Mathematical Society, Boston, MA, USA, 1996.

[KS07] M. Kočvara and M. Stingl. On the solution of large-scale SDP problems by the modified barrier method using iterative solvers. *Mathematical Programming*, 109:413–444, 2007.

[LNM07] Z. Lu, A. Nemirovski, and R.D.C. Monteiro. Large-scale semidefinite programming via a saddle point Mirror-Prox algorithm. *Mathematical Programming*, 109(2-3, Ser. B):211–237, 2007.

[Lov79] L. Lovász. On the shannon capacity of a graph. *IEEE Trans. Inform. Theory*, 25:1–7, 1979.

[Mal04] J. Malick. A dual approach to semidefinite least-squares problems. *SIAM Journal on Matrix Analysis and Applications*, 26, Number 1:272–284, 2004.

[Mal05] J. Malick. *Algorithmique en analyse variationnelle et optimisation semidéfinie*. PhD thesis, Université Joseph Fourier de Grenoble (France), November 25, 2005.

[Mar70] B. Martinet. Régularisation d'inéquations variationnelles par approximations successives. *Revue Française d'Informatique et Recherche Opérationnelle*, R-3:154–179, 1970.

[Mit03] D. Mittelmann. An independent benchmarking of sdp and socp solvers. *Mathematical Programming B*, 95:407–430, 2003.

[Mon03] R.D.C. Monteiro. First- and second-order methods for semidefinite programming. *Mathematical Programming*, 97:663–678, 2003.

[Pow69] M.J.D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*. Academic Press, London, New York, 1969.

[PRW06] J. Povh, F. Rendl, and A. Wiegele. A boundary point method to solve semidefinite programs. *Computing*, 78:277–286, 2006.

[PT72] B.T. Polyak and N.V. Tretjakov. On an iterative method for linear programming and its economical interpretations. *Èkonom. i Mat. Methody*, 8:740–751, 1972.

[QS06] H. Qi and D. Sun. Quadratic convergence and numerical experiments of Newton's method for computing the nearest correlation matrix. *SIAM Journal on Matrix Analysis and Applications*, 28:360–385, 2006.

[Ren05] F. Rendl. A boundary point method to solve semidefinite programs. Report 2/2005, Annals of Oberwolfach, Springer, 2005.

[Roc76a] R.T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1:97–116, 1976.

[Roc76b] R.T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14:877–898, 1976.

[Sch79] A. Schrijver. A comparison of the Delsarte and Lovász bounds. *IEEE Transactions on Information Theory*, IT-25:425–429, 1979.

[Tod01] M.J. Todd. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001.

[Toh04] K.C. Toh. Solving large scale semidefinite programs via an iterative solver on the augmented systems. *SIAM Journal on Optimization*, 14:670–698, 2004.

[Web] Website. `http://www.math.uni-klu.ac.at/or/Software`.

[WSV00] H. Wolkowicz, R. Saigal, and L. Vandenberghe. *Handbook of Semidefinite Programming*. Kluwer, 2000.

[ZST08] X. Zhao, D. Sun, and K. Toh. A Newton-CG augmented Lagrangian method for semidefinite programming. Technical report, National University of Singapore, March 2008.