Cahier du GERAD G-2012-12

A PRIMAL-DUAL REGULARIZED INTERIOR-POINT METHOD FOR SEMIDEFINITE PROGRAMMING

AHAD DEHGHANI, JEAN-LOUIS GOFFIN, AND DOMINIQUE ORBAN

Abstract. Interior-point methods in semidefinite programming (SDP) require the solution of a sequence of linear systems which are used to derive the search directions. Safeguards are typically required in order to handle rank-deficient Jacobians and free variables. We generalize the primal-dual regularization of Friedlander and Orban (2012) to SDP and show that it is possible to recover an optimal solution of the original primal-dual pair via inaccurate solves of a sequence of regularized SDPs for both the NT and dual HKM directions. Computationally, a sparse LDL^T factorization may be used on a sparse augmented system instead of the more costly symmetric indefinite factorization. Benefits of our approach include increased robustness and a simpler implementation. Our method does not require the constraints to be linearly independent and does not assume that Slater's condition holds. We report numerical experience on standard problems that illustrate our findings.

1. Introduction

In this paper, we consider the primal-dual pair of semidefinite programs in standard form

$$\text{(P)} \qquad \underset{X \in \mathbb{S}^n}{\text{minimize}} \quad C \bullet X \quad \text{subject to} \quad \mathcal{A}X = b, \quad X \succcurlyeq 0,$$

$$\text{(D)} \qquad \underset{y \in \mathbb{R}^m,\ Z \in \mathbb{S}^n}{\text{maximize}} \quad b^T y \quad \text{subject to} \quad \mathcal{A}^* y + Z = C, \quad Z \succcurlyeq 0,$$

where $\mathbb{S}^n$ is the set of symmetric matrices in $\mathbb{R}^{n \times n}$, C, X, $Z \in \mathbb{S}^n$, $b \in \mathbb{R}^n$, $C \bullet X := \operatorname{tr}(CX)$, $\mathcal{A}$ is a linear operator on $\mathbb{S}^n$, $\mathcal{A}^*$ is the adjoint of $\mathcal{A}$, and $X \succcurlyeq 0$ means that X is positive semidefinite. The purpose of our regularization is to address degeneracy issues that cause difficulties in interior-point methods for (P)–(D). We justify later why there is no loss of generality in considering problems in standard form.

In interior-point methods for SDP, most of the computational cost lies in the solution of a symmetric indefinite system of linear equations that determines the search direction. At each iteration, a saddle-point system of the following form must be solved

$$\begin{bmatrix} -\mathcal{D} & \mathcal{A}^* \\ \mathcal{A} & 0 \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \tag{1.1}$$

Date: December 2, 2012.

2010 *Mathematics Subject Classification.* 90C06, 90C20, 90C30, 90C51, 90C53, 90C55, 65F10, 65F50.

Key words and phrases. Semidefinite Programming, Regularization, Interior-Point Method, Symmetric Quasi-Definite, Degeneracy, Slater's Condition .

Research partially supported by NSERC Discovery Grant 4152.

Research partially supported by NSERC Discovery Grant 299010-04.

where $\mathcal{D}$ is an invertible linear operator on $\mathbb{S}^n$, not necessarily symmetric. The operator $\mathcal{D}$ and the right-hand side (f, g) change at each iteration. Since $\mathcal{D}$ is invertible, one step of block elimination shows that Δy may alternatively be computed via the *normal equations*

$$\mathcal{A}\mathcal{D}^{-1}\mathcal{A}^*\Delta y = \mathcal{A}\mathcal{D}^{-1}f + g, \tag{1.2}$$

after which we recover $\Delta X = \mathcal{D}^{-1}(\mathcal{A}^*\Delta y - f)$. Whenever $\mathcal{D}$ is symmetric and positive definite and $\mathcal{A}$ is onto, the advantage of (1.2) over (1.1) is that its coefficient operator is symmetric and positive definite, while that of (1.1), though nonsingular, is indefinite. In numerical linear algebra terms, (1.2) may be solved using an iterative method such as the conjugate gradient method or using the Cholesky factorization of its coefficient matrix. By contrast, if (1.1) were to be solved with an iterative method, a method for indefinite systems must be employed, such as MINRES or SYMMLQ (Paige and Saunders, 1975). Alternatively, the symmetric indefinite factorization of its coefficient matrix, such as that of Duff (2004), may be computed. The situation worsens when $\mathcal{A}$ is not onto, for then the coefficient matrix of both (1.1) and (1.2) is singular. When (P) contains free real variables, $\mathcal{D}$ is singular and only a partial block elimination is possible, leaving a smaller indefinite linear system to be solved. We propose a primal-dual regularization of (P)–(D) that allows us to dispense with the assumption that $\mathcal{A}$ is onto and that produces a nonsingular $\mathcal{D}$ even in the presence of free variables. The regularization is *exact* in the sense that it allows to recover a primal-dual solution of (P)–(D).

Our regularization does not change the solutions of (P)–(D). Its net effect on the interior-point method is to change $\mathcal{D}$ into $\mathcal{D} + \rho I$ where $\rho > 0$ is a parameter and I is the identity operator, and to change the $(2, 2)$ block of (1.1) from zero to δI, where $\delta > 0$ is a second parameter. This effectively always renders (1.1) nonsingular. The block elimination is always possible, yielding

$$(\mathcal{A}(\mathcal{D} + \rho I)^{-1}\mathcal{A}^* + \delta I)\Delta y = \mathcal{A}(\mathcal{D} + \rho I)^{-1}f + g. \tag{1.3}$$

Except in the case where free variables are present, our numerical experience is with this latter system because of the relative rigidity of the software that we modified to perform our numerical tests. However, using the perturbed augmented system has computational advantages. Indeed a matrix of the form

$$\begin{bmatrix} -\bar{\mathcal{D}} & \mathcal{A}^* \\ \mathcal{A} & \mathcal{G} \end{bmatrix},$$

where both $\bar{\mathcal{D}}$ and $\mathcal{G}$ are symmetric and positive definite is termed *quasi definite* and has the important property of being strongly factorizable (Vanderbei, 1995). In other words, any symmetric permutation of this matrix possesses a LDL^T factorization with L unit lower triangular and D diagonal. A simpler factorization scheme than the symmetric indefinite factorization may thus be employed.

The primal-dual regularization, closely related to augmented Lagrangian and proximal-point methods, is directly inspired by recent work of Friedlander and Orban (2012) in convex quadratic programming. The convergence analysis does not carry over trivially, though all results still hold.

The rest of this paper is organized as follows. Section 2 discusses the motivation for this work and §3 establishes the notation. Section 4 presents the primal-dual regularization of (P)–(D). Our interior-point method for regularized problems appears in §5. In §6 and §7, we review the central path, introduce its neighborhood

used by our algorithms, and emphasize the changes required when considering problems with free variables. A prelude to the convergence results of the algorithm appears in §8. Convergence of a variant with fixed regularization parameters is given in §9 while convergence of a variant with decreasing regularization parameters appears in §10. Numerical experience is reported in §11. Closing comments and conclusions appear in §12.

2. Motivation and Related Work

Semidefinite programming (SDP) has been an active topic in optimization for almost two decades. SDP has numerous applications in computational geometry, experiment design, information and communication theory, statistics, control theory, eigenvalue maximization, and linear matrix inequalities, to name a few (Vandenberghe and Boyd, 1996, 1999). It has also been recognized in combinatorial optimization as a valuable technique for obtaining bounds on the solution of NP-hard problems (Alizadeh, 1995; Helmberg, 2000). Two handbooks cover theory, algorithms, applications, and software for SDP (Wolkowicz et al., 2000; Anjos and Lasserre, 2012).

The approach most closely related to ours is the primal regularization for conic optimization of Anjos and Burer (2007), who provide a convergence proof for a primal regularization scheme proposed by Mészáros (1998). Their main motivation is to handle free variables. Our convergence analysis and that of Friedlander and Orban (2012) improve on that of Anjos and Burer (2007) in at least three respects. Firsly, they only consider primal regularization, while we consider both primal and dual regularization. Secondly, they explicitly assume boundedness of the Newton direction. Finally, they assume that $\mathcal{A}$ is onto. We make neither of the last two assumptions—dual regularization allows us to dispense with the assumption on $\mathcal{A}$ while the combination of primal and dual regularization ensures boundedness of the Newton direction. In addition, their regularization parameter update occurs if the Newton direction just computed violates a certain condition. A clear disadvantage is that each time the parameter value is changed, the Newton direction must be recomputed.

3. Notation

For any matrix $G \in \mathbb{R}^{m\times n}$, we use $\sigma_n(G)$ and $\sigma_1(G)$ to denote the smallest and the largest singular values of G, respectively. If $G \in \mathbb{S}^n$, we use $\lambda_n(G)$ and $\lambda_1(G)$ to denote the smallest and the largest eigenvalues of G. The Frobenius norm of any matrix G is denoted $\|G\|_F$ and the Euclidian norm of any vector x is simply denoted $\|x\|$. For $G \in \mathbb{S}^n$, we use $G \succcurlyeq 0$ and $G \succ 0$ to mean that G is positive semidefinite and positive definite, respectively.

If $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$ is a linear operator on $\mathbb{S}^n$, it is defined componentwise as $(\mathcal{A}X)_i := A_i \bullet X$, for $i = 1, \ldots, m$, and all $X \in \mathbb{S}^n$, where each $A_i \in \mathbb{S}^n$. The adjoint $\mathcal{A}^* : \mathbb{R}^m \to \mathbb{S}^n$ of $\mathcal{A}$ is defined as $\mathcal{A}^* y := \sum_{i=1}^m y_i A_i$ for all $y \in \mathbb{R}^m$. By definition, $y^T(\mathcal{A}X) = (\mathcal{A}^* y) \bullet X$.

In order to work with matrices instead of operators we use the following standard notation. For any $m \times n$ matrix M, $\mathbf{vec}\, M$ denotes the mn-vector obtained from stacking the columns of M on top of one another. For conciseness, we will use boldface $\mathbf{M}$ to denote $\mathbf{vec}\, M$ with the understanding that, e.g, $\mathbf{M}^{-1}$ stands for

$\mathbf{vec}(M^{-1})$. By definition, $\|M\|_F = \|\mathbf{M}\|$. For matrices $A_i \in \mathbb{S}^n$, $i = 1, \dots m$, we use

$$A^T := \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_m \end{bmatrix}.$$

For any two matrices $B \in \mathbb{R}^{m\times n}$ and $C \in \mathbb{R}^{p\times q}$, the Kronecker product of B and C, denoted by $B \otimes C$, is defined as a $(mp) \times (nq)$ block matrix consisting of mn blocks, the $ij-$th block being $b_{ij}C$. We collect key properties of the Kronecker product used in this paper in Appendix A.

4. A Primal-Dual Regularization

SDPs are convex problems in the sense that they consist in minimizing a linear function over a convex set, but they are not convex in the sense of nonlinear programming, because the condition $X \succcurlyeq 0$ amounts to a number of nonconvex inequality constraints. In this section, we first recall useful definitions from (Luenberger, 1997) that generalize the notion of inequality, convexity, and convex programs to vector spaces. We next state our primal-dual regularization for SDPs.

Definition 4.1 (Generalized inequality)**.** Let P be a convex cone in a vector space V. For any x, $y \in V$ we write $x \succcurlyeq_P y$ if $x - y \in P$. The cone P defining this relation is called the positive cone in V. To simplify notation and whenever there is no ambiguity, we omit the index P and write $x \succcurlyeq y$ for $x \succcurlyeq_P y$. The cone P is said to be a pointed cone if whenever $x \in P$ and $-x \in P$ then $x = 0$.

It is easy to verify that $\succcurlyeq$ is transitive and reflexive. It is also antisymmetric if P is pointed. Since we have a generalized definition of inequality between vectors, it is possible to generalize the notion of convexity to mappings.

Definition 4.2 (Generalized convexity)**.** Let V be a vector space and let W be a vector space having a cone P specified as the positive cone. A mapping $g : V \to W$ is said to be convex if its domain Ω is a convex set and if for all $\alpha \in [0, 1]$,

$$g(\alpha x_1 + (1 - \alpha)x_2) \preccurlyeq \alpha g(x_1) + (1 - \alpha)g(x_2) \quad \text{for all } x_1, x_2 \in \Omega.$$

A general convex program on a vector space V can be stated as

$$\underset{x\in\Omega}{\text{minimize}} \;\; f(x) \quad \text{subject to } \; g(x) \preccurlyeq 0, \; Bx = b,$$

where f is a real-valued convex functional defined on a convex subset of V, g is the convex map introduced in Definition 4.2, and $B : V \to W$ is a linear operator. In the sense above, (P) and the following regularization of it are convex programs:

$$\begin{aligned} \underset{X\in\mathbb{S}^n,\; r\in\mathbb{R}^m}{\text{minimize}} \quad & C \bullet X + \tfrac{1}{2}\rho\|X - X_k\|_F^2 + \tfrac{1}{2}\delta\|r + y_k\|^2 \\ \text{subject to} \quad & \mathcal{A}X + \delta r = b, \; X \succcurlyeq 0, \end{aligned} \tag{4.1}$$

where $\rho \geq 0$ and $\delta \geq 0$ are regularization parameters, and X_k and y_k are current estimates of primal and dual solutions of (P) and (D), X^* and y^*, respectively. The relevant cone P in this case is the semidefinite cone.

