Inexact primal-dual path-following algorithms for a special class of convex quadratic SDP and related problems

K.C. Toh [*], R. H. Tütüncü [†], and M. J. Todd [‡]

March 3, 2005

Dedicated to the memory of Jos Sturm: an outstanding friend and colleague

Abstract

We propose a primal-dual path-following Mehrotra-type predictor-corrector method for solving convex quadratic semidefinite programming (QSDP) problems. For the special case when the quadratic term has the form $\frac{1}{2}X \bullet (UXU)$, we compute the search direction at each iteration from the Schur complement equation. We are able to solve the Schur complement equation efficiently via the preconditioned symmetric quasi-minimal residual iterative solver with two appropriately constructed preconditioners. Numerical experiments on a variety of QSDPs with matrices of dimensions up to 2000 are performed and the computational results show that our methods are efficient and robust. Our methods can also be extended to linear SDP problems with upper bound constraints on primal matrix variables.

Keywords: semidefinite programming, semidefinite least squares, path-following methods, Nesterov-Todd scaling, symmetric quasi-minimum residual iteration

AMS subject classification: 90C22, 90C25, 90C51, 65F10

[*]Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543 (mattohkc@nus.edu.sg).

[†]Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213, USA (reha@cmu.edu). Research supported in part by NSF through grants CCR-9875559, CCF-0430868 and by ONR through grant N00014-05-1-0147

[‡]School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York 14853, USA (miketodd@cs.cornell.edu). Research supported in part by NSF through grant DMS-0209457 and by ONR through grant N00014-02-1-0057.

1 Introduction

Let $\mathcal{S}^n$ denote the space of real symmetric matrices of order n. Our interest is in interior-point methods for convex quadratic semidefinite programming problems, i.e., optimization problems over $\mathcal{S}^n$ in which the matrix variable is required to be positive semidefinite, where now the usual linear objective function is augmented by a quadratic function of the symmetric matrix variable. Specifically, we consider the following convex quadratic semidefinite program (QSDP)

$$\begin{array}{ll} (QSDP) & \min_X \ \frac{1}{2}X \bullet \mathcal{Q}(X) + C \bullet X \\ & \mathcal{A}(X) \ = \ b, \quad X \ \succeq \ 0, \end{array} \tag{1}$$

where $\mathcal{Q} : \mathcal{S}^n \rightarrow \mathcal{S}^n$ is a given self-adjoint positive semidefinite operator on $\mathcal{S}^n$ and $\mathcal{A} : \mathcal{S}^n \rightarrow I\!R^m$ is a linear map. The notation $X \succeq 0$ indicates that X is in $\mathcal{S}^n_+$, the cone of positive semidefinite real symmetric matrices of order n, and $U \bullet V$ denotes $\mathrm{Tr}(U^T V)$, the usual trace inner product. The adjoint of $\mathcal{A}$ with respect to the standard inner products in $\mathcal{S}^n$ and $I\!R^m$ is denoted by $\mathcal{A}^T$. Given an integer n, we let $\bar{n} = n(n+1)/2$. Consider the isometry $\mathbf{svec} : \mathcal{S}^n \rightarrow I\!R^{\bar{n}}$ defined by $\mathbf{svec}(X) = [X_{11}, \sqrt{2}X_{12}, X_{22}, \ldots, \sqrt{2}X_{1n}, \ldots \sqrt{2}X_{n-1,n}, X_{nn}]$. We write the matrix representation of $\mathcal{Q}(X)$ in the standard basis of $\mathcal{S}^n$ as $\mathbf{svec}(\mathcal{Q}(X)) = Q\mathbf{svec}(X)$, where Q is a positive semidefinite matrix in $\mathcal{S}^{\bar{n}}$; similarly, the matrix representation of $\mathcal{A}(X)$ is written as $\mathcal{A}(X) = A\mathbf{svec}(X)$, where A is a matrix in $I\!R^{m \times \bar{n}}$. The dual of (1) is given as follows:

$$\begin{array}{ll} (QSDD) & \max_{X,y,Z} \ -\frac{1}{2}X \bullet \mathcal{Q}(X) + b^T y \\ & \mathcal{A}^T(y) - \mathcal{Q}(X) + S \ = \ C, \quad S \ \succeq \ 0. \end{array} \tag{2}$$

Given any square matrices P and Q of order n, $P \circledast Q$ denotes the symmetrized Kronecker product operator on $\mathcal{S}^n$ defined by $P \circledast Q(M) = (QMP^T + PMQ^T)/2$. For details on the properties of $\circledast$ and its relation to the standard Kronecker product, see the Appendix of [30].

Consider the Cholesky factorization $Q = R^T R$, where $R \in I\!R^{p \times \bar{n}}$, with p being the rank of Q. (Note that when Q has full rank, $p = \bar{n}$.) It is readily shown that (1) can be reformulated as a standard semidefinite-quadratic-linear programming (SQLP) by introducing an additional p linear constraints and $p+1$ variables as follows:

$$\min \left\{ \frac{1}{2}t + C \bullet X \ : \ \begin{bmatrix} A \\ R \end{bmatrix} \mathbf{svec}(X) + \begin{bmatrix} 0 & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}, \ X \succeq 0, \ \|s\|_2^2 \leq t \right\}, \tag{3}$$

where the constraint $\|s\|_2^2 \leq t$ can easily be converted into a standard second order cone constraint. The computational cost required to solve the reformulated problem (3) grows at least like $O((m+p)^3)$ and the memory requirement grows like $O((m+p)^2)$. Thus, unless $m+p$ is small it is extremely expensive to solve (QSDP) by reformulating it into a standard SQLP. Given that p can be as large as $\bar{n} = \Theta(n^2)$, it is safe to say that an approach based on (3) can comfortably solve only problems with n at most 100 on a high end PC available today.

Another form of quadratic SDP has been considered in [7], namely, $\min_y\{\frac{1}{2}y^THy + b^Ty : \mathcal{A}^Ty \preceq C,\ y \in I\!R^m\}$, where H is a given positive semidefinite matrix. In this case, the Schur complement matrix arising at each interior-point iteration has the form $H + \mathcal{A}\mathcal{E}^{-1}\mathcal{F}\mathcal{A}^T$ (with $\mathcal{E}$ and $\mathcal{F}$ as in Section 3), and the computation presents no difficulties, being very similar to that for a standard linear SDP. As our interest is in problems with quadratic terms involving the matrix variables, we shall not consider this form of quadratic SDP further. In Section 2, we describe some applications leading to problems of the form (QSDP) and previous work on algorithms for such problems.

In [26], an interior-point algorithm based on reducing a primal-dual potential function was proposed to solve QSDP problems. The algorithm has an iteration complexity of $O(\sqrt{n}\ln(1/\epsilon))$ for computing an ϵ-optimal solution. At each iteration, the search direction needs to be computed from an augmented system of dimension $m + \bar{n}$. As the linear system is generally very large, the authors proposed using the conjugate gradient (CG) method to compute an approximate direction, but no preconditioning was discussed although it is crucial to do so to ensure that the CG method has a reasonable convergence rate. Furthermore, the authors do not report any numerical implementation to test the performance of their proposed method.