Friedlander and Orban (2012) derive the regularized problem (4.1) by using a proximal-point approach (Rockafellar, 1976a) to solve (P)—which explains the addition of the term $\|X - X_k\|_F^2$ in the objective—and subsequently applying an augmented Lagrangian method to the equality constraints of the resulting problem. The peculiarity of the augmented Lagrangian here is that equality constraints are reintroduced by way of the new variables r. This serves the purpose of emphasizing the similarity between the regularized and original problems but is also more natural

in the context of the interior-point method of the next section. Rockafellar (1976b) refers to the combination of proximal-point and augmented-Lagrangian terms as the *proximal method of multipliers*. An important distinction however is that we will not attempt to solve (4.1), but merely use it to compute a single Newton step before moving on to the next subproblem—one that contains regularization terms involving X_{k+1} and y_{k+1} and, possibly, updated parameters ρ and δ.

We now illustrate the strength of the primal-dual regularization, namely that the dual of (4.1) is an identical regularization of (D).

The regularized problem (4.1) is convex in the variable (X, r) and always strictly feasible when $\delta > 0$. The dual of this problem can be derived by duality theory for general convex problems (Luenberger, 1997, Chapter 8, Theorem 1). Letting $v := (X, r, y, Z)$, the Lagrangian of (4.1) may be written

$$\mathcal{L}_{\rho,\delta}(v) := C \bullet X + \tfrac{1}{2}\rho\|X - X_k\|_F^2 + \tfrac{1}{2}\delta\|r + y_k\|^2 - y^T(\mathcal{A}X + \delta r - b) - Z \bullet X.$$

Since $\mathcal{L}_{\rho,\delta}(v)$ is convex with respect to (X, r), minimizing it in (X, r) is equivalent to setting its gradient to zero, that is

$$C + \rho(X - X_k) - \mathcal{A}^* y - Z = 0, \tag{4.2a}$$

$$\delta(r + y_k) - \delta y = 0. \tag{4.2b}$$

Using (4.2) and introducing $S := X - X_k$, we may rewrite

$$\begin{aligned}
\mathcal{L}_{\rho,\delta}(v) &= b^T y - \rho S \bullet X + \tfrac{1}{2}\rho\,\|S\|_F^2 - \tfrac{1}{2}\delta\|y - y_k\|^2 + \tfrac{1}{2}\delta\|y_k\|^2 \\
&= b^T y - \rho S \bullet (S + X_k) + \tfrac{1}{2}\rho\,\|S\|_F^2 - \tfrac{1}{2}\delta\|y - y_k\|^2 + \tfrac{1}{2}\delta\|y_k\|^2 \\
&= b^T y - \tfrac{1}{2}\rho\,(S \bullet S + 2S \bullet X_k) - \tfrac{1}{2}\delta\|y - y_k\|^2 + \tfrac{1}{2}\delta\|y_k\|^2 \\
&= b^T y - \tfrac{1}{2}\rho\,\|S + X_k\|_F^2 - \tfrac{1}{2}\delta\|y - y_k\|^2 + \tfrac{1}{2}\rho\,\|X_k\|_F^2 + \tfrac{1}{2}\delta\|y_k\|^2.
\end{aligned}$$

Since the last two terms in the last equality are constant, the dual of (4.1) may be written

$$\begin{aligned}
&\underset{S,y,Z}{\text{maximize}} && b^T y - \tfrac{1}{2}\rho\|S + X_k\|_F^2 - \tfrac{1}{2}\delta\|y - y_k\|^2 \\
&\text{subject to} && \mathcal{A}^* y + Z - \rho S = C,\ Z \succcurlyeq 0.
\end{aligned} \tag{4.3}$$

Note that upon letting $\rho = \delta = 0$, (4.1) and (4.3) simply reduce to (P) and (D).

It is easy to see that for $\rho > 0$ and $\delta > 0$, both (4.1) and (4.3) are strictly feasible. Therefore, strong duality always applies to them, even though it may not be the case for (P) and (D).

5. An Interior-Point Method for the Regularized SDP

In this section, we develop a primal-dual interior point method to solve the primal and dual regularizations (4.1) and (4.3) of (P) and (D) simultaneously. The logarithmic barrier problem associated to (4.1) is

$$\begin{aligned}
&\underset{X \in \mathbb{S}^n, r \in \mathbb{R}^m}{\text{minimize}} && C \bullet X + \tfrac{1}{2}\rho\|X - X_k\|_F^2 + \tfrac{1}{2}\delta\|r + y_k\|^2 - \tau \log(\det X) \\
&\text{subject to} && \mathcal{A}X + \delta r = b,
\end{aligned} \tag{5.1}$$

where $\tau > 0$ is the *barrier parameter*.

By strict convexity, (5.1) possesses a unique primal solution (X_τ, r_τ). Since the equality constraints are onto, there also exists a unique dual solution y_τ. Since the dual of (5.1) is the logarithmic barrier problem associated to (4.3) and is also strictly convex, there exists a unique dual solution $Z_\tau = \tau X^{-1}$. The *central path* is

defined as the set of such exact solutions when $\tau = \mu(X, Z) := (X \bullet Z)/n$. For any $\tau > 0$, upon letting $Z := \tau X^{-1}$ and $w := (X, r, S, y, Z)$, the necessary and sufficient optimality conditions of (5.1) can be written as

$$\Psi(w;\tau) := \begin{bmatrix} C + \rho S - \mathcal{A}^* y - Z \\ \delta(r + y_k) - \delta y \\ \rho X - \rho(S + X_k) \\ \mathcal{A}X + \delta r - b \\ XZ - \tau I \end{bmatrix} = 0, \qquad (X, Z) \succ 0. \tag{5.2}$$

Let (X_k, y_k) be temporarily fixed. A primal-dual interior-point method for the regularized primal-dual pair (4.1)—(4.3) is based on applying Newton's method to a sequence of nonlinear systems of the form

$$\Psi(w;\sigma_k\mu_k) = 0, \qquad (X, Z) \succ 0. \tag{5.3}$$

where $\sigma_k \in [0, 1]$ is the *centering parameter*, and $\mu_k := X_k \bullet Z_k/n > 0$ is the current duality measure. When applying Newton's method to (5.3) an immediate difficulty is that the resulting $X + \Delta X$ must be symmetric. Looking at the last block equation of (5.2), XZ is usually not symmetric even if X and Z are, so there is no guarantee that ΔX will be symmetric. A first remedy consists in identifying an operator $H : \mathbb{R}^{n\times n} \to \mathbb{S}^n$ such that $H(XZ) = \tau I$ if and only if $XZ = \tau I$ provided X and Z are symmetric. The last block equation of (5.2) is then replaced by $H(XZ) - \tau I$ and subsequently linearized (Monteiro, 1997; Monteiro and Zhang, 1998; Alizadeh et al., 1998; Zhang, 1998). A second approach consists in applying Newton's method to (5.2) without modification and subsequently symmetrizing ΔX (Helmberg et al., 1996; Kojima et al., 1997). The resulting direction may unfortunately not be a descent direction. In this paper, we follow the former strategy. As we will see, if ΔX is symmetric then both ΔZ and ΔS are symmetric as well. Upon performing the substitution in the last block equation (5.3), a Newton step Δw from w_k solves the linear system

$$\begin{bmatrix} 0 & 0 & \rho I & -\mathcal{A}^* & -I \\ 0 & \delta I & 0 & -\delta I & 0 \\ \rho I & 0 & -\rho I & 0 & 0 \\ \mathcal{A} & \delta I & 0 & 0 & 0 \\ \mathcal{E}_k & 0 & 0 & 0 & \mathcal{F}_k \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta r \\ \Delta S \\ \Delta y \\ \Delta Z \end{bmatrix} = - \begin{bmatrix} C + \rho S - \mathcal{A}^* y - Z \\ \delta(r + y_k) - \delta y \\ \rho X - \rho(S + X_k) \\ \mathcal{A}X + \delta r - b \\ H(XZ) - \tau I \end{bmatrix}, \tag{5.4}$$

where $\mathcal{E}_k := \nabla_X H(X_k Z_k)$ and $\mathcal{F}_k := \nabla_Z H(X_k Z_k)$. Reducing (5.4) by eliminating Δr and ΔS, we obtain

$$\begin{bmatrix} -\rho I & \mathcal{A}^* & I \\ \mathcal{A} & \delta I & 0 \\ \mathcal{E}_k & 0 & \mathcal{F}_k \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta y \\ \Delta Z \end{bmatrix} = \begin{bmatrix} C - \mathcal{A}^* y_k - Z_k \\ b - \mathcal{A}X_k \\ \tau I - H(X_k Z_k) \end{bmatrix}, \tag{5.5}$$

together with

$$\Delta r = -r + \Delta y, \tag{5.6a}$$

$$\Delta S = -S + \Delta X. \tag{5.6b}$$

If $\mathcal{F}_k$ is a nonsingular operator on $\mathbb{S}^n$ we can further eliminate ΔZ from the last row of (5.5) to obtain

$$\Delta Z = \mathcal{F}_k^{-1} \left(\tau I - H(X_k Z_k) - \mathcal{E}_k \Delta X\right) \tag{5.7}$$

and

$$\begin{bmatrix} -(\mathcal{F}_k^{-1}\mathcal{E}_k + \rho I) & \mathcal{A}^* \\ \mathcal{A} & \delta I \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta y \end{bmatrix} = \begin{bmatrix} C - \mathcal{A}^* y_k - Z_k - \mathcal{F}_k^{-1}(\tau I - H(X_k Z_k)) \\ b - \mathcal{A} X_k \end{bmatrix}.$$

Note that if we apply an interior-point method directly to the original primal and dual problems (P) and (D), the linear system to be solved at each iteration has precisely the form above with $\rho = \delta = 0$. In addition, the right-hand side is independent of ρ and δ.

Performing an additional step of block elimination yields (1.3) with $\mathcal{D} = \mathcal{F}_k^{-1}\mathcal{E}_k$, $f = C - \mathcal{A}^* y_k - Z_k - \mathcal{F}_k^{-1}(\tau I - H(X_k Z_k))$ and $g = b - \mathcal{A} X_k$.

Zhang (1998) introduces a general symmetrization scheme based on a similarity transformation as the operator $H_P : \mathbb{R}^{n\times n} \to \mathbb{S}^n$ defined by

$$H_P(M) := \tfrac{1}{2}(PMP^{-1} + (PMP^{-1})^T), \tag{5.8}$$

where P is any nonsingular matrix. Zhang (1998, Proposition 4.1) shows that for any matrix M with real eigenvalues, $M = \tau I$ is equivalent to $H_P(M) = \tau I$ for any nonsingular matrix P. In the present context, it is easy to see that XZ has real eigenvalues since $XZ = X^{\frac{1}{2}}(X^{\frac{1}{2}} Z X^{\frac{1}{2}})X^{-\frac{1}{2}}$. Therefore, the central path may be equivalently defined as $H_P(XZ) = \tau I$ for any nonsingular P.

Different choices of P lead to different Newton directions. Alizadeh et al. (1998) set $P = I$ to find the so-called AHO direction. Zhang (1998) uses $P = Z^{\frac{1}{2}}$ and establishes the complexity analysis of some path following methods, including an infeasible long-step path-following method. Monteiro (1997) uses $P = Z^{\frac{1}{2}}$ and $P = X^{-\frac{1}{2}}$ to establish polynomial complexity of the short-step feasible path-following method. There are many other choices of P as well as other kinds of symmetrization used to find a suitable search direction in SDP. For a comprehensive discussion of these search directions we refer the reader to the compendium of Todd (1999). In this paper, our analysis is based on two well-known directions: the dual HKM direction, with $P = X^{-\frac{1}{2}}$, and the Nesterov-Todd (NT) direction, with $P = W^{-\frac{1}{2}}$, where

$$W := X^{\frac{1}{2}}(X^{\frac{1}{2}} Z X^{\frac{1}{2}})^{-\frac{1}{2}} X^{\frac{1}{2}} = Z^{-\frac{1}{2}}(Z^{\frac{1}{2}} X Z^{\frac{1}{2}})^{\frac{1}{2}} Z^{-\frac{1}{2}}. \tag{5.9}$$

In order to cast (5.5) in matrix form, we need a representation of the operators $\mathcal{E}$ and $\mathcal{F}$. The first and the second rows of (5.5) are equivalent to

$$-\rho \mathbf{\Delta X} + A^T \Delta y + \mathbf{\Delta Z} = \mathbf{C} - A^T y_k - \mathbf{Z}_k \tag{5.10}$$

$$A\mathbf{\Delta X} + \delta \Delta y = b - A\mathbf{X}_k, \tag{5.11}$$

respectively.

In the next section, we use properties of the Kronecker product (see Appendix A) to translate the third row of (5.5) to matrix form.

6. Linearization

For a given constant nonsingular matrix P, applying the similarity transformation H_P to the perturbed complementarity condition and subsequently linearizing is equivalent to first linearizing the perturbed complementarity condition and then applying H_P. A direct linearization of $XZ - \sigma_k \mu_k I = 0$ yields

$$XZ + \Delta X Z + X \Delta Z = \sigma_k \mu_k I. \tag{6.1}$$

The dual HKM direction is obtained by applying the transformation (5.8) with $P = X^{-\frac{1}{2}}$ to (6.1). Upon pre- and post-multiplying $H_P(XZ + X\Delta Z + \Delta XZ) = \sigma_k\mu_k I$ by $\sqrt{2}X^{\frac{1}{2}}$, we obtain

$$\begin{aligned} 2X(\Delta Z)X + XZ(\Delta X) + (\Delta X)ZX &= 2(\sigma_k\mu_k X - XZX) \\ &= 2X^{\frac{1}{2}}(\sigma_k\mu_k I - X^{\frac{1}{2}}ZX^{\frac{1}{2}})X^{\frac{1}{2}}, \end{aligned} \tag{6.2}$$

where we used the fact that ΔX and ΔZ are now symmetric. Using (A.1b), we can write (6.2) in Kronecker product notation as

$$2(X \otimes X)\mathbf{\Delta Z} + (XZ \otimes I + I \otimes XZ)\mathbf{\Delta X} = \mathbf{R_c}, \tag{6.3}$$

where

$$R_c := 2X^{\frac{1}{2}}(\sigma_k\mu_k I - X^{\frac{1}{2}}ZX^{\frac{1}{2}})X^{\frac{1}{2}}. \tag{6.4}$$

If we define

$$E := XZ \otimes I + I \otimes XZ \tag{6.5}$$

$$F := 2X \otimes X, \tag{6.6}$$

then (6.3) can be written more compactly as

$$E\mathbf{\Delta X} + F\mathbf{\Delta Z} = \mathbf{R_c}. \tag{6.7}$$

Using (5.10), (5.11) and (6.7), we can write (5.5) at $w = w_k$ in matrix form as

$$\begin{bmatrix} -\rho I & A^T & I \\ A & \delta I & 0 \\ E_k & 0 & F_k \end{bmatrix} \begin{bmatrix} \mathbf{\Delta X} \\ \Delta y \\ \mathbf{\Delta Z} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_d \\ b - A\mathbf{X}_k \\ \mathbf{R_c} \end{bmatrix}, \tag{6.8}$$

where $\mathbf{R}_d = \mathbf{C} - A^T y_k - \mathbf{Z}_k$. Eliminating $\mathbf{\Delta Z}$ yields $\mathbf{\Delta Z} = F_k^{-1}(\mathbf{R}_c - E_k\mathbf{\Delta X})$ and

$$\begin{bmatrix} -(F_k^{-1}E_k + \rho I) & A^T \\ A & \delta I \end{bmatrix} \begin{bmatrix} \mathbf{\Delta X} \\ \Delta y \end{bmatrix} = \begin{bmatrix} \mathbf{R}_d - F_k^{-1}\mathbf{R_c} \\ b - A\mathbf{X}_k, \end{bmatrix}, \tag{6.9}$$

where the remaining search directions are recovered via (5.6). Note that $F_k^{-1}E_k$ and F_k are symmetric and positive definite, but E_k may not be symmetric. Using the definitions of F_k, $\mathbf{R}_d$, and $\mathbf{R}_c$ we have

$$\begin{aligned} \mathbf{R}_d - F_k^{-1}\mathbf{R_c} &= \mathbf{C} - A^T y_k - \mathbf{Z}_k - \tfrac{1}{2}(X_k^{-1} \otimes X_k^{-1})\mathbf{R_c} \\ &= \mathbf{C} - A^T y_k - \mathbf{Z}_k - \mathbf{vec}(\sigma_k\mu_k X_k^{-1} - Z_k) \\ &= \mathbf{C} - A^T y_k - \sigma_k\mu_k \mathbf{X}_k^{-1}. \end{aligned}$$

Therefore, we can rewrite (6.9) as

$$\begin{bmatrix} -(F_k^{-1}E_k + \rho I) & A^T \\ A & \delta I \end{bmatrix} \begin{bmatrix} \mathbf{\Delta X} \\ \Delta y \end{bmatrix} = \begin{bmatrix} \mathbf{C} - A^T y_k - \sigma_k\mu_k\mathbf{X}_k^{-1} \\ b - A\mathbf{X}_k \end{bmatrix}. \tag{6.10}$$

Turning now to the NT direction, we use the scaling matrix $P = W^{-\frac{1}{2}}$. The perturbed complementarity condition may be written

$$W^{-1}\Delta X W^{-1} + \Delta Z = \sigma\mu X^{-1} - Z. \tag{6.11}$$

Following the methodology of Todd et al. (1998), a Newton step from w_k on the corresponding primal-dual conditions solves the system

$$\begin{bmatrix} -\rho I & A^T & I \\ A & \delta I & 0 \\ \bar{E}_k & 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{\Delta X} \\ \Delta y \\ \mathbf{\Delta Z} \end{bmatrix} = \begin{bmatrix} \mathbf{C} - A^T y_k - \mathbf{Z}_k \\ b - A\mathbf{X}_k \\ \sigma_k\mu_k\mathbf{X}_k^{-1} - \mathbf{Z}_k \end{bmatrix}, \tag{6.12}$$

where $\bar{E}_k := W_k^{-1} \otimes W_k^{-1}$. Eliminating $\mathbf{\Delta Z}$ now yields

$$\begin{bmatrix} -(\bar{E}_k + \rho I) & A^T \\ A & \delta I \end{bmatrix} \begin{bmatrix} \mathbf{\Delta X} \\ \Delta y \end{bmatrix} = \begin{bmatrix} \mathbf{C} - A^T y_k - \sigma_k \mu_k \mathbf{X}_k^{-1} \\ b - A\mathbf{X}_k \end{bmatrix} \tag{6.13}$$

together with $\mathbf{\Delta Z} = \sigma_k \mu_k \mathbf{X}_k^{-1} - \mathbf{Z}_k - \bar{E}_k \mathbf{\Delta X}$.

We use (6.10) and (6.13) to design algorithms in the next section. We end this section with two technical results.

The following lemma gives properties of the transformation H_P that are useful in the analysis of the algorithms of §9 and §10.

Lemma 6.1. *Let X, $Z \in \mathbb{R}^{n\times n}$ be symmetric, $P \in \mathbb{R}^{n\times n}$ be nonsingular, and $M \in \mathbb{R}^{n\times n}$ then*

1. *if X and Z are positive definite, then $\lambda(H_P(XZ)) = \lambda(XZ)$ for both $P = X^{-\frac{1}{2}}$ and $P = W^{-\frac{1}{2}}$,*
2. *$tr(H_P(M)) = tr(M)$,*
3. *$\|H_P(M)\|_F \leq \|M\|_F$.*

Proof. We establish the first claim. The second and third are obvious. When $P = X^{-\frac{1}{2}}$ we have $H_P(XZ) = \frac{1}{2}(X^{-\frac{1}{2}}XZX^{\frac{1}{2}} + X^{\frac{1}{2}}ZXX^{-\frac{1}{2}}) = X^{\frac{1}{2}}ZX^{\frac{1}{2}}$ and the result is obtained by using (A.1g). When $P = W^{-\frac{1}{2}}$, (5.9) implies that $WZW = X$ and hence $WZ = XW^{-1}$ and obviously $P^TP = W^{-1}$. Therefore, W and P are symmetric and

$$\begin{aligned} H_p(XZ) &= \tfrac{1}{2}(PXZP^{-1} + P^{-1}ZXP) = \tfrac{1}{2}P(XZ + P^{-2}ZXP^2)P^{-1} \\ &= \tfrac{1}{2}P(XZ + WZXW^{-1})P^{-1} = \tfrac{1}{2}P(XZ + WZWZ)P^{-1} \\ &= P(XZ)P^{-1}, \end{aligned}$$

which establishes that $H_P(XZ)$ and XZ are similar. □

The following lemma is due to Todd et al. (1998, Theorem 3.5) and states the relation between (6.11) and the linearization of the central path.

Lemma 6.2. *Let $P = W^{-\frac{1}{2}}$. Then ΔX and ΔZ satisfy (6.11) if and only if they satisfy $H_P(XZ + \Delta XZ + X\Delta Z) = \sigma\mu I$.*

6.1. **Effect of Free Variables.** In this section, we outline the changes required in the notation of the preceding sections to accommodate problems with free variables. Consider the primal-dual pair

$$\underset{X,\ t}{\text{minimize}} \quad C \bullet X \quad \text{subject to} \quad \mathcal{A}X + Bt = b, \qquad X \succcurlyeq 0, \tag{6.14a}$$

$$\underset{y,\ Z}{\text{maximize}} \quad b^T y \quad \text{subject to} \quad \mathcal{A}^* y + Z = C, \quad B^T y = 0, \quad Z \succcurlyeq 0. \tag{6.14b}$$

The regularized primal-dual pair takes the form

$$\begin{aligned} \underset{X,\ t,\ r}{\text{minimize}} \quad & C \bullet X + \tfrac{1}{2}\rho\|X - X_k\|_F^2 + \tfrac{1}{2}\rho_t\|t - t_k\|^2 + \tfrac{1}{2}\delta\|r + y_k\|^2 \\ \text{subject to} \quad & \mathcal{A}X + Bt + \delta r = b,\ X \succcurlyeq 0, \end{aligned} \tag{6.15a}$$

$$\begin{array}{ll} \underset{y,\ S,\ u,\ Z}{\text{maximize}} & b^T y - \frac{1}{2}\rho\|S + X_k\|^2 - \frac{1}{2}\rho_t\|u + t_k\|^2 - \frac{1}{2}\delta\|y - y_k\|^2 \\ \text{subject to} & \mathcal{A}^* y + Z - \rho S = C,\ B^T y - \rho_t u = 0,\ Z \succcurlyeq 0. \end{array} \tag{6.15b}$$

The Newton system corresponding to (6.14a) and (6.14b) that needs to be solved in Step 2 of Algorithms 7.1 and 7.2 has the form

$$\begin{bmatrix} -(Q+\rho I) & 0 & A^T \\ 0 & -\rho_t I & B^T \\ A & B & \delta I \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta t \\ \Delta y \end{bmatrix} = \begin{bmatrix} C - A^T - Z + F^{-1}H_P(XZ) \\ -B^T y \\ b - \mathcal{A}X - Bt \end{bmatrix},$$

along with formulae analogous to (5.6) to recover Δr, ΔS, Δu and ΔZ, and where Algorithm 7.1 uses $Q = F^{-1}E$ and Algorithm 7.2 uses $Q = \bar{E}$.

7. A Long-Step Path-Following Interior-Point Method

Let $M \in \mathbb{R}^{n\times n}$ be a real matrix. We denote the vector of eigenvalues of M by $\lambda(M) \in \mathbb{C}^n$. We define

$$w(\alpha) := w + \alpha\Delta w, \quad \mu(\alpha) := (X(\alpha) \bullet Z(\alpha))/n,$$

where Δw is defined by (5.5)–(5.6). Recall from the previous section that the central path $XZ = \mu I = (X \bullet Z)/nI$ is equivalently described as $\{(X, y, Z)) \mid H_P(XZ) = \mu I\}$.