In this paper, we propose a primal-dual path-following Mehrotra-type predictor-corrector method for (1) and (2). For a general self-adjoint positive semidefinite $\mathcal{Q}$, the search direction at each iteration must be computed from an augmented system of dimension $m+\bar{n}$ similar to that appearing in [26]. Our ultimate goal is to investigate the efficient computation of the search direction by applying a preconditioned symmetric quasi-minimal residual (PSQMR) method to the augmented system. However, in this paper, we focus our attention on the efficient computation of the search direction only for the important special case where $\mathcal{Q}$ has the form $\mathcal{Q} = U \circledast U$, with U being a given matrix in $\mathcal{S}^n_+$. In this case, the search direction can be computed from a Schur complement system of dimension m. The cost is comparable to that of computing the AHO direction [3] for a standard linear SDP. We also discuss the conditioning of this Schur complement matrix asymptotically.

As the cost of computing the search direction in the special case can still be very expensive, we propose to use a PSQMR method to solve the governing Schur complement equation. The Schur complement matrix has the form $M = \mathcal{A}(G_1 \circledast G_1 + G_2 \circledast G_2)^{-1}\mathcal{A}^T$, where G_1, G_2 are symmetric positive definite matrices. We show that $(G_1 \circledast G_1 + G_2 \circledast G_2)^{-1}$ admits a semi-analytical expression of the form $\mathcal{J}\mathcal{D}\mathcal{J}^T$. We propose two preconditioners for M. The first preconditioner is based on a low rank approximation of the diagonal operator (with respect to the usual basis) $\mathcal{D}$. The second preconditioner is constructed by approximating the sum of two Kronecker products by a single Kronecker product.

It turns out that a Schur complement matrix of the form given in the last paragraph also arises at each interior-point iteration for a linear SDP with an upper bound. Thus we also apply our PSQMR method to solve such a problem.

Our interest here is in the efficient computation of the search direction at each iteration and we do not explicitly address the issue of the computational complexity of our algorithm. However, it is worth pointing out that for the more general monotone semidefinite complementarity problem, Kojima, Shindoh, and Hara [19] showed that

their path-following method using the dual-HKM direction has polynomial complexity. A similar result would hold for the HKM direction. Our algorithm differs in that we use the NT direction (which has advantages in computing the search direction) and also Mehrotra's predictor-corrector approach.

The rest of this article is organized as follows. As already mentioned, Section 2 focuses on examples of QSDPs and earlier algorithmic studies of these problems. In Section 3, we describe our primal-dual interior-point algorithm. Sections 4 and 5 contain our main contributions; namely the detailed discussion of the efficient computation of the search direction for general $\mathcal{Q}$ and structured $\mathcal{Q}$, respectively, including the preconditioners used in the solution of large Schur complement systems. Section 6 discusses the upper-bounded linear SDP and its relation to QSDPs. Finally, in Section 7, we provide numerical results for a variety of QSDP problems discussed in Sections 2 and 6, with matrices of order up to 2000. The results are encouraging.

2 Examples and Existing Work

One of the most common types of QSDPs encountered in the literature is the linearly constrained semidefinite least squares problem:

$$(SDLS) \quad \min_X \left\{ \left\| \mathcal{L}(X) - \widehat{K} \right\|_F : \mathcal{A}(X) = b, \;\; X \succeq 0 \right\}, \tag{4}$$

where $\mathcal{L} : \mathcal{S}^n \to \mathcal{S}^p$ is a linear map and $\widehat{K}$ is a given symmetric matrix in $\mathcal{S}^p$. Very often, $\widehat{K} = \mathcal{L}(K)$ for some $K \in \mathcal{S}^n$, and then the objective function is $\|\mathcal{L}(X-K)\|_F$: we seek a positive definite X satisfying certain linear restrictions and as close as possible to K in some suitable weighted sense. However, the EDM problem below uses a more general $\widehat{K}$ as above.

In finding the nearest correlation matrix to a given data matrix K [16], $\mathcal{A}(X) = b$ represents the constraints that fix the diagonal elements of X to one, and $\mathcal{L}$ (with $p = n$) typically has the form $\mathcal{L}(X) = U^{1/2} X U^{1/2}$ with a symmetric positive definite weight matrix U, or $\mathcal{L}(X) = U \circ X$, with a symmetric elementwise positive weight matrix U. Here and below, the notation "$\circ$" means element-wise matrix multiplication. In the QSDP formulation of these problems, $\mathcal{Q}$ takes the form $U \circledast U$ in the first case and the element-wise multiplication operator with the matrix $U \circ U$ in the second case. It is worth noting that the operator in the second case is positive definite and diagonal. For these problems, Higham proposes and analyzes a modified alternating projection solution method.

For the special case of the unweighted nearest correlation matrix problem for which $\mathcal{L}(X) = X$ (and the corresponding operator $\mathcal{Q}$ in (1) is the identity), Anjos et al. [6] proposed a feasible primal-dual interior-exterior algorithm for (1) based on inexact Gauss-Newton directions computed from overdetermined systems each with n^2 equations and $\bar{n}$ unknowns. Preconditioned CG methods with diagonal and block diagonal incomplete Cholesky preconditioning are used to compute the inexact directions. Unfortunately, preliminary numerical results obtained by the inexact Gauss-Newton approach do not show that it is numerically more efficient than the standard formulation (3).

In [23], Malick proposed a partial Lagrangian dual algorithm for solving the nearest correlation matrix problem by dualizing the linear constraints. A quasi-Newton method is used to solve the dual problem $\max\{b^T y - \|\mathcal{P}(\mathcal{A}^T y - C)\|_F^2 : y \in I\!R^m\}$, where $\mathcal{P}(U)$ is the projection of U onto $\mathcal{S}_+^n$. According to the numerical results reported in [23], this method performs very well on the nearest correlation matrix problem. More recently, Boyd and Xiao [8], apparently unaware of the work in [23], also proposed a Lagrangian dual approach combined with a projected sub-gradient method to solve the nearest correlation matrix problem.

In [27], Qi and Sun proposed a non-smooth Newton method for the same problem. Based on recent results on strongly semismooth matrix valued functions, they were able to establish quadratic convergence of their method. Numerical experiments in [27] show that the non-smooth Newton method is highly effective.

SDLS also arises from the problem of finding the nearest Euclidean distance matrix for a given weighted graph $G = (V, E, \omega)$ on p nodes [4]. Let $\widehat{K}_G$ be a $p \times p$ matrix whose elements are given by $k_{ij} = \omega_{ij}$ if $(i, j) \in E$, and $k_{ij} = 0$ if $(i, j) \notin E$. We seek points $q_1, q_2, \ldots, q_p$ in R^n for $n = p-1$ such that $\|q_i - q_j\|^2$ is close to k_{ij} for $(i, j) \in E$. Then the optimization problem (see [4]) is the following:

$$(EDM) \quad \min_X \{ \left\| \mathcal{L}(X) - \widehat{K}_G \right\|_F : \mathcal{A}(X) = b, \;\; X \succeq 0 \}, \tag{5}$$

where now $\mathcal{L}(X) = \Sigma \circ (\mathrm{diag}(VXV^T)\, e^T + e\, \mathrm{diag}(VXV^T)^T - 2VXV^T)$. Here Σ is the adjacency matrix of the graph G, and V is a $p \times n$ matrix such that $V^T e = 0$ and $V^T V = I_n$. If we factor the solution X as RR^T, then the q's can be taken to be the columns of $R^T V^T$.