Path-following methods generate iterates that remain within a neighborhood of the central path. We define a (iteration-dependent) neighborhood $\mathcal{N}_k$ of the central path as the set of points (X, r, S, y, Z) that satisfy a specific *subset* of the conditions

$$\gamma_C\mu e \le \lambda(H_P(XZ)) \le \bar{\gamma}_C\mu e, \tag{7.1a}$$

$$\|A\mathbf{X} + \delta_k r - b\| \le \gamma_P\mu, \tag{7.1b}$$

$$\|\mathbf{C} + \rho_k\mathbf{S} - A^T y - \mathbf{Z}\| \le \gamma_D\mu, \tag{7.1c}$$

$$\|\delta_k(r + y_k) - \delta_k y\| \le \gamma_R\mu, \tag{7.1d}$$

$$\|\rho_k\mathbf{X} - \rho_k(\mathbf{S} + \mathbf{X}_k)\| \le \gamma_S\mu, \tag{7.1e}$$

where $0 < \gamma_C < 1 < \bar{\gamma}_C \le +\infty$ and $(\gamma_P, \gamma_D, \gamma_R, \gamma_S) > 0$ are given constants. In particular, for a steplength α to yield an iterate lying in this neighborhood, we must have

$$\gamma_C\mu(\alpha)e \le \lambda(H_P(X(\alpha)Z(\alpha))) \le \bar{\gamma}_C\mu(\alpha)e. \tag{7.2}$$

Our interior-point scheme generates the next iterate w_{k+1} as follows. We compute $\Delta w = (\Delta X, \Delta r, \Delta S, \Delta y, \Delta Z)$ from either (6.9) or (6.13), and (5.6), and a steplength $\alpha_k \in (0, 1]$ such that

$$w_k(\alpha_k) := w_k + \alpha_k\Delta w \in \mathcal{N}_{k+1}. \tag{7.3}$$

Since the neighborhood $\mathcal{N}_{k+1}$ involves ρ_{k+1} and δ_{k+1}, appropriate new values for those parameters must be selected together with $\alpha_k \in (0, 1]$ to ensure that (7.3) is satisfied. The complete process is stated as Algorithm 7.1.

Algorithm 7.2 only differs from Algorithm 7.1 in Step 2, where we solve a system based on the NT direction. The definition of $\mathcal{N}$ in Step 0 of Algorithms 7.1 and 7.2 will be specified in Sections 9 and 10.

Algorithm 7.1 Primal-Dual Regularized Algorithm for the Dual HKM Direction

Step 0 (Initialize): Choose minimum and maximum centering parameters $0 < \sigma_{\min} \leq \sigma_{\max} < 1$, a constant $\sigma_{\max} < \beta < 1$. Choose proximity parameters $0 < \gamma_C < 1 < \bar{\gamma}_C$, $(\gamma_P, \gamma_D, \gamma_R, \gamma_S) > 0$ and define the neighborhood $\mathcal{N}$ as an appropriate subset of the conditions (7.1). Select initial regularization parameters $\rho_0 > 0$, $\delta_0 > 0$. Choose initial primal $X_0 \succ 0$, $r_0 \in \mathbb{R}^m$ and dual guesses $S_0 \in \mathbb{S}^n$, $y_0 \in \mathbb{R}^m$, $Z_0 \succ 0$ so that $w_0 \in \mathcal{N}_0$. Let $\mu_0 = X_0 \bullet Z_0/n$. Choose a tolerance $\varepsilon > 0$ and set $k = 0$.

Step 1 (Test convergence): If $\mu_k \leq \varepsilon$, declare convergence and stop. Otherwise, proceed to the next step.

Step 2 (Step computation): Choose a centering parameter $\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$. Compute the Newton step Δw_k from w_k by solving (6.10) and using (5.6).

Step 3 (Linesearch): Select $\delta_{k+1} \in (0, \delta_k]$ and $\rho_{k+1} \in (0, \rho_k]$ and compute α_k as the largest $\alpha \in (0, 1]$ such that

$$w_k(\alpha_k) \in \mathcal{N}_{k+1}, \tag{7.4a}$$

$$\mu_k(\alpha_k) \leq (1 - \alpha_k(1 - \beta))\mu_k. \tag{7.4b}$$

Step 4 (Update iterate): Set $w_{k+1} = w_k(\alpha_k), \mu_{k+1} = \mu_k(\alpha_k)$. Increment k by 1 and go to Step 1.

Algorithm 7.2 Primal-Dual Regularized Algorithm for the NT Direction

Apply Algorithm 7.1 where, in Step 2, (6.13) is solved instead of (6.10).

8. Convergence Preliminaries

We provide a convergence analysis for two particular strategies for updating the regularization parameters, corresponding to the two variants of Friedlander and Orban (2012). Each strategy uses a different subset of the conditions (7.1) to define a neighborhood $\mathcal{N}_k$.

The following result follows directly from (Friedlander and Orban, 2012, Theorem 5.1) and parallels (Friedlander and Orban, 2012, Corollary 5.2). The proof is identical and follows directly from the observation that both $F^{-1}E$ and $\bar{E}$ are symmetric and positive definite.

Lemma 8.1. *Let K be the coefficient matrix of either (6.10) or (6.13). For all $\rho > 0$ and $\delta > 0$,*

$$\|K^{-1}\| \leq 1/\min(\rho, \delta). \tag{8.1}$$

Lemma 8.1 thus guarantees that the coefficient matrices in Algorithms 7.1 and 7.2 have uniformly bounded inverses provided ρ and δ do not approach zero.

By linearity of the primal and dual feasibility residuals, we have from (5.4) that

$$A\mathbf{X}(\alpha) + \delta r(\alpha) - b = (1 - \alpha)(A\mathbf{X} + \delta r - b), \tag{8.2}$$

and

$$\mathbf{C} + \rho\mathbf{S}(\alpha) - A^T y(\alpha) - \mathbf{Z}(\alpha) = (1 - \alpha)(\mathbf{C} + \rho\mathbf{S} - A^T y - \mathbf{Z}). \tag{8.3}$$

Since the Newton step is computed from the current w_k,

$$\delta r(\alpha) = \delta(r + \alpha\Delta r) = (1-\alpha)\delta r + \alpha\delta\Delta y \tag{8.4a}$$

$$\rho S(\alpha) = \rho(S + \alpha\Delta S) = (1-\alpha)\rho S + \alpha\rho\Delta X. \tag{8.4b}$$

If ΔX and ΔZ only satisfy (6.1) then we obtain

$$\begin{aligned}\mu(\alpha) &= (X + \alpha\Delta X) \bullet (Z + \alpha\Delta Z)/n \\ &= (1-\alpha)\mu + \alpha(X \bullet Z + X \bullet \Delta Z + \Delta X \bullet Z)/n + \alpha^2(\Delta X \bullet \Delta Z)/n,\end{aligned} \tag{8.5}$$

and using (6.1) again we obtain

$$X \bullet Z + X \bullet \Delta Z + \Delta X \bullet Z = \operatorname{tr}(XZ + X\,\Delta Z + \Delta X\, Z) = n\sigma\mu. \tag{8.6}$$

Therefore,

$$\mu(\alpha) = (1 - \alpha + \sigma\alpha)\mu + \alpha^2(\Delta X \bullet \Delta Z)/n. \tag{8.7}$$

As it turns out, (8.6) and (8.7) still hold true if (6.1) is transformed by H_P for any nonsingular matrix P.

Lemma 8.2. *Let P be any nonsingular matrix and ΔX and ΔZ satisfy $H_P(XZ + \Delta XZ + X\Delta Z) = \sigma\mu I$. Then $X \bullet Z + X \bullet \Delta Z + \Delta X \bullet Z = n\sigma\mu$ and (8.7) holds. In addition,*

$$H_P(X(\alpha)Z(\alpha)) = (1-\alpha)H_P(XZ) + \alpha\sigma\mu I + \alpha^2 H_P(\Delta X\Delta Z). \tag{8.8}$$

Proof. The proof of the first part is an immediate application of part 2 of Lemma 6.1. For the second part, we have

$$X(\alpha)Z(\alpha) = (X + \alpha\Delta X)(Z + \alpha\Delta Z) \tag{8.9}$$

$$= XZ + \alpha X\Delta Z + \alpha\Delta XZ + \alpha^2\Delta X\Delta Z \tag{8.10}$$

$$= (1-\alpha)XZ + \alpha(XZ + X\Delta Z + \Delta XZ) + \alpha^2\Delta X\Delta Z. \tag{8.11}$$

Linearity of H_P and (6.1) conclude the proof. □

Since Lemma 8.2 applies for any nonsingular P, we have in particular for $P = X^{-\frac{1}{2}}$ and $P = W^{-\frac{1}{2}}$ that (8.6), (8.7), and (8.8) hold true when $(\Delta X, \Delta Z)$ satisfy Step 2 of Algorithm 7.1 or 7.2.

9. Algorithm Based on Fixed Regularization Parameters

Our first method leaves the regularization parameters ρ_k and δ_k fixed throughout the iterations. It forces the iterates to satisfy (7.1a)-(7.1c). Since regularization parameters are fixed we denote them ρ and δ for readability. Algorithm 9.1 formalizes the context.

Algorithm 9.1 Variation with Constant Regularization

Apply Algorithm 7.1 or 7.2 with $\rho_k = \rho_0$ and $\delta_k = \delta_0$ for all k. In Step 3, only conditions (7.1a), (7.1b), and (7.1c) are enforced.

Convergence properties rely on the following technical lemma, which parallels (Friedlander and Orban, 2012, Lemma 5.3). Although most of the proof is identical

to the case of quadratic programming, we state it for completeness because of its central role in the convergence analysis.

Lemma 9.1. *Suppose $(\Delta X, \Delta y, \Delta Z)$ is given by Step 2 of Algorithm 9.1, and that $\{r_k\}$, $\{S_k\}$ and $\{Z_k\}$ are bounded. Then $(\Delta X, \Delta y, \Delta Z)$ is also bounded.*

Proof. We start by showing that the common right-hand side to (6.10) and (6.13) is bounded. Note that the mechanism of Algorithm 9.1 ensures that μ_k is decreasing. From (7.1b) we have

$$\|A\mathbf{X}_k - b\| \leq \|A\mathbf{X}_k + \delta r_k - b\| + \delta\|r_k\| \leq \gamma_P \mu_0 + \delta \sup_k \|r_k\|.$$

Using the centering condition (7.1a) and part (1) of Lemma 6.1, we have

$$\gamma_C \mu_k \leq \lambda_i(X_k Z_k) \leq \bar{\gamma}_C \mu_k, \tag{9.1}$$

and hence

$$\frac{\sigma_k}{\bar{\gamma}_C} - 1 \leq \sigma_k \mu_k \lambda_i(X_k^{-1} Z_k^{-1}) - 1 \leq \frac{\sigma_k}{\gamma_C} - 1, \tag{9.2}$$

which implies

$$\left|\lambda_i(\sigma_k \mu_k X_k^{-1} Z_k^{-1} - I)\right| \leq M, \quad \text{for all} \quad i = 1, \ldots n,$$

where $M := \max(|\frac{\sigma_{\min}}{\bar{\gamma}_C} - 1|, |\frac{\sigma_{\max}}{\gamma_C} - 1|)$. Therefore,

$$\|\sigma_k \mu_k X_k^{-1} Z_k^{-1} - I\|_F \leq \sqrt{n} M. \tag{9.3}$$

Consequently,

$$\|\sigma_k \mu_k X_k^{-1} - Z_k\|_F = \|(\sigma_k \mu_k X_k^{-1} Z_k^{-1} - I) Z_k\|_F \leq \sqrt{n} M \sup_k \|Z_k\|_F. \tag{9.4}$$

Using now (7.1c) and the bound just obtained,

$$\begin{aligned}
\|\mathbf{C} - A^T y_k - \sigma_k \mu_k \mathbf{X}_k^{-1}\| &\leq \|\mathbf{C} - A^T y_k + \rho \mathbf{S}_k - \mathbf{Z}_k\| + \rho\|\mathbf{S}_k\| + \|\sigma_k \mu_k X_k^{-1} - Z_k\|_F \\
&\leq \gamma_D \mu_k + \rho \sup_k \|S_k\|_F + \|\sigma_k \mu_k X_k^{-1} - Z_k\|_F \\
&\leq \gamma_D \mu_0 + \rho \sup_k \|S_k\|_F + \sqrt{n} M \sup_k \|Z_k\|_F.
\end{aligned}$$

We have showed that the common right-hand side to (6.10) and (6.13) is bounded. By Corollary 8.1, the inverse of the coefficient matrix of (6.10) and (6.13) is uniformly bounded and therefore, $(\Delta X, \Delta y)$ is bounded. To see that ΔZ is bounded, note that the first block row of both (6.8) and (6.13) reads

$$\boldsymbol{\Delta Z} = \mathbf{C} - A^T y_k - \mathbf{Z}_k + \rho \boldsymbol{\Delta X} - A^T \Delta y,$$

which has a bounded right-hand side. □

In the next lemma, we show that sufficiently small steplengths satisfy (7.4), which, in the context of Algorithm 9.1, amounts to (7.1a), (7.1b), (7.1c) and (7.4b).