For semidefinite least squares problems (4) arising from the nearest Euclidean distance matrix problem Alfakih et al. [4] proposed a primal-dual interior-point algorithm based on the Gauss-Newton approach to solve the perturbed optimality conditions. Unfortunately, the linear system that needs to be solved at each iteration has dimension about n^2. Consequently, only small problems with $n \leq 50$ can be comfortably solved on a standard PC.

3 A primal-dual path-following interior-point algorithm

We propose to solve the primal-dual pair (1) and (2) using primal-dual path-following methods based on their perturbed KKT conditions:

$$\begin{array}{rcrcrcl} -\mathcal{Q}(X) & + & \mathcal{A}^T(y) & + & S & = & C, \quad S \succeq 0 \\ \mathcal{A}(X) & & & & & = & b, \quad X \succeq 0 \\ & & XS & & & = & \nu I, \end{array} \tag{6}$$

where $\nu > 0$ is a positive parameter. Given the current iterate (X, y, S) with X and S positive definite, the search direction $(\Delta X, \Delta y, \Delta S)$ at each interior-point iteration is

the solution of the following symmetrized Newton system:

$$\begin{array}{rcrcrcl} -\mathcal{Q}(\Delta X) & + & \mathcal{A}^T(\Delta y) & + & \Delta S & = & R_d := C - S - \mathcal{A}^T y + \mathcal{Q}(X) \\ \mathcal{A}(\Delta X) & & & & & = & r_p := b - \mathcal{A}(X) \\ \mathcal{E}(\Delta X) & & & + & \mathcal{F}(\Delta S) & = & R_c := \sigma\mu I - H_P(XS), \end{array} \tag{7}$$

where $\mathcal{E}$ and $\mathcal{F}$ are linear operators on $\mathcal{S}^n$ that depend on the symmetrization scheme $H_P(\cdot)$ chosen, with P being the symmetrization matrix; for more details, see for example [30]. Here, $\mu = X \bullet S/n$, and $\sigma \in (0, 1)$ is the centering parameter.

We start with an initial iterate (X^0, y^0, S^0) with $X^0, S^0 \succ 0$, and step-length parameter $\tau^0 = 0.9$. The details of an iteration of our primal-dual path-following algorithm are explained below. In this description we denote the current and the next iterates by (X, y, S) and (X^+, y^+, S^+), and the current and the next step-length parameters by τ and τ^+, respectively.

Algorithm IP-QSDP.

- Set $\mu = X \bullet S/n$.
- (Convergence test)

  Stop the iteration if the accuracy measure ϕ is sufficiently small, where

$$\phi \;=\; \max\left\{ \frac{X \bullet S}{1 + |\text{pobj}| + |\text{dobj}|}\,,\; \frac{\|r_p\|_2}{1 + \|b\|_2}\,,\; \frac{\|R_d\|_F}{1 + \|C\|_F} \right\} \tag{8}$$

  with r_p, R_d defined as in (7), pobj $= \frac{1}{2} X \bullet \mathcal{Q}(X) + C \bullet X$, and dobj $= -\frac{1}{2} X \bullet \mathcal{Q}(X) + b^T y$.
- (Predictor step)

  Compute the predictor search direction $(\delta X, \delta y, \delta S)$ from (7) by choosing $\sigma = 0$.
- (Predictor step-length)

  Compute

$$\alpha_p \;=\; \min\left(1, \tau\,\alpha\right). \tag{9}$$

  Here α is the maximum step length that can be taken so that $X + \alpha\delta X$ and $S + \alpha\delta S$ remain positive definite.
- (Centering rule)

  Set $\sigma = (X + \alpha_p \delta X) \bullet (S + \alpha_p \delta S)/X \bullet S$.
- (Corrector step)

  Compute the search direction $(\Delta X, \Delta y, \Delta S)$ from (7), with R_c replaced by

$$R_c' \;=\; \sigma\mu I - H_P(XS + \delta X \delta S).$$

- (Corrector step-length)

  Compute α_c as in (9) but with $(\delta X, \delta S)$ replaced by $(\Delta X, \Delta S)$.

- Update (X, y, S) to (X^+, y^+, S^+) by

$$X^+ = X + \alpha_c \Delta X, \quad y^+ = y + \alpha_c \Delta y, \quad S^+ = S + \alpha_c \Delta S. \tag{10}$$

- Update the step-length parameter by $\tau^+ = 0.9 + 0.08\alpha_c$.

4 Computation of search direction: general $\mathcal{Q}$

The linear system of equations (7) is a non-symmetric system of dimension $m+n(n+1)$. This is generally a very large system even for a moderate n, say, $n = 200$. Thus it is extremely expensive to solve the system directly. By eliminating ΔS, we get the following augmented equation with dimension $m + \bar{n}$:

$$\underbrace{\begin{bmatrix} -\mathcal{H} & \mathcal{A}^T \\ \mathcal{A} & 0 \end{bmatrix}}_{\mathcal{B}} \begin{bmatrix} \Delta X \\ \Delta y \end{bmatrix} = \begin{bmatrix} R_d - \mathcal{F}^{-1} R_c \\ r_p \end{bmatrix}, \tag{11}$$

where

$$\mathcal{H} = \mathcal{F}^{-1}\mathcal{E} + \mathcal{Q}. \tag{12}$$

Note that under the assumptions that $\mathcal{A}$ is a surjective map and $X, S \succ 0$, the matrix $\mathcal{B}$ is nonsingular when the scaling matrix P is chosen to be in the Monteiro-Zhang family described in [25]. Specific members of the family include the Nesterov-Todd scaling [30] and the HKM scaling, independently proposed by Helmberg et al. [15] and Kojima et al. [19], and later rederived from a different viewpoint by Monteiro [24]. For the NT scaling, we have $\mathcal{F}^{-1}\mathcal{E} = W^{-1} \circledast W^{-1}$, where W is the unique matrix in $\mathcal{S}^n_+$ satisfying $WSW = X$. For the HKM scaling, we have $\mathcal{E}^{-1}\mathcal{F} = X \circledast S^{-1}$, but unfortunately, $\mathcal{F}^{-1}\mathcal{E}$ does not have such a simple analytical expression. For the dual HKM scaling, we do not have a simple expression for $\mathcal{E}^{-1}\mathcal{F}$, but $\mathcal{F}^{-1}\mathcal{E} = S \circledast X^{-1}$.

By further eliminating ΔX, we get the Schur complement equation of dimension m below:

$$\underbrace{\mathcal{A}\mathcal{H}^{-1}\mathcal{A}^T}_{M} \Delta y \;=\; h \;:=\; r_p + \mathcal{A}\mathcal{H}^{-1}\Big(R_d - \mathcal{F}^{-1} R_c\Big). \tag{13}$$

For a general $\mathcal{Q}$, even for the simple case where $\mathcal{Q}$ is a diagonal operator, $\mathcal{H}$ cannot be inverted with moderate cost, so the computation of the Schur complement matrix M is extremely expensive. Thus unlike the case of linear SDP, computing the direction based on (13) is computationally not feasible with the possible exception of the case when $\mathcal{Q}$ is of the form $U \circledast U$. The best alternative seems to be to compute the direction based on the augmented equation (11). Using (11) instead of (13), we avoid the costly construction of the matrix M. However, the coefficient matrix $\mathcal{B}$ in (11) is typically very large and the matrix $\mathcal{H}$ in its $(1,1)$ block is typically dense. As a result, solving (11) by a direct method such as the LDL^T factorization method is out of consideration. It is necessary to use an iterative solver such as the preconditioned symmetric quasi-minimal residual (PSQMR) method [10] to solve (11).

We will discuss in the next section the special case where $\mathcal{Q} = U \circledast U$ for which $\mathcal{H}^{-1}$ has an analytical expression, and the computational cost of the Schur complement matrix is more moderate.

Remark 4.1 *To avoid the need to handle the sum $\mathcal{F}^{-1}\mathcal{E} + \mathcal{Q}$ whose inverse is expensive to compute, we explore another route starting from the augmented system (11). Consider the Cholesky factorization $\mathcal{Q} = \mathcal{R}^T\mathcal{R}$, where the matrix representation of $\mathcal{R}$ is a matrix in $\mathbb{R}^{p\times\bar{n}}$. Let $\Delta Z = -\mathcal{R}\Delta X$. Then the augmented system can be rewritten as follows:*

$$\begin{bmatrix} -\mathcal{F}^{-1}\mathcal{E} & \mathcal{A}^T & \mathcal{R}^T \\ \mathcal{A} & 0 & 0 \\ \mathcal{R} & 0 & I \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta y \\ \Delta Z \end{bmatrix} = \begin{bmatrix} R_d - \mathcal{F}^{-1}R_c \\ r_p \\ 0 \end{bmatrix}. \tag{14}$$

The introduction of the auxiliary variable ΔZ is motivated by the paper [21] for convex quadratic programming in $\mathbb{R}^n$. Upon eliminating ΔX from (14), we get the following linear system of dimension $m + p$:

$$\begin{bmatrix} M_{AA} & M_{RA}^T \\ M_{RA} & I + M_{RR} \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta Z \end{bmatrix} = \begin{bmatrix} h_y \\ h_Z \end{bmatrix} \tag{15}$$

where $M_{AA} = \mathcal{A}\mathcal{E}^{-1}\mathcal{F}\mathcal{A}^T$, $M_{RA} = \mathcal{R}\mathcal{E}^{-1}\mathcal{F}\mathcal{A}^T$, $M_{RR} = \mathcal{R}\mathcal{E}^{-1}\mathcal{F}\mathcal{R}^T$, and

$$h_y \;=\; r_p + \mathcal{A}\mathcal{E}^{-1}\mathcal{F}R_d - \mathcal{A}\mathcal{E}^{-1}R_c, \quad h_Z \;=\; \mathcal{R}\mathcal{E}^{-1}\mathcal{F}R_d - \mathcal{R}\mathcal{E}^{-1}R_c.$$

Unfortunately, the system (15) has exactly the same dimension and structure as the Schur complement matrix arising at each interior-point iteration for the standard SQLP (3). In particular, note that the matrices M_{AA}, M_{RA}, M_{RR} are typically all dense, even if $\mathcal{R}$ is a diagonal operator. Thus, unless $m + p$ is small, the approach based on (15) would not offer any computational advantage over the augmented system (11). As such, we shall not consider this approach any further in this paper.

When using an iterative method to solve (11), it is necessary to know how accurately one must solve the equation. For this purpose, we will construct a stopping condition based on the residual norm of the computed search direction with respect to (7).

Proposition 4.1 *(a) Suppose $(\Delta X, \Delta y)$ is computed from (11) with residual vector given by $(\eta_1\,,\,\eta_2)$. Suppose that ΔS is computed exactly from*

$$\Delta S \;=\; R_d - \mathcal{A}^T\Delta y + \mathcal{Q}(\Delta X).$$

Then the residual vector associated with the direction $(\Delta X, \Delta y, \Delta S)$ for (7) is given by $(0, \eta_2, -\mathcal{F}(\eta_1))$.

(b) Suppose Δy is computed from (13) with residual vector given by ξ. Suppose that ΔX and ΔS are computed exactly as follows:

$$\Delta X \;=\; \mathcal{H}^{-1}\Big(\mathcal{A}^T\Delta y - (R_d - \mathcal{F}^{-1}R_c)\Big), \quad \Delta S \;=\; R_d - \mathcal{A}^T\Delta y + \mathcal{Q}(\Delta X).$$

Then the residual vector associated with the direction $(\Delta X, \Delta y, \Delta S)$ for (7) is given by $(0, \xi, 0)$.

Proof. We omit the proof since it is straightforward. □

We deem the computed $(\Delta X, \Delta y)$ from (11) to be sufficiently accurate if the following relative stopping condition is satisfied:

$$\|(\eta_2\,,\,-\mathcal{F}(\eta_1))\| \;\le\; \kappa\,\|(r_p\,,\,R_d\,,\,R_c)\|\,,$$

where $\kappa \in (0,1)$ is an accuracy parameter. Similarly, we deem the computed Δy from (13) to be sufficiently accurate if $\|\xi\| \le \kappa\,\|(r_p\,,\,R_d\,,\,R_c)\|$. In the numerical experiments in Section 7, we take $\kappa = 10^{-3}$.

4.1 Conditioning of M

The convergence rate of a Krylov subspace method such as the SQMR method depends heavily on the condition number of the coefficient matrix of the linear system being solved. Thus it is of interest to analyze the conditioning of the coefficient matrix M in (13). For simplicity, we assume in this subsection that the NT scaling is used, that is,

$$\mathcal{H} \;=\; W^{-1} \circledast W^{-1} + \mathcal{Q}. \tag{16}$$

This assumption is without loss of generality: for points on the central path, which we consider below, the NT scaling, HKM scaling, and dual HKM scaling coincide. We further assume here that $\mathcal{Q}$ is positive definite. We will show that, under suitable additional assumptions, the condition number of M remains bounded on the central path of (1) and (2).

It is easy to prove the following proposition.

Proposition 4.2 *Suppose that $\mathcal{Q}$ is positive definite. Then $\|M\| \le \left\|\mathcal{A}\mathcal{Q}^{-1}\mathcal{A}^T\right\|$.*

Proof. The result follows from the fact that $\mathcal{H} \succeq \mathcal{Q} \succ 0$. □

Assume that the problems (1) and (2) are strictly feasible, and that $\mathcal{A}$ is surjective. These are necessary and sufficient conditions for the existence and uniqueness of solutions (X_ν, y_ν, S_ν) of the central path equations (6). Also, these solutions converge to some (X_*, y_*, S_*) as ν tends to zero; see Halicka, de Klerk, and Roos [13], and Luo, Sturm, and Zhang [22]. Both papers additionally contain many results on the behavior of X_ν and S_ν. We further assume that X_* is primal nondegenerate and that strict complementarity holds in the sense of Alizadeh, Haeberly, and Overton [2]. Thus (y_*, S_*) is the unique optimal dual solution, and the ranks of X_* and S_* sum to n. We will show that, under these conditions, the condition number of M_ν, the Schur complement matrix in (13) corresponding to (X_ν, y_ν, S_ν), remains bounded as ν tends to zero.