Lemma 9.2. *Suppose $(\Delta X, \Delta y, \Delta Z)$ is computed at Step 2 of Algorithm 9.1, and that $\{r_k\}$, $\{S_k\}$, and $\{Z_k\}$ are bounded. Suppose also that there exists $\varepsilon > 0$ and an index k_0 such that $\mu_k \geq \varepsilon$ for all $k \geq k_0$. Then there exists $\alpha^* \in (0, 1]$ such that for all $\alpha \in (0, \alpha^*]$, the conditions (7.4) are satisfied.*

Proof. Define $G(\alpha) := H_P(X_k(\alpha)Z_k(\alpha)) - \gamma_C\mu_k(\alpha)I$. From (8.7) and (8.8) we have

$$G(\alpha) = (1-\alpha)[H_P(X_kZ_k) - \gamma_C\mu_k I] + \alpha\sigma_k\mu_k(1-\gamma_C)I + \alpha^2\left[H_P(\Delta X\Delta Z) - \gamma_C\frac{\Delta X \bullet \Delta Z}{n}I\right].$$

Lemmas 6.1, 9.1 and (A.2a)–(A.2d) yield

$$\lambda_n(G(\alpha)) \geq (1-\alpha)\lambda_n(H_P(X_kZ_k) - \gamma_C\mu_k I) + \alpha\sigma_k\mu_k(1-\gamma_C) - \alpha^2\pi_1, \tag{9.5}$$

for some $\pi_1 > 0$ that depends only on n. Using (9.5), (7.1a) and our assumption that $\mu_k \geq \varepsilon$, we see that $\lambda_n(G(\alpha)) \geq 0$ will be satisfied if we choose $\alpha > 0$ such that $\alpha \leq \sigma_{\min}(1-\gamma_C)\varepsilon/\pi_1$. Similarly, $X(\alpha)$ and $Z(\alpha)$ will satisfy the right inequality of (7.1a) if $\alpha \leq \sigma_{\min}(\bar{\gamma}_C - 1)\varepsilon/\pi_2$ for some $\pi_2 > 0$ depending only on n.

We now use (7.1b), (8.2) and (8.7) to get

$$\begin{aligned}\gamma_P\mu_k(\alpha) - \|A\mathbf{X}_k(\alpha) + \delta r_k(\alpha) - b\| &\geq \alpha\sigma_k\gamma_P\mu_k + \alpha^2\gamma_P\Delta X \bullet \Delta Z/n \\ &\geq \gamma_P(\alpha\sigma_{\min}\mu_k - \alpha^2\pi_3),\end{aligned}$$

for some $\pi_3 > 0$ depending only on n. Thus (8.3) yields

$$\begin{aligned}\gamma_D\mu_k(\alpha) - \|\mathbf{C} + \rho\mathbf{S}_k(\alpha) - A^Ty_k(\alpha) - \mathbf{Z}_k(\alpha)\| &\geq \alpha\sigma_k\gamma_D\mu_k + \alpha^2\gamma_D\Delta X \bullet \Delta Z/n \\ &\geq \gamma_D(\alpha\sigma_{\min}\varepsilon - \alpha^2\pi_3).\end{aligned}$$

Therefore, it is enough to choose $\alpha \leq \sigma_{\min}\varepsilon/\pi_3$. Using (8.7) again, we have

$$\begin{aligned}(1-\alpha(1-\beta))\mu_k - \mu_k(\alpha) &= \alpha(\beta - \sigma_k)\mu_k + \alpha^2\Delta X \bullet \Delta Z/n \\ &\geq \alpha(\beta - \sigma_{\max})\varepsilon - \alpha^2\pi_3,\end{aligned}$$

and any $\alpha > 0$ such that $\alpha \leq (\beta - \sigma_{\max})\varepsilon/\pi_3$ will satisfy (7.4b).

Finally, $X_k(\alpha)$ and $Z_k(\alpha)$ must be positive definite. Since the eigenvalues of $X_k(\alpha)$ and $Z_k(\alpha)$ are continuous functions of α and $X_k(0) = X_k$ and $Z_k(0) = Z_k$ are positive definite, there exists $\bar{\alpha}$ such that $X_k(\alpha)$ and $Z_k(\alpha)$ are positive definite for all $\alpha \in (0, \bar{\alpha}]$. The proof of the lemma is completed if we let

$$\alpha^* := \min\left\{1, \frac{\sigma_{\min}(1-\gamma_C)\varepsilon}{\pi_1}, \frac{\sigma_{\min}(\bar{\gamma}_C - 1)\varepsilon}{\pi_2}, \frac{\sigma_{\min}\varepsilon}{\pi_3}, \frac{(\beta - \sigma_{\max})\varepsilon}{\pi_3}, \bar{\alpha}\right\},$$

and note that Algorithm 9 in Step 3 chooses the largest possible α. □

Now, we are in a position to establish the convergence properties of Algorithm 9.1. In the following theorem, we show that the duality measure μ_k converges to zero under a boundedness assumption.

Theorem 9.3. *Suppose that Algorithm 9.1 generates the sequence $\{w_k\}$, and that the sequence $\{(r_k, S_k, Z_k)\}$ is bounded. Then $\mu_k \to 0$ as $k \to +\infty$.*

Proof. If $\varepsilon := \liminf \mu_k > 0$, then, using Lemma 9.2, we can find $\alpha^* > 0$ such that Step 3 of Algorithm 9.1 is satisfied for all $\alpha \in (0, \alpha^*]$ and for all k. In particular,

$$0 < \varepsilon \leq \mu_{k+1} \leq (1 - \alpha^*(1-\beta))\mu_k \leq \cdots \leq (1 - \alpha^*(1-\beta))^{k+1}\mu_0.$$

This is a contradiction because the rightmost term above converges to zero. □

The nature of limit points becomes clear in the next result. The proof is a simple adaptation of (Friedlander and Orban, 2012, Theorem 5.5).

Theorem 9.4. *Suppose Algorithm 9.1 with $\varepsilon = 0$ generates the sequence $\{w_k\}$, and that $\{(r_k, S_k, Z_k)\}$ remains bounded. Then if $\hat{r}$ and $\hat{S}$ denote particular limit points of $\{r_k\}$ and $\{S_k\}$ along subsequences indexed by $\mathcal{K} \subset \mathbb{N}$, every limit point of $\{(X_k, Z_k)\}_{\mathcal{K}}$ determines a primal-dual solution of the primal-dual pair*

$$\underset{X}{\text{minimize}} \ (C + \rho\hat{S}) \bullet X \quad \text{subject to} \quad \mathcal{A}X = b - \delta\hat{r}, \ X \succeq 0, \tag{9.6}$$

$$\underset{y,Z}{\text{maximize}} \ (b - \delta\hat{r})^T y \quad \text{subject to} \quad \mathcal{A}^* y + Z = C + \rho\hat{S}, \ Z \succeq 0. \tag{9.7}$$

Theorem 9.4 has the flavor of a backward stability result and states that Algorithm 9.1 always identifies solutions to a nearby problem. In certain situations, this nearby problem coincides with (P)–(D).

In the next theorem, we state some results regarding the limit points of $\{r_k\}$ and $\{S_k\}$ and the feasibility of (P) and (D). The proof is a direct adaptation of (Friedlander and Orban, 2012, Theorem 5.6).

Theorem 9.5. *Suppose Algorithm 9.1 with $\varepsilon = 0$ generates the sequence $\{w_k\}$ and that $\{(r_k, S_k, Z_k)\}$ remains bounded. Then*

1. *If $\{r_k\}_{\mathcal{K}} \to 0$ for some index set $\mathcal{K} \subseteq \mathbb{N}$, every limit point of $\{X_k\}_{\mathcal{K}}$ is feasible for (P).*
2. *If $\{S_k\}_{\mathcal{K}'} \to 0$ for some index set $\mathcal{K}' \subseteq \mathbb{N}$, every limit point of $\{(X_k, Z_k)\}_{\mathcal{K}'}$ determines a feasible point for (D).*
3. *If $\{r_k\}_{\mathcal{K}''} \to 0$ and $\{S_k\}_{\mathcal{K}''} \to 0$ for some index set $\mathcal{K}'' \subseteq \mathbb{N}$, every limit point of $\{X_k, Z_k\}_{\mathcal{K}''}$ determines a primal-dual solution of (P) and (D).*

10. Algorithm Based on Decreasing Regularization Parameters

Our second variant permits decreasing regularization parameters ρ_k and δ_k and defines the neighborhood $\mathcal{N}_k$ via conditions (7.1a), (7.1d), and (7.1e). It is formalized as Algorithm 10.1.

Algorithm 10.1 Variation with Variable Regularization

Apply Algorithm 7.1 or 7.2 with the following specializations. In Step 3, only conditions (7.1a), (7.1d) and (7.1e) are enforced, and ρ_{k+1} and δ_{k+1} are chosen so that

$$\rho_{k+1} \geq \kappa_\rho \rho_k, \tag{10.1a}$$

$$\delta_{k+1} \geq \kappa_\delta \delta_k, \tag{10.1b}$$

for some $0 < \kappa_\rho < 1$ and $0 < \kappa_\delta < 1$.

The properties of Algorithm 10.1 are summarized in the following result, which parallels (Friedlander and Orban, 2012, Theorem 5.8).

Theorem 10.1. *Suppose Algorithm 10.1 with $\varepsilon = 0$ generates the sequence $\{w_k\}$ and that $\{(X_k, Z_k)\}$ remains bounded. Suppose also that there exists $k_0 \in \mathbb{N}$ and $\alpha^* \in (0, 1]$ such that $\alpha_k \geq \alpha^*$ for all $k \geq k_0$. Then the sequence $\{(X_k, r_k, S_k, Z_k)\}$ is bounded and every limit point of $\{(X_k, Z_k)\}$ determines a primal-dual solution of (P)–(D).*

11. Numerical Experiments

We implemented a predictor-corrector variant of Algorithms 9.1 and 10.1 as a modification of SDPT3 (Toh et al., 1999). SDPT3 implements a standard predictor-corrector method for the conic optimization problem

$$\underset{x}{\text{minimize}}\ \langle c, x\rangle \quad \text{subject to } Gx \preccurlyeq_K h,\ Ax = b \tag{11.1}$$

in the variable x where K is a closed pointed convex cone with nonempty interior and $\langle\cdot,\cdot\rangle$ is an appropriate inner product. The cone K may be a cross product of semidefinite cones, second-order cones and nonnegative orthants. A problem formulated as (11.1) involving only the positive semidefinite cone may be rewritten as (P).

SDPT3 uses the normal equations formulation to compute a Newton direction. By eliminating $\mathbf{\Delta X}$ form (6.10) and (6.13) we obtain the following system which is the regularized version of the system used in the SDPT3 solver

$$\left[A(Q_k + \rho I)^{-1}A^T + \delta I\right]\Delta y = r_p + A(Q_k + \rho I)^{-1}(\mathbf{C} - A^T y_k - \sigma_k\mu_k\mathbf{X}_k^{-1}), \tag{11.2}$$

where $Q_k = F_k^{-1}E_k$ or $Q_k = \bar{E}_k$ depending on whether (6.10) or (6.13) is used. In both cases, it is obvious that the coefficient matrix of (11.2) is positive definite and its smallest eigenvalue is larger than δ. Using the proof of Theorem 9.1 and the fact that the smallest eigenvalue of $(Q_k + \rho I)$ is larger than ρ, it is clear that the right-hand side of (11.2) remains uniformly bounded. Therefore, Δy remains uniformly bounded. One can easily show that $\mathbf{\Delta X}$ is also bounded by using (6.10) and (6.13). The boundedness of $\mathbf{\Delta Z}$ follows from the proof of Theorem 9.1. Therefore, our convergence analysis also applies if Algorithms 7.1, 7.2, 9.1, and 10.1 use (11.2) in place of (6.10) or (6.13).

The Newton step computation in SDPT3 is based on the normal equations (1.2) with the difference that when (11.1) contains free real variables, those equations take the form

$$\begin{bmatrix} 0 & A_u^T \\ A_u & M \end{bmatrix}\begin{bmatrix} \Delta x_u \\ \Delta y \end{bmatrix} = \begin{bmatrix} q \\ r \end{bmatrix} \tag{11.3}$$

for some right-hand sides q and r, where A_u is the constraint matrix associated to free real variables and M is the Schur complement of all other constraint blocks. Since the above coefficient matrix is indefinite, SDPT3 removes all free variables by decomposing them into positive and negative part. As a consequence, SDPT3 solves a larger problem in which A_u is vacuous and (11.3) reduces to a symmetric positive definite system whose coefficient matrix we denote $\tilde{M}$. A specific concern is that $\tilde{M}$ is typically dense and may be very ill conditioned. The same concern already applied to M.