Let us suppose not, and choose a monotonically decreasing sequence $\{\nu_k\}$ such that $\lim_{k\to\infty} \nu_k = 0$ and $\lim_{k\to\infty} \|M_{\nu_k}\|_2 \left\|M_{\nu_k}^{-1}\right\|_2 = \infty$. For simplicity of notation, we write M_k, X_k, S_k, etc., for M_{ν_k}, X_{ν_k}, S_{ν_k}, and so on. Since X_k and S_k commute, there is an orthogonal matrix P_k that simultaneously diagonalizes X_k and S_k so that

$$X_k = P_k \Lambda_k P_k^T, \quad S_k = P_k \Sigma_k P_k^T,$$

where the eigenvalue matrices

$$\Lambda_k = \mathrm{Diag}(\lambda_1^k, \ldots, \lambda_n^k), \quad \Sigma_k = \mathrm{Diag}(\sigma_1^k, \ldots, \sigma_n^k)$$

satisfy $\lambda_i^k \sigma_i^k = \nu_k$, and the eigenvalues are ordered such that

$$\lambda_1^k \geq \ldots \geq \lambda_n^k > 0, \quad 0 < \sigma_1^k \leq \ldots \leq \sigma_n^k.$$

Let P_* be a limit point of the set $\{P_k\}$. We refine the sequence if necessary so that $\{P_k\}$ converges to P_*. Then P_* is an orthogonal matrix that simultaneously diagonalizes X_* and S_* with

$$X_* = P_* \Lambda_* P_*^T, \quad S_* = P_* \Sigma_* P_*^T, \tag{17}$$

where

$$\Lambda_* = \mathrm{Diag}(\lambda_1^*, \ldots, \lambda_n^*), \quad \Sigma_* = \mathrm{Diag}(\sigma_1^*, \ldots, \sigma_n^*)$$

satisfy $\lambda_i^* \sigma_i^* = 0$, and

$$\lambda_1^* \geq \ldots \geq \lambda_r^* > \lambda_{r+1}^* = \ldots = \lambda_n^* = 0, \quad 0 = \sigma_1^* = \ldots = \sigma_{n-s}^* < \sigma_{n-s+1}^* \leq \ldots \leq \sigma_n^*,$$

where r and s are the ranks of X_* and S_*, respectively.

By the strict complementarity assumption, $r + s = n$. For k sufficiently large, the NT scaling matrix associated with (X_k, S_k), given by $W_k = P_k D_k^{-1} P_k^T$ with $D_k = \Lambda_k^{-1/2} \Sigma_k^{1/2}$, has r and s eigenvalues of the order $\Theta(1/\sqrt{\nu_k})$ and $\Theta(\sqrt{\nu_k})$, respectively. This implies that $W_k^{-1} \circledast W_k^{-1}$ has $\bar{r}$, rs, and $\bar{s}$ eigenvalues of the order $\Theta(\nu_k)$, $\Theta(1)$, and $\Theta(1/\nu_k)$, respectively.

Recall that M_k denotes the Schur complement matrix in (13) corresponding to (X_k, y_k, S_k). We observe that

$$M_k \;=\; \mathcal{A}\mathcal{P}_k \Big(D_k \circledast D_k + \widetilde{\mathcal{Q}}_k \Big)^{-1} \mathcal{P}_k^T \mathcal{A}^T,$$

where $\mathcal{P}_k = P_k \circledast P_k$, and $\widetilde{\mathcal{Q}}^k = \mathcal{P}_k^T \mathcal{Q} \mathcal{P}_k$. Assume that the eigenvalues of $W_k^{-1} \circledast W_k^{-1}$ are sorted in increasing order. Consider the following partition

$$D_k \circledast D_k + \widetilde{\mathcal{Q}}_k \;=\; \begin{bmatrix} \mathcal{D}_1^k + \widetilde{\mathcal{Q}}_{11}^k & \widetilde{\mathcal{Q}}_{12}^k \\ \widetilde{\mathcal{Q}}_{21}^k & \widetilde{\mathcal{Q}}_{22}^k + \mathcal{D}_2^k \end{bmatrix},$$

where $\mathcal{D}_1^k$ and $\mathcal{D}_2^k$ correspond to diagonal matrices whose diagonal elements consist of the first $\bar{r} + rs$, and the last $\bar{s}$ eigenvalues of $W_k^{-1} \circledast W_k^{-1}$, respectively. Note that $\left\| \mathcal{D}_1^k \right\| = O(1)$, and that the diagonal entries of the matrix corresponding to $\mathcal{D}_2^k$ are all $\Theta(1/\nu_k)$. Note also that $\widetilde{\mathcal{Q}}_{11}^k$ and $\widetilde{\mathcal{Q}}_{22}^k$ are positive definite since $\mathcal{Q}$ is assumed to be positive definite in this subsection. By using the formula for the inverse of a 2×2 block matrix in [28, p.389], one can show that

$$\Big(D_k \circledast D_k + \widetilde{\mathcal{Q}}_k \Big)^{-1} \;=\; \begin{bmatrix} (\mathcal{D}_1^k + \widetilde{\mathcal{Q}}_{11}^k)^{-1} & 0 \\ 0 & 0 \end{bmatrix} + O(\nu_k). \tag{18}$$

Thus

$$M_k \;=\; \mathcal{A}\mathcal{P}_{k,1}\Big(\mathcal{D}_1^k + \widetilde{\mathcal{Q}}_{11}^k\Big)^{-1}\mathcal{P}_{k,1}^T\mathcal{A}^T + O(\nu_k),$$

where $\mathcal{P}_{k,1}$ is the submatrix obtained from $\mathcal{P}_k$ by extracting its first $\bar{r} + rs$ columns.

Similarly, let $\mathcal{P}_{*,1}$ be the submatrix obtained from $\mathcal{P}_* := P_* \circledast P_*$ by extracting its first $\bar{r} + rs$ columns, and $\widetilde{\mathcal{Q}}^* = \mathcal{P}_*^T\mathcal{Q}\mathcal{P}_*$, with $\widetilde{\mathcal{Q}}_{11}^*$ its part corresponding to $\widetilde{\mathcal{Q}}_{11}^k$.

The discussion above leads to the following conclusion:

Theorem 4.1 *Suppose that $\mathcal{Q}$ is positive definite, that the problems (1) and (2) are strictly feasible, and that primal nondegeneracy and strict complementarity hold for the optimal solution (X_*, y_*, S_*). Then there exists a positive constant c such that $\limsup_{k\to\infty} \left\|M_k^{-1}\right\|_2 \leq c\left\|M_*^{-1}\right\|_2 < \infty$, where $M_* = \mathcal{A}\mathcal{P}_{*,1}(\widetilde{\mathcal{Q}}_{11}^*)^{-1}\mathcal{P}_{*,1}^T\mathcal{A}^T$.*

Proof. Since $\lim_{k\to\infty}\widetilde{\mathcal{Q}}_{11}^k = \widetilde{\mathcal{Q}}_{11}^*$ (which is positive definite) and $\{\left\|\mathcal{D}_1^k\right\|\}$ is bounded, there is a constant $c > 0$ such that for sufficiently large k, we have $c\widetilde{\mathcal{Q}}_{11}^* \succeq \mathcal{D}_1^k + \widetilde{\mathcal{Q}}_{11}^k$. This implies that $(\widetilde{\mathcal{Q}}_{11}^*)^{-1} \preceq c(\mathcal{D}_1^k + \widetilde{\mathcal{Q}}_{11}^k)^{-1}$, and

$$\mathcal{A}\mathcal{P}_{k,1}(\widetilde{\mathcal{Q}}_{11}^*)^{-1}\mathcal{P}_{k,1}^T\mathcal{A}^T \preceq c(M_k + O(\nu_k)). \tag{19}$$

Now we show that the matrix on the left-hand side is positive definite for sufficiently large k, which follows if M_* is. But this holds as long as $\mathcal{A}\mathcal{P}_{*,1}$ is surjective. Suppose the symmetric matrices A_i are defined by $\mathcal{A}^T y = \sum_i y_i A_i$. The surjectivity condition is equivalent to saying that the matrices $P_*^T A_i P_*$, when their trailing $s \times s$ submatrices are replaced by zeroes (this operation corresponds to considering only the first $\bar{r} + rs$ columns of $\mathcal{P}_*$), are linearly independent. But by Theorem 6 of [2], this is equivalent to the primal nondegeneracy of X_*. Now, by considering the smallest eigenvalues of the matrices on both sides of (19) and taking the limit infimum, we get $\lambda_{\min}(M_*) \leq c\,\liminf_{k\to\infty}\lambda_{\min}(M_k)$. The required result follows by noting that the 2-norm of the inverse of a symmetric positive definite matrix is equal to the reciprocal of its smallest eigenvalue. ∎

Corollary 4.1 *Under the assumptions of the theorem, the condition number of M_ν remains bounded as ν tends to zero.*

Proof. The result follows straightforwardly from Proposition 4.2 and Theorem 4.1. Indeed, we have shown that assumption that $\left\|M_\nu^{-1}\right\|_2$ is unbounded leads to a contradiction. ∎

Motivated by a result in the paper [27], we can show that X^* is primal nondegenerate when $\mathcal{A}$ is the diagonal map and $b > 0$.

Proposition 4.3 *Consider the linear map $\mathcal{A} : \mathcal{S}^n \to I\!R^n$ defined by $\mathcal{A}(X) = \mathrm{diag}(X)$, and assume that $b > 0$. Then X^* is primal nondegenerate.*

Proof. We note that the adjoint $\mathcal{A}^T : I\!R^n \to \mathcal{S}^n$ is given by $\mathcal{A}^T(y) = \mathrm{Diag}(y)$. Suppose P_* and Λ_* are defined as in (17). Since $b = \mathcal{A}(X_*) = \mathrm{diag}(P_*\Lambda_* P_*^T)$, we have $b = \mathrm{diag}(\sum_{i=1}^r \lambda_i^* P_{*,i} P_{*,i}^T)$, where $P_{*,i}$ denotes the ith column of P_* and again r is the rank of X_*.

Consider the linear system $(P_*^T \mathrm{Diag}(y) P_*)_{ij} = 0$ for all $1 \le i \le r$, $1 \le j \le n$; or equivalently, $P_{*,i}^T \mathrm{Diag}(y) P_* = 0$ for all $1 \le i \le r$. We want to show that y must be zero, so that by the results of [2] X_* is primal nondegenerate. We have $P_{*,i}^T \mathrm{Diag}(y) = 0$ for all $1 \le i \le r$ since P_* is nonsingular. Taking the transpose yields $\mathrm{Diag}(y) P_{*,i} = 0$ for all $1 \le i \le r$. Multiply this equation on the right by $\lambda_i^* P_{*,1}^T$ and sum over i to get $\mathrm{Diag}(y)(\sum_{i=1}^r \lambda_i^* P_{*,i} P_{*,i}^T) = 0$. Taking the diagonal of this matrix gives $\mathrm{Diag}(y) b = 0$. The assumption made on b implies that we have $y = 0$, and thus X^* is primal nondegenerate. ∎

5 Computation of search direction when $\mathcal{Q} = U \circledast U$

For the special case where $\mathcal{Q} = U \circledast U$ with $U \succeq 0$, it is possible to compute the inverse of the mapping $\mathcal{H}$ given in (12) with a more moderate cost if the NT direction is used. The motivation for choosing the NT scaling instead of other scalings in the Monteiro-Zhang family will become clear after we have presented Lemma 5.1 below. Note that for such a choice of $\mathcal{Q}$ and scaling, we have

$$X \bullet \mathcal{Q}(X) \;=\; \left\| U^{1/2} X U^{1/2} \right\|_F^2,$$

and $\mathcal{H}$ has the form

$$\mathcal{H} = U \circledast U + W^{-1} \circledast W^{-1}. \tag{20}$$

The inverse of such an $\mathcal{H}$ can be computed via one of the two procedures described in the following lemma.

Lemma 5.1 *Let $\mathcal{H}$ be as in (20). Then*

$$\mathcal{H}^{-1} \;=\; (P \circledast P)\Big(\mathcal{I} + D \circledast D\Big)^{-1} (P \circledast P)^T, \tag{21}$$

where $\mathcal{I}$ is the identity operator on $\mathcal{S}^n$ and P and the diagonal matrix D are computed in one of the two following ways.

a) Suppose the Cholesky-like factorization $W = R^T R$ and the eigenvalue decomposition $RUR^T = QDQ^T$ are computed. Then set $P = R^T Q$.

b) Assume that U is positive definite. Suppose the Cholesky factorization $U = L^T L$ and the eigenvalue decomposition $LWL^T = Q\hat{D}Q^T$ are computed. Then set $P = L^{-1}Q$ and $D = \hat{D}^{-1}$.

Proof.

a) Given any $V \in \mathcal{S}^n$, $\mathcal{H}Y = V$ implies that

$$UYU + W^{-1}YW^{-1} = V.$$

Thus

$$(RUR^T)(R^{-T}YR^{-1})(RUR^T) + R^{-T}YR^{-1} \;=\; RVR^T.$$

Let $\tilde{Y} = R^{-T}YR^{-1}$. Then we have

$$(RUR^T)\tilde{Y}(RUR^T) + \tilde{Y} \;=\; (RVR^T).$$

With the above identity and the eigenvalue decomposition of RUR^T, it is readily shown that

$$Y \;=\; (R^TQ \circledast R^TQ)\Big(\mathcal{I} + D \circledast D\Big)^{-1}(R^TQ \circledast R^TQ)^TV,$$

and the required result is shown.

b) We first note that $W^{-1} = L^TQDQ^TL$. $\mathcal{H}$ can be written as follows (see also the multiplication formulas for $\circledast$ provided in the Appendix of [30]):

$$\begin{aligned}\mathcal{H} &= W^{-1} \circledast W^{-1} + U \circledast U = L^TQDQ^TL \circledast L^TQDQ^TL + L^TL \circledast L^TL \\ &= \Big(L^TQ \circledast L^TQ\Big)\,(D \circledast D + I \circledast I)\,\Big(L^TQ \circledast L^TQ\Big)^T.\end{aligned}$$

Now, since $(G \circledast G)^{-1} = G^{-1} \circledast G^{-1}$ for an invertible G and $P = (L^TQ)^{-T}$, (21) follows easily.