When regularization is in effect, we have the opportunity to solve a larger but considerably sparser system by not eliminating free variables. We thus disabled

the variable splitting. The equivalent of (1.3) is then intricate to form because it involves propagating the effect of primal regularization inside M. Since this turned out to require substantial modifications to SDPT3, we implemented a simple variant of Algorithms 9.1 and 10.1 in which only free real variables are assigned a primal regularization term, much in the vein of Anjos and Burer (2007). Thus we use the following simpler regularization of (11.3)

$$\begin{bmatrix} -\rho I & A_u^T \\ A_u & M + \delta I \end{bmatrix} \begin{bmatrix} \Delta x_u \\ \Delta y \end{bmatrix} = \begin{bmatrix} q \\ r \end{bmatrix}. \tag{11.4}$$

The coefficient matrix of the latter system is symmetric and quasi-definite and therefore is strongly factorizable (Vanderbei, 1995). When the problem has no free variables, this means that only dual regularization is in effect. Since dual regularization guards against rank-deficient constraint matrices, we also disabled the detection of dependent constraints. SDPT3 automatically selects the appropriate type of factorization for each given problem based on structure and makes provision for sparse symmetric indefinite factorization, i.e., a factorization of the form LBL^T where L is unit lower triangular and B is block diagonal—see, e.g., (Duff, 2004). Whenever both $\rho > 0$, $\delta > 0$ and the problem has free variables, we enforce usage of the latter but set the pivot threshold to zero so as to compute a sparse Cholesky-type factorization, i.e., a factorization of the form LDL^T where D is diagonal. Computing the LDL^T factorization of a quasi-definite matrix instead of its LBL^T factorization typically results in lower factorization time and sparser factors because pivoting is only necessary to improve sparsity (Gill et al., 1996). However, it is important to realize that in our implementation, we perform the LDL^T factorization of (11.4) while SDPT3 computes the Cholesky factorization of $\tilde{M}$, which can be considerably smaller. Because the density of each system is almost identical across our test problems, it is typically the case that our variant is slightly slower. If the problem has no free variables, SDPT3 solves (1.3) and we let it choose between a dense or sparse Cholesky factorization.

By default, SDPT3 adds a regularization term to $\tilde{M}$ of the form $\lambda_1 \operatorname{diag}(\tilde{M}) + \lambda_2 AA^T$ for some positive values of λ_1 and λ_2, presumably so as to guard against rank-deficient constraint matrices. Whenever $\delta > 0$, we disabled the addition of this term. In summary, our dual regularization term is simpler and cheaper to compute, our primal regularization terms allows to do away with free variable splitting and both are supported by a global convergence theory.

We initialize both regularization parameters to 10 and use the simple update

$$\rho_{k+1} = \min(\tfrac{1}{5}\rho_k, 10^{-8}), \qquad \delta_{k+1} = \min(\tfrac{1}{5}\delta_k, 10^{-8})$$

at each iteration. SDPT3 declares convergence when $\phi(X, y, Z) \leq \epsilon$ where

$$\phi(X, y, Z) := \max\left(\frac{X \bullet Z/n}{1 + |C \bullet X| + |b^T y|}, \frac{\|\mathcal{A}X - b\|}{1 + \|b\|}, \frac{\|\mathcal{A}^* y + Z - C\|}{1 + \|C\|} \right),$$

and where ϵ was set to 10^{-6}. A maximum of 100 iterations were allowed for each method. We found that various tests for slow convergence in SDPT3 were not well calibrated and interfered with convergence in a number of cases. Those tests were deactivated. When regularization is in effect, this resulted in some previously failing problems being solved successfully.

Our results are presented in the form of performance profiles (Dolan and Moré, 2002). Figure 1 shows performance profiles in terms of the number of iterations

and CPU time on a collection of 30 problems with free variables available from plato.asu.edu/ftp/sdp_free. The default version of SDPT3 solves 25 of those problems while our version solves all of them. The iteration profile also shows that the regularized method requires slightly less iterations on average. The CPU profile shows however that each iteration appears to be more time consuming. The failures of the default version of SDPT3 are shown in Table 1. Negative values of the number of iterations and cpu time are used to indicate failures. On all five problems, the residuals are fairly small and the algorithm has reached near optimality. An exit code of -3 means that either X or Z (or both) was found to not be positive definite while -6 means that the maximum number of iterations was reached. While the latter is only a minor kind of failure, the former appear to be much more serious.

Figure 2 shows corresponding profiles for the 92 problems of the SDPLIB collection, available from euler.nmt.edu/~brian/sdplib/sdplib.html. The default version of SDPT3 fails on 10 problem while the regularized variant fails on 9 on them. In all cases, all residuals are fairly small. Tables 2 and 3 detail the failures. An exit code of -2 means that lack of progress due to short steps was detected in either the predictor or the corrector step. Almost all failures are minor.

In both figures, CPU time must be taken cautiously given that Matlab is an interpreted language with substantial overhead, but the profile illustrates how fast a Cholesky factorization of $\tilde{M}$ may be compared to a LDL^T factorization of (11.4). As stated earlier, this apparent lack of advantage of using the LDL^T factorization is due to $\tilde{M}$ not being very dense across our test set.

We repeated the same tests after deactivating the default regularization in SDPT3 and obtained virtually identical results. This suggests that the test problems in these two collections do not suffer from serious degeneracy and confirms that regularization does not have a serious impact on performance when applied to non-degenerate problems.

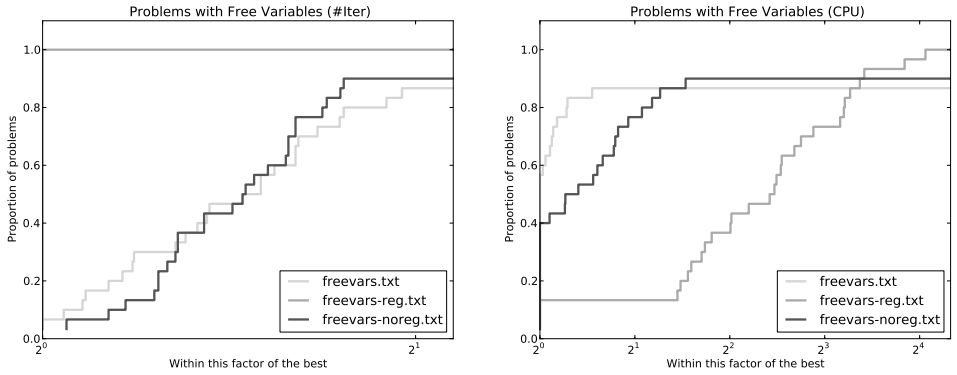

FIGURE 1. Results on a collection of 30 problems with free variables.

TABLE 1. Failures of the default version of SDPT3 on problems with free variables.

Name	Iter	pFeas	dFeas	gap	cpu	exit
ems_17_0.5_2_2	−54	1.6e−06	3.8e−08	2.1e−06	−9.3e+02	−3
ems_17_0.5_4_2	−86	9.1e−06	5.1e−09	7.1e−07	−1.9e+03	−3
ems_17_0.5_5_2	−100	4.6e−06	2.9e−08	3.0e−06	−2.6e+03	−6
ems_17_0.5_6_2	−56	1.9e−06	1.1e−08	8.3e−07	−1.1e+03	−3
ems_17_0.5_7_2	−58	2.0e−06	1.4e−09	1.9e−07	−1.1e+03	−3

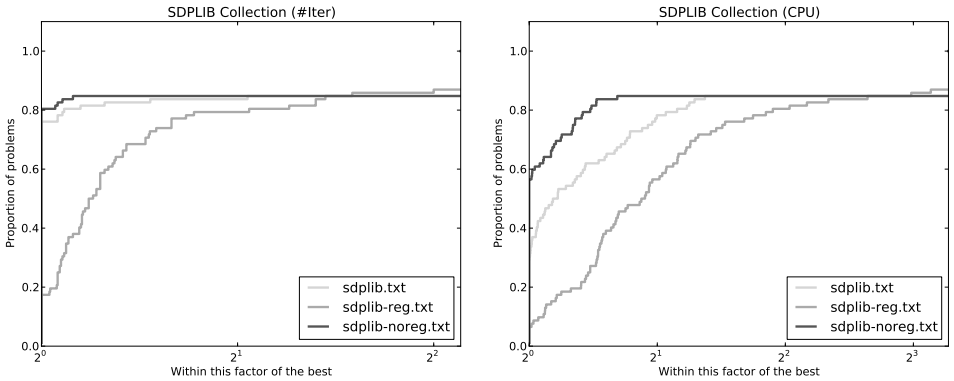

FIGURE 2. Results on the SDPLIB collection (92 problems).

TABLE 2. Failures of the default version of SDPT3 on problems from the SDPLIB collection.

Name	Iter	pFeas	dFeas	gap	cpu	exit
control10	−100	5.3e−07	4.8e−07	9.5e−05	−4.3e+02	−6
hinf13	−100	4.8e−06	1.4e−07	9.7e−03	−3.9e+00	−6
hinf14	−100	3.4e−07	2.1e−07	4.0e−04	−4.2e+00	−6
hinf15	−100	5.0e−06	6.3e−07	5.0e−02	−5.4e+00	−6
hinf2	−100	3.2e−06	1.5e−11	6.0e−08	−1.7e+00	−6
hinf3	−100	9.7e−06	2.5e−12	1.1e−07	−1.7e+00	−6
hinf5	−24	2.1e−04	7.2e−08	2.6e−02	−3.3e−01	−2
hinf6	−100	5.6e−06	3.6e−12	1.1e−06	−1.7e+00	−6
hinf7	−100	8.2e−06	9.4e−14	4.1e−08	−1.4e+00	−6
hinf8	−39	1.9e−05	1.3e−11	1.8e−06	−6.0e−01	−2

TABLE 3. Failures of the regularized version of SDPT3 on problems from the SDPLIB collection.

Name	Iter	pFeas	dFeas	gap	cpu	exit
control10	−100	4.4e−08	1.5e−09	5.1e−04	−4.48e+02	−6
hinf10	−100	5.1e−07	6.7e−08	3.4e−04	−2.15e+00	−6
hinf12	−100	9.9e−07	2.7e−07	3.1e−04	−2.17e+00	−6
hinf13	−31	2.3e−05	2.9e−05	1.1e+00	−8.67e−01	−2
hinf15	−91	3.5e−04	4.1e−08	6.6e−03	−4.47e+00	−2
hinf5	−100	1.2e−05	8.4e−12	2.8e−06	−1.46e+00	−6
hinf6	−100	8.6e−06	3.9e−12	8.2e−07	−1.89e+00	−6
hinf7	−61	3.5e−06	1.4e−12	1.2e−05	−9.35e−01	−2
hinf8	−100	1.5e−05	3.1e−11	4.2e−06	−1.97e+00	−6

We additionally report on experiments with a set of 42 problems with free variables from polynomial optimization (Waki et al., 2008), this time with a maximum of 500 iterations. Table 4 documents the failures of the variant of SDPT3 equipped with the regularization described in the present paper, while Table 5 documents the failures of SDPT3 equipped with its default regularization. Exit code −1 indicates that the final relative gap is less than infeasibility, −5 indicates lack of progress in the relative gap, and 3 indicates that the solution appears to diverge. Performance profiles in terms of number of iterations and CPU time appear in Figure 3 and show that the variant equipped with our regularization has benefits in robustness. We note that our regularization also often yields smaller final optimality residuals, as can be seen in the complete results in Appendix B.

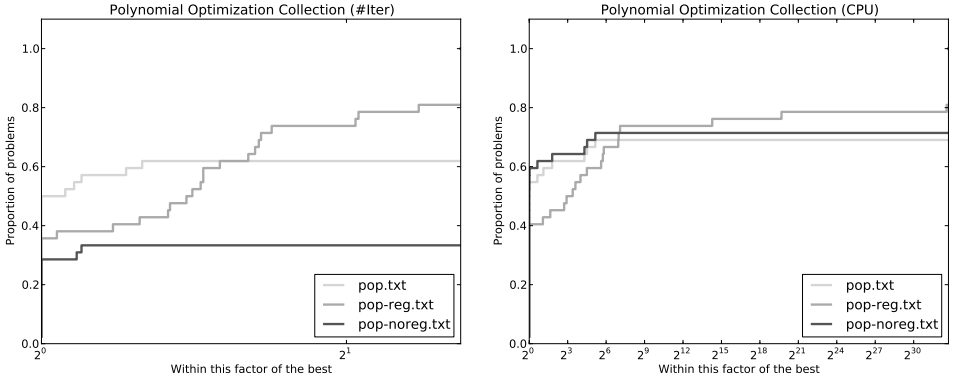

FIGURE 3. Results on the polynomial optimization collection (42 problems).