□

Since a Cholesky-like factorization, $W = R^TR$, is typically performed in the process of computing W (see [30]), the additional work required for the first method in Lemma 5.1 is only the computation of the matrix RUR^T, its eigenvalue decomposition, and the matrix product $P = R^TQ$. In contrast, for the second method, one needs to compute (but only once) an additional Cholesky factorization of the U matrix, the matrix product LWL^T and its eigenvalue decomposition as in the first method, and $P = L^{-1}Q$, which takes comparable work to the product R^TQ in the first method. Thus the work required in the two methods is comparable. An important exception is the case when U is the identity matrix. Then, the first method requires the computation of the dense matrix products RR^T and R^TQ in addition to the eigenvalue decomposition required for both methods. In any case, the differences between the flop counts required for these two methods will be relatively insignificant given the more expensive parts of the iteration, such as the computation of the M matrix.

Using (21), the Schur complement matrix M in (13) becomes:

$$M \;:=\; \mathcal{A}\mathcal{H}^{-1}\mathcal{A}^T \;=\; \mathcal{A}(P \circledast P)\Big(\mathcal{I} + D \circledast D\Big)^{-1}(P \circledast P)^T\mathcal{A}^T, \tag{22}$$

where the term $\mathcal{I} + D \circledast D$ is a positive definite diagonal operator. The complexity of computing M is $4\frac{1}{3}mn^3 + \frac{1}{2}m^2n^2$ floating point operations if sparsity in $\mathcal{A}$ is totally ignored; see [33]. But even if sparsity in $\mathcal{A}$ is exploited, the structural formula in (22) makes it non-conducive for one to apply the techniques presented in [11] to exploit the sparsity; thus the computational complexity is generally not much lower, and a savings of at most 50% is typical. (The problem is that, if A_i denotes $\mathcal{A}^T$ times a unit vector, it appears necessary to compute *a*ll entries of $P^T A_i P$ rather than just those corresponding to a nonzero in some A_h, as in linear SDP.) Note that when $m \approx n$, the computational complexity grows like $O(n^4)$.

Remark 5.1 *(a) Our ability to compute $\mathcal{H}^{-1}$ via the semi-analytical formulas presented in Lemma 5.1 depend critically on $\mathcal{H}$ having the form $\mathcal{H} = U \circledast U + G \circledast G$. Even for a slight change to the form $\mathcal{H} = U \circledast U + G \circledast K$, the technique used in the proof of Lemma 5.1 would fail, and we are not aware of the existence of an analogous semi-analytical formula. This is the reason for our focus on the special case $\mathcal{Q} = U \circledast U$ as well as the use of the Nesterov-Todd scaling in the computation of the search directions.*

(b) The matrix of the form given in (22) also arises from the Schur complement equation when solving a standard linear SDP by primal-dual interior point methods using the AHO direction. The Schur complement matrix in that case has the form

$$\mathcal{A}(P \circledast P)\Big(I \circledast D\Big)^{-1}(P \circledast P)^T \mathcal{B}^T, \tag{23}$$

where $\mathcal{B} = \mathcal{A}(I \circledast X)$, P is orthogonal, and D is diagonal. Thus, the cost of computing M in (22) is comparable to that of (23) for the AHO direction associated with a standard linear SDP.

(c) When the search direction is computed from (22) at each interior-point iteration, the full eigenvalue decompositions of two $n \times n$ dense matrices are required: one for computing the NT scaling matrix and the other for obtaining the semi-analytical formula of $\mathcal{H}^{-1}$. Each of these eigenvalue decompositions has a computational complexity of $26/3n^3$ [9]. This can easily become the dominant computational cost when n is large because it is difficult to exploit sparsity in computing a full eigenvalue decomposition. For the machine that we use to conduct the numerical experiments in Section 7, the computation of such a decomposition for a dense 2000×2000 matrix takes about 56.7 seconds.

To illustrate part (b) of Remark 5.1 further we examine the calculation of the search directions using formulas given in Lemma 5.1 in a bit more detail. The similarities with the calculation of the AHO direction will become apparent. For this purpose, we recall the nearest correlation matrix problem we discussed in Section 2. To keep things simple, we do not consider a weighting matrix. Given a $n \times n$ symmetric matrix K, the nearest correlation matrix to K can be found by solving the following problem:

$$\begin{array}{lll} \min & \frac{1}{2}X \bullet X + C \bullet X & \\ \text{s.t.} & E_{ii} \bullet X = 1, & i = 1, \ldots, n, \\ & X \succeq 0, & \end{array} \tag{24}$$

where $C = -K$ and E_{ii} is the $n \times n$ matrix whose only non-zero entry is a 1 in the (i,i) position. We now describe the computation of the Schur complement matrix $M = \mathcal{A}\mathcal{H}^{-1}\mathcal{A}^T$ for this problem using (21) and part (b) of the lemma. First, note that $U = I$, so we choose $L = I$. Let $Q\hat{D}Q^T = W$ be the eigenvalue decomposition of W and $D = \hat{D}^{-1}$. Observe that (21) simplifies to:

$$\mathcal{H}^{-1} \;=\; (Q \circledast Q)\,(\mathcal{I} + D \circledast D)^{-1}\left(Q^T \circledast Q^T\right).$$

Next, we observe that $(Q^T \circledast Q^T)(E_{ii}) = Q^T E_{ii} Q = Q_{i,:}^T Q_{i,:}$ where $Q_{i,:}$ denotes the i^{th} row of Q. Therefore, the i^{th} column of $M := \mathcal{A}\mathcal{H}^{-1}\mathcal{A}^T$ can be found as follows:

1. $T_i^1 = Q_{i,:}^T Q_{i,:}$.
2. $\left[T_i^2\right]_{jk} = \dfrac{\left[T_i^1\right]_{jk}}{1 + d_j d_k},\ j,k = 1,\ldots,n.$
3. $T_i^3 = Q T_i^2 Q^T$.
4. $M_{:,i} = \text{diag}(T_i^3)$.

Above, the d_i's are the diagonal elements of the matrix D, i.e., the reciprocals of the eigenvalues of the W matrix and the T_i^j's for $j = 1,2,3$ are temporary matrices that can be discarded after Step 4. While the first and second steps of this computation require $\Theta(n^2)$ floating point operations (flops), the third step needs $\Theta(n^3)$ flops and the overall complexity of computing the Schur complement matrix is $\Theta(n^4)$.

Since only the main diagonal elements of T_i^3 are eventually needed in computing $M_{:,i}$, it appears that the $\Theta(n^3)$ complexity in Step 3 can be reduced by computing just the diagonal elements of T_i^3. However, the alternative approach of computing the diagonal elements via the formula $Q_{j,:} T_i^2 Q_{j,:}^T$, $j = 1,\ldots,n$, also needs $\Theta(n^3)$ flops.