Table 4: Failures of the regularized version of SDPT3 on 42 problems with free variables from polynomial optimization.

Name	Iter	pFeas	dFeas	gap	cpu	exit
Bex5_3_2	−182	1.96e−08	9.95e−10	3.57e−06	−3.18e+01	−2
Bex9_1_2	−112	1.50e−07	1.09e−07	2.68e−06	−1.42e+01	−3

Bex9_1_5	−426	3.63e−08	1.80e−10	1.29e−10	−3.07e+01	−3
Bex9_2_2	−325	2.15e−07	9.35e−09	6.19e−08	−1.18e+01	−2
Bex9_2_6	−84	1.20e−08	6.07e−12	1.30e−09	−1.98e+01	−3
Bhaverly	−500	3.37e−07	4.98e−09	1.21e−08	−1.35e+02	−6
alkylation	−80	3.55e−07	3.46e−13	1.28e−10	−2.81e+01	−3
meanvarx	−119	3.99e+15	4.42e−11	1.10e−10	−6.11e+01	3
st_glmp_fp2	−70	2.97e−05	1.21e−13	8.37e−10	−4.76e+00	−3

Table 5: Failures of the default version of SDPT3 on 42 problems with free variables from polynomial optimization.

Name	Iter	pFeas	dFeas	gap	cpu	exit
Babel	−42	1.60e−04	6.65e−11	1.55e−07	−3.46e+00	−5
Balkyl	−58	3.94e−04	5.81e−10	7.71e−07	−1.10e+01	−1
Bex2_1_8	−50	1.06e−03	5.73e−07	2.72e−03	−1.14e+01	−5
Bex5_3_2	−55	1.83e−04	2.37e−08	6.38e−05	−4.92e+00	−5
Bex9_1_2	−66	4.73e−04	2.33e−06	2.15e−03	−6.65e+00	−5
Bex9_2_1	−36	3.24e−04	3.71e−08	2.20e−05	−2.21e+00	−1
Bex9_2_2	−37	4.88e−05	3.21e−07	7.36e−05	−9.62e−01	−5
Bex9_2_6	−31	1.95e−04	9.92e−08	1.88e−04	−3.53e+00	−5
Bex9_2_7	−38	4.01e−04	1.22e−07	6.25e−05	−2.46e+00	−1
Bhaverly	−50	1.12e−04	4.54e−07	4.82e−04	−2.20e+00	−5
Bst_e07	−44	2.10e−07	5.09e−13	2.84e−10	−1.38e+00	−1
Bst_e33	−41	4.82e−08	2.15e−14	9.93e−12	−1.18e+00	−3
Bst_e42	−82	6.12e−07	3.05e−08	1.74e−05	−2.01e+00	−5
alan	−35	1.33e−07	2.60e−09	1.23e−07	−1.48e+00	−5
himmel11	−66	2.89e−07	3.13e−11	5.49e−08	−3.04e+00	−5
meanvarx	−66	4.64e−03	5.92e−06	4.54e−03	−5.97e+00	−5
st_glmp_fp2	−39	8.70e−08	1.04e−11	2.58e−09	−2.12e+00	−3

12. Discussion

Let V denote a Hilbert space endowed with a self-dual norm $\|\cdot\|_V$ induced by an inner product $\langle\cdot,\cdot\rangle_V$. For clarity, we denote $\|\cdot\|_{V^*}$ the dual norm of $\|\cdot\|_V$ when applied to elements of the dual space V^*. In this section, we denote elements of the dual space in boldface. By way of Riesz's representation theorem, to each $x \in V$ corresponds a unique $\mathbf{x} \in V^*$ such that $\|x\|_V = \|\mathbf{x}\|_{V^*}$ and conversely, to each $\mathbf{x} \in V^*$ corresponds a unique $x \in V$. Let then $\langle\cdot,\cdot\rangle_{V^*,V}$ denote the duality pairing between V and V^* defined by $\langle\mathbf{x}, y\rangle_{V^*,V} := \langle x, y\rangle_V$.

The desirable duality properties of the primal-dual regularization generalize to the quadratic cone problem in slack variable form

$$\underset{x\in\mathbb{R}^n, s\in V}{\text{minimize}}\ q(x), \quad \text{subject to } \mathcal{G}x + s = h,\ Ax = b,\ s \succcurlyeq_K 0, \tag{12.1}$$

where $q(x) := c^T x + \frac{1}{2}x^T H x$, $H \in \mathbb{S}^n$, $A \in \mathbb{R}^{m\times n}$, $\mathcal{G}$ is an operator from $\mathbb{R}^n$ into V, $h \in V$, and $K \subset V$ is a self-dual convex cone. The primal-dual regularization of

(12.1) reads

$$\begin{aligned} \underset{x,r,s,\mathbf{t}}{\text{minimize}} \quad & q(x) + \tfrac{1}{2}\rho\|x - x_k\|^2 + \tfrac{1}{2}\rho\|s - s_k\|_V^2 + \tfrac{1}{2}\delta\|r + y_k\|^2 + \tfrac{1}{2}\delta\|\mathbf{t} + \mathbf{w}_k\|_{V^*}^2 \\ \text{subject to} \quad & \mathcal{G}x + s + \delta t = h, \ Ax + \delta r = b, \ s \succcurlyeq_K 0. \end{aligned}$$

As a convex problem, the dual of this regularized problem is

(12.2)

$$\begin{aligned} \underset{u,v,\mathbf{w},x,y,\mathbf{z}}{\text{maximize}} \quad & b^T y + \langle \mathbf{w}, h \rangle_{V^*,V} - \tfrac{1}{2}x^T H x \\ & - \tfrac{1}{2}\rho\|u + x_k\|^2 - \tfrac{1}{2}\rho\|v + s_k\|_V^2 - \tfrac{1}{2}\delta\|y - y_k\|^2 - \tfrac{1}{2}\delta\|\mathbf{w} - \mathbf{w}_k\|_{V^*}^2 \\ \text{subject to} \quad & Hx - \mathcal{G}^*\mathbf{w} - A^T y + \rho u = -c, \ \mathbf{w} + \mathbf{z} - \rho\mathbf{v} = 0, \ \mathbf{z} \succcurlyeq_{K^*} 0, \end{aligned}$$

where $\mathcal{G}^*$ is the adjoint of $\mathcal{G}$. Upon setting $\rho = \delta = 0$, (12.2) reduces to the dual of (12.1).

Once again, if $\rho > 0$ and $\delta > 0$, the above regularized primal-dual pair is always strictly feasible and therefore, strong duality always holds for it. This provides for a convenient framework in which solutions to a possibly degenerate primal-dual pair may be approached by approximate solutions to a sequence of non-degenerate primal-dual pairs.

The similarity between interior-point methods for (12.1) and for (P) suggests that our entire computational framework generalizes as well.

In practice, the main computational advantage of using (6.9) instead of (1.3) is that a standard sparse LDL^T factorization may be employed, even though the coefficient matrix is indefinite, instead of a Choleksy factorization with special provision for dense columns. We expect that this could yield savings in time and memory consumption on problems for which (1.3) is relatively dense. Recent results in convex quadratic programming (Greif et al., 2012) indicate however that (6.9) is expected to have a condition number that increases without bound, which is not the case for (11.4). Though using (6.9) involves fundamental changes in the implementation of SDPT3, it has the advantage of a simpler and more natural implementation. It is possible to perform the requisite factorization in Matlab by calling `ldl()` with a pivot threshold set to zero. In addition, dual regularization allows us to dispense with the assumption that the operator $\mathcal{A}$ of (P) is onto.

13. Future Research

Our primal-dual regularization for SDPs yields a convergent algorithm for both the dual HKM and NT directions. There are about twenty known directions in SDP (Todd, 1999) and we believe it is possible to extend the convergence analysis to some of them.

Prior to regularization, the condition number of (1.1) may be orders of magnitude smaller than that of (1.2) (Greif et al., 2012). This fact may not be well known. What is well known is that the normal equations are typically dense and ill conditioned. It is our understanding that most implementations of interior-point methods rely on them nevertheless because the numerical linear algebra of this system is well understood—the Cholesky factorization or the conjugate gradient method. By contrast, direct and iterative methods for symmetric indefinite systems, i.e., including regularization terms, still appear far less widespread in cone optimization. Symmetric quasi-definite systems provide an advantageous combination—the ability to use the well-understood Cholesky factorization with a lower condition number and sparse factors. Arioli

and Orban (2012) establish that it is also perfectly valid to apply the conjugate gradient method on (6.9) even though, again, the coefficient matrix is indefinite. Other well-known Krylov-type methods are applicable as well and they open the door to novel matrix-free interior-point methods.

Recently, Greif et al. (2012) proposed an even more interesting alternative in the framework of quadratic programming, where X and Z are the diagonal matrices $\text{diag}(x)$ and $\text{diag}(z)$. Instead of (6.9), they propose to solve the block 3×3 system

$$\begin{bmatrix} -\rho I & A^T & Z_k^{\frac{1}{2}} \\ A & \delta I & \\ Z_k^{\frac{1}{2}} & & -X_k \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ -Z_k^{-\frac{1}{2}} \Delta z \end{bmatrix} = \begin{bmatrix} c - A^T y_k - z_k \\ b - A x_k \\ Z_k^{\frac{1}{2}} (\sigma_k \mu_k e - X_k z_k) \end{bmatrix}.$$

The favorable properties of this last coefficient matrix is that it remains uniformly bounded and uniformly nonsingular throughout the iterations and in the limit provided strict complementarity holds and ρ_k and δ_k remain uniformly bounded away from zero. Whether an equivalent system with similar favorable properties exists for semidefinite programming is the subject of current research.

Appendix A. Properties of the Kronecker Product

This section lists the important properties of the Kronecker product used in this paper. Although some of the proofs are easy to derive, we refer to (Horn and Johnson, 1991) for a complete treatment. We use $\lambda(A)$ to denote the spectrum of a matrix A.

(A.1a) $A \otimes B = [a_{ij} B]_{ij}$,

(A.1b) $\mathbf{vec}(AXB) = (B^T \otimes A)\mathbf{X}$,

(A.1c) $(A \otimes B)^T = A^T \otimes B^T$,

(A.1d) $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$,

(A.1e) $(A \otimes B)(C \otimes D) = (AC \otimes BD)$,

(A.1f) If $\lambda(A) = \mu_i$ and $\lambda(B) = \nu_j$ then $\lambda(A \otimes B) = \mu_i \nu_j$,

(A.1g) $\lambda(AB) = \lambda(BA)$, where $A, B \in \mathbb{R}^{n \times n}$.

If A is a symmetric $n \times n$ real matrix, we order eigenvalues of A as $\lambda_1(A) \geq \lambda_2(A) \geq \ldots \geq \lambda_n(A)$. The following lemma gives useful relations between the smallest and the largest eigenvalues of the sum of two matrices.

Lemma A.1. *If A, B and C are $n \times n$ real symmetric matrices then*

(A.2a) $\lambda_n(A + B) \geq \lambda_n(A) + \lambda_n(B)$,

(A.2b) $\lambda_1(A + B) \leq \lambda_1(A) + \lambda_1(B)$,

(A.2c) $\|A - C\|_F \geq \lambda_n(A) - \lambda_n(C)$,

(A.2d) $|\lambda_n(A)| \leq \dfrac{1}{\sqrt{n}} \|A\|_F$.

Proof. Using (Golub and Loan, 1996, Theorem 8.1.5), we have

$$\lambda_i(A) + \lambda_n(B) \leq \lambda_i(A+B) \leq \lambda_i(A) + \lambda_1(B) \quad i = 1, \ldots, n.$$

In particular, for $i = n$ and $i = 1$ we obtain (A.2a) and (A.2b). To prove (A.2d), it is enough to let $A+B = C$, and use (A.2a) and the inequality $\lambda_n(B) \geq -\|B\|_F$. □

Appendix B. Complete Results

This section gives the complete results on the polynomial optimization test set. A zero exit code indicates a success. Nonzero exit codes are as in §11.

Table 6: Results of the regularized version of SDPT3 on 42 problems with free variables from polynomial optimization.

Name	Iter	pFeas	dFeas	gap	cpu	exit
Babel	41	3.05e−09	1.69e−12	1.86e−10	2.95e+00	0
Balkyl	44	3.43e−09	1.02e−10	9.20e−09	7.71e+00	0
Bex2_1_8	46	1.59e−10	8.53e−12	7.56e−09	9.00e+00	0
Bex5_2_2_case1	65	2.24e−09	3.02e−10	1.12e−08	1.05e+01	0
Bex5_2_2_case2	63	6.64e−10	6.85e−11	6.25e−09	1.02e+01	0
Bex5_2_2_case3	57	5.07e−11	8.88e−11	1.32e−08	9.38e+00	0
Bex5_2_5	20	4.62e−09	6.54e−10	1.63e−15	1.95e+00	0
Bex5_3_2	−182	1.96e−08	9.95e−10	3.57e−06	−3.18e+01	−2
Bex9_1_1	75	4.54e−09	4.03e−11	4.62e−09	4.89e+00	0
Bex9_1_2	−112	1.50e−07	1.09e−07	2.68e−06	−1.42e+01	−3
Bex9_1_4	59	1.85e−10	1.54e−10	1.12e−08	2.06e+00	0
Bex9_1_5	−426	3.63e−08	1.80e−10	1.29e−10	−3.07e+01	−3
Bex9_1_8	35	2.11e−09	1.96e−09	2.10e−09	1.36e+00	0
Bex9_2_1	38	6.47e−10	2.99e−10	3.99e−09	2.46e+00	0
Bex9_2_2	−325	2.15e−07	9.35e−09	6.19e−08	−1.18e+01	−2
Bex9_2_3	52	5.45e−10	1.05e−09	7.37e−09	4.97e+00	0
Bex9_2_4	50	4.33e−11	1.38e−11	5.26e−10	3.48e+00	0
Bex9_2_5	33	5.67e−09	1.96e−09	3.27e−09	9.73e−01	0
Bex9_2_6	−84	1.20e−08	6.07e−12	1.30e−09	−1.98e+01	−3
Bex9_2_7	34	1.39e−09	4.80e−09	7.09e−09	2.58e+00	0
Bex9_2_8	24	4.83e−10	1.10e−09	1.95e−11	6.18e−01	0
Bhaverly	−500	3.37e−07	4.98e−09	1.21e−08	−1.35e+02	−6
Bst_e05	56	4.42e−11	5.28e−10	2.65e−09	1.95e+00	0
Bst_e07	55	6.26e−10	3.61e−11	2.52e−09	2.49e+00	0
Bst_e33	35	3.11e−10	1.35e−10	1.13e−09	1.33e+00	0
Bst_e42	78	7.47e−09	2.03e−11	3.80e−07	2.12e+00	0
alan	41	1.85e−10	1.61e−10	1.20e−09	3.18e+00	0
alkylation	−80	3.55e−07	3.46e−13	1.28e−10	−2.81e+01	−3
himmel11	51	3.30e−10	7.53e−10	8.75e−09	4.26e+00	0
meanvarx	−119	3.99e+15	4.42e−11	1.10e−10	−6.11e+01	3
mhw4d	49	1.63e−09	5.65e−12	5.69e−10	1.70e+00	0
qp1	65	2.09e−11	1.86e−12	7.13e−09	4.65e+01	0
qp2	65	1.83e−11	1.84e−12	7.13e−09	4.58e+01	0
st_e10	33	2.10e−09	4.53e−09	2.98e−11	6.84e−01	0
st_glmp_fp1	27	3.28e−09	7.84e−10	7.26e−08	5.93e−01	0

t_glmp_fp2	−70	2.97e−05	1.21e−13	8.37e−10	−4.76e+00	−3
st_glmp_fp3	24	4.78e−09	7.85e−09	1.89e−08	5.25e−01	0
st_glmp_kk90	37	1.51e−10	1.36e−09	1.43e−09	1.18e+00	0
st_glmp_kk92	28	1.64e−10	2.41e−10	4.49e−08	6.06e−01	0
st_glmp_kky	29	1.54e−10	5.34e−10	3.14e−08	7.23e−01	0
st_glmp_ss1	31	7.18e−09	1.44e−11	3.25e−09	6.88e−01	0
st_glmp_ss2	21	5.92e−10	2.39e−09	1.27e−10	4.55e−01	0

Table 7: Results of the default version of SDPT3 on 42 problems with free variables from polynomial optimization.

Name	Iter	pFeas	dFeas	gap	cpu	exit
Babel	−42	1.60e−04	6.65e−11	1.55e−07	−3.46e+00	−5
Balkyl	−58	3.94e−04	5.81e−10	7.71e−07	−1.10e+01	−1
Bex2_1_8	−50	1.06e−03	5.73e−07	2.72e−03	−1.14e+01	−5
Bex5_2_2_case1	40	2.08e−05	8.95e−08	7.10e−05	2.15e+00	0
Bex5_2_2_case2	42	4.79e−05	4.01e−08	4.34e−05	2.28e+00	0
Bex5_2_2_case3	41	7.17e−05	7.99e−08	6.17e−05	2.20e+00	0
Bex5_2_5	17	9.23e−13	3.74e−11	2.91e−09	1.04e+00	0
Bex5_3_2	−55	1.83e−04	2.37e−08	6.38e−05	−4.92e+00	−5
Bex9_1_1	56	2.22e−05	3.81e−09	1.95e−06	2.04e+00	0
Bex9_1_2	−66	4.73e−04	2.33e−06	2.15e−03	−6.65e+00	−5
Bex9_1_4	36	3.23e−06	1.31e−07	3.04e−05	1.06e+00	0
Bex9_1_5	41	4.72e−06	6.30e−08	1.10e−05	1.40e+00	0
Bex9_1_8	44	1.26e−06	1.42e−09	1.51e−07	1.47e+00	0
Bex9_2_1	−36	3.24e−04	3.71e−08	2.20e−05	−2.21e+00	−1
Bex9_2_2	−37	4.88e−05	3.21e−07	7.36e−05	−9.62e−01	−5
Bex9_2_3	56	8.67e−06	1.40e−08	1.45e−05	2.07e+00	0
Bex9_2_4	40	4.21e−05	7.82e−10	2.05e−07	1.54e+00	0
Bex9_2_5	40	8.61e−08	4.79e−10	2.48e−08	1.01e+00	0
Bex9_2_6	−31	1.95e−04	9.92e−08	1.88e−04	−3.53e+00	−5
Bex9_2_7	−38	4.01e−04	1.22e−07	6.25e−05	−2.46e+00	−1
Bex9_2_8	18	3.34e−13	2.72e−10	3.90e−09	4.24e−01	0
Bhaverly	−50	1.12e−04	4.54e−07	4.82e−04	−2.20e+00	−5
Bst_e05	39	5.98e−08	2.92e−09	4.35e−07	1.09e+00	0
Bst_e07	−44	2.10e−07	5.09e−13	2.84e−10	−1.38e+00	−1
Bst_e33	−41	4.82e−08	2.15e−14	9.93e−12	−1.18e+00	−3
Bst_e42	−82	6.12e−07	3.05e−08	1.74e−05	−2.01e+00	−5
alan	−35	1.33e−07	2.60e−09	1.23e−07	−1.48e+00	−5
alkylation	45	2.49e−05	1.51e−07	8.08e−05	6.22e+00	0
himmel11	−66	2.89e−07	3.13e−11	5.49e−08	−3.04e+00	−5
meanvarx	−66	4.64e−03	5.92e−06	4.54e−03	−5.97e+00	−5
mhw4d	24	9.84e−09	1.66e−09	1.14e−08	8.48e−01	0
qp1	45	1.05e−10	4.51e−11	1.28e−09	9.66e+00	0
qp2	45	1.04e−10	4.85e−11	1.28e−09	9.35e+00	0
st_e10	14	1.44e−12	3.22e−10	2.16e−09	3.05e−01	0
st_glmp_fp1	16	4.62e−10	4.13e−09	1.10e−07	3.54e−01	0

st_glmp_fp2	−39	8.70e−08	1.04e−11	2.58e−09	−2.12e+00	−3
st_glmp_fp3	15	1.20e−09	7.62e−09	1.76e−07	3.32e−01	0
st_glmp_kk90	19	2.35e−12	4.28e−10	1.12e−08	5.96e−01	0
st_glmp_kk92	17	4.98e−10	2.87e−09	7.27e−08	3.73e−01	0
st_glmp_kky	28	2.16e−11	5.85e−10	1.73e−08	6.43e−01	0
st_glmp_ss1	22	4.45e−09	2.27e−09	6.96e−08	5.03e−01	0
st_glmp_ss2	23	3.17e−11	5.04e−10	1.39e−08	5.02e−01	0

References

F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5(1):13–51, 1995.

F. Alizadeh, J.P.A. Haeberly, and M.L. Overton. Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results. *SIAM Journal on Optimization*, 8(3):746–768, 1998.

M. F. Anjos and S. Burer. On handling free variables in interior-point methods for conic linear optimization. *SIAM Journal on Optimization*, 18(4):1310–1325, 2007.

M. F. Anjos and J.-B. Lasserre. *Handbook on Semidefinite, Conic and Polynomial Optimization*. Springer, 2012.

M. Arioli and D. Orban. Iterative methods for symmetric quasi-definite linear systems. Cahier du GERAD G-2012-xx, GERAD, Montréal, Québec, Canada, 2012. In preparation.

E. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91:201–213, 2002. URL http://www.mcs.anl.gov/~more/cops/.

I. S. Duff. MA57—a code for the solution of sparse symmetric definite and indefinite systems. *Transactions of the ACM on Mathematical Software*, 30(2):118–144, 2004.

M. P. Friedlander and D. Orban. A primal-dual regularized interior-point method for convex quadratic programs. *Mathematical Programming Computation*, 4(1): 71–107, 2012. DOI: 10.1007/s12532-012-0035-2.

P. E. Gill, M. A. Saunders, and J. R. Shinnerl. On the stability of Cholesky factorization for symmetric quasidefinite systems. *SIAM Journal on Matrix Analysis and Applications*, 17(1):35–46, 1996.

G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins University Press, 3rd edition, 1996.

C. Greif, E. Moulding, and D. Orban. Bounds on the eigenvalues of block matrices arising from interior-point methods. Cahier du GERAD G-2012-42, GERAD, Montréal, Québec, Canada, 2012.

C. Helmberg. *Semidefinite programming for combinatorial optimization*. Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2000.

C. Helmberg, F. Rendl, R.J. Vanderbei, and H. Wolkowicz. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, 6(2):342–361, 1996.

R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Presss, London, 1991.

M. Kojima, S. Shindoh, and S. Hara. Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices. *SIAM Journal on Optimization*, 7:86, 1997.

D. G. Luenberger. *Optimization by vector space methods*. Wiley-Interscience, 1997.

C. Mészáros. On free variables in interior point methods. *Optimization Methods and Software*, 9(1-3):121–139, 1998.

R. D. C. Monteiro. Primal-dual path-following algorithms for semidefinite programming. *SIAM Journal on Optimization*, 7(3):663–678, 1997.

R. D. C. Monteiro and Y. Zhang. A unified analysis for a class of long-step primal-dual path-following interior-point algorithms for semidefinite programming. *Mathematical Programming*, 81(3):281–299, 1998. DOI: 10.1007/BF01580085.

C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12(4):617–629, 1975.

R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976a.

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

M. J. Todd. A study of search directions in primal-dual interior-point methods for semidefinite programming. *Optimization methods and software*, 11(1-4):1–46, 1999.

M. J. Todd, K. C. Toh, and R. H. Tütüncü. On the Nesterov-Todd direction in semidefinite programming. *SIAM Journal on Optimization*, 8(3):769–796, 1998.

K. C. Toh, M. J. Todd, and R. H. Tütüncü. SDPT3—a Matlab software package for semidefinite programming. *Optimization Methods and Software*, (11):545–581, 1999. DOI: 10.1080/10556789908805762.

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

L. Vandenberghe and S. Boyd. Applications of semidefinite programming. *Applied Numerical Mathematics*, 29(3):283–299, 1999.

R.J Vanderbei. Symmetric quasi-definite matrices. *SIAM Journal on Optimization*, 5(1):100–113, 1995.

H. Waki, S. Kim, M. Kojima, M. Muramatsu, and H. Sugimoto. Algorithm 883: Sparsepop—a sparse semidefinite programming relaxation of polynomial optimization problems. *ACM Transactions on Mathematical Software*, 35(2):1–13, 2008. DOI: 10.1145/1377612.1377619.

H. Wolkowicz, R. Saigal, and L. Vandenberghe. *Handbook of semidefinite programming: theory, algorithms, and applications*, volume 27. Springer, 2000.

Y. Zhang. On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming. *SIAM Journal on Optimization*, 8: 365–386, 1998.

GERAD, Montréal, QC, Canada.
E-mail address: ahad.dehghani@gerad.ca

Desautels Faculty of Management, Management Science,, McGill University, Montréal, QC, Canada.
E-mail address: jean-louis.goffin@mcgill.ca

GERAD and Mathematics and Industrial Engineering Department, École Polytechnique, Montréal, Canada
URL: www.gerad.ca/~orban
E-mail address: dominique.orban@gerad.ca