5.1 Preconditioners for M, part I

Let $\widetilde{\mathcal{A}} = \mathcal{A}(P \circledast P)$, and consider the matrix in (22):

$$M \;=\; \widetilde{\mathcal{A}}\,(\mathcal{I} + D \circledast D)^{-1}\widetilde{\mathcal{A}}^T. \tag{25}$$

Here P and D are computed as in part (a) or part (b) of Lemma 5.1; in the first case PP^T is easily seen to be W, while in the second it is U^{-1}, so in either case $(P \circledast P)(P \circledast P)^T = PP^T \circledast PP^T$ is easily obtained. The complexity of computing the matrix in (25) is comparable to that for the AHO direction because sparsity in $\mathcal{A}$ cannot be exploited fully. To lower the computational complexity in solving (13), an alternative is to use the PSQMR method with an appropriate preconditioner. In this case, the matrix M need not be constructed explicitly, and only matrix-vector multiplications of the form My for a given y are required. It is easy to show, based on the formula for $\mathcal{H}^{-1}$ given in Lemma 5.1, that each matrix-vector product My costs $2mn^2 + \frac{14}{3}n^3$ flops if sparsity in $\mathcal{A}$ is totally ignored. We observe that a matrix product of the form PVP^T for a symmetric V can be computed at a cost of $\frac{7}{3}n^3$ flops by first computing the Aasen decomposition of V.

We note that for solving a symmetric positive definite system, the PCG method is most commonly used, but since PSQMR has very similar convergence behavior and computational cost as PCG, we will continue to use PSQMR here.

We try to approximate the middle term $(\mathcal{I} + D \circledast D)^{-1}$ in (25) as the sum of a small number of terms of the form $\Lambda_k \circledast \Lambda_k$. Specifically, for a fixed positive integer q, suppose that Λ_k's, $k = 1, \ldots, q$, are diagonal matrices such that

$$\sum_{k=1}^{q} \alpha_k \Lambda_k \circledast \Lambda_k \;\approx\; (\mathcal{I} + D \circledast D)^{-1}, \tag{26}$$

where each α_k is a scalar. We will make clear the meaning of "$\approx$" later. Let $V_k = P\Lambda_k P^T$. Suppose $G = (P \circledast P)(\sum_{k=1}^{q} \alpha_k \Lambda_k \circledast \Lambda_k)(P \circledast P)^T = \sum_{k=1}^{q} \alpha_k V_k \circledast V_k$. Then G is an approximation to $\mathcal{H}^{-1}$. Thus, it is natural to consider the following preconditioner for M:

$$\widehat{M} \;=\; \sum_{k=1}^{q} \alpha_k \mathcal{A}(V_k \circledast V_k)\mathcal{A}^T. \tag{27}$$

The complexity for computing $\widehat{M}$ is at most q times that of the Schur complement matrix associated with the NT direction for a standard linear SDP. The cost of computing each term $\mathcal{A}(V_k \circledast V_k)\mathcal{A}^T$ is $2mn^3 + \frac{1}{2}m^2n^2$ if sparsity in $\mathcal{A}$ is ignored [33]. But we should emphasize that in (27), sparsity in $\mathcal{A}$ can fully be exploited using the techniques in [11], which is not the case for the matrix M in (25). Thus, the computational cost of $\widehat{M}$ is potentially much lower than that of M. It is precisely the difference between (27) and (25) in exploiting sparsity that motivated us to consider the preconditioner (27).

For the approximation problem (26), one can consider minimizing the 2-norm of the difference of the two diagonal matrices on both sides of the approximation. We focus on the vectors consisting of their diagonal elements and then consider the symmetric matrices obtained from these vectors as follows. The elements of each vector (of length $\bar{n}$) are extracted sequentially to fill in the upper-triangular part of an $n \times n$ matrix column-wise, and the lower-triangular part is then filled in to yield a symmetric matrix. For the right-hand-side matrix in (26), this process yields $K := 1./(ee^T + dd^T)$, the matrix whose entries are the reciprocals of the entries of $ee^T + dd^T$, where $d = \mathrm{diag}(D)$ and e is the vector of ones. For the left-hand-side matrix, the corresponding term is $\sum_{k=1}^{q} \alpha_k \lambda_k \lambda_k^T$ where $\lambda_k = \mathrm{diag}(\Lambda_k)$. Now, minimizing the 2-norm of the difference of the two diagonal matrices in (26) is equivalent solving the problem: $\min\{\max_{ij} |(K - \sum_{k=1}^{a} \alpha_k \lambda_k \lambda_k^T)_{ij}| \;:\; \lambda_k \in I\!R^n, \alpha_k \in I\!R, k = 1, \ldots, q\}$. Unfortunately, this problem cannot be solved easily. However, the variant that is based on minimizing the upper bound $\left\| K - \sum_{k=1}^{q} \alpha_k \lambda_k \lambda_k^T \right\|_2$ on its objective function can easily be solved. That is, we consider the following approximation problem for (26):

$$\min_{\alpha_k \in I\!R, \lambda_k \in I\!R^n, k=1,\ldots,q} \left\| K - \sum_{k=1}^{q} \alpha_k \lambda_k \lambda_k^T \right\|_2, \tag{28}$$

The matrix 2-norm is chosen because this problem admits an analytical solution given by

$$\alpha_k = \sigma_k, \qquad \lambda_k \;=\; u_k, \qquad k = 1, \ldots, q, \tag{29}$$

where σ_k is the kth largest eigenvalue (in absolute value) of the matrix K and u_k is the corresponding unit eigenvector vector.

Theorem 5.1 *For the preconditioner $\widehat{M}$ given in (27) with the α_k's and λ_k's given by (29), we have*

$$\left\|M - \widehat{M}\right\|_2 \;\leq\; \left\|\widetilde{\mathcal{A}}\right\|_2^2 |\sigma_{q+1}|,$$

where σ_{q+1} is the $q+1$-st largest eigenvalue of the matrix K. In case P and D were computed by part (a) of Lemma 5.1, $\left\|\widetilde{\mathcal{A}}\right\|_2^2 = \left\|\mathcal{A}(W \circledast W)\mathcal{A}^T\right\|_2$, while if part (b) was used, $\left\|\widetilde{\mathcal{A}}\right\|_2^2 = \left\|\mathcal{A}(U^{-1} \circledast U^{-1})\mathcal{A}^T\right\|_2$.

Proof. It is readily shown that

$$\begin{aligned}
\left\|M - \widehat{M}\right\|_2 &= \left\|\widetilde{\mathcal{A}}\left((\mathcal{I} + D \circledast D)^{-1} - \sum_{k=1}^{q} \sigma_k \mathrm{Diag}(u_k) \circledast \mathrm{Diag}(u_k)\right)\widetilde{\mathcal{A}}^T\right\|_2 \\
&\leq \left\|\widetilde{\mathcal{A}}\right\|_2^2 \left\|(\mathcal{I} + D \circledast D)^{-1} - \sum_{k=1}^{q} \sigma_k \mathrm{Diag}(u_k) \circledast \mathrm{Diag}(u_k)\right\|_2 \\
&\leq \left\|\widetilde{\mathcal{A}}\right\|_2^2 \left\|K - \sum_{k=1}^{q} \sigma_k u_k u_k^T\right\|_2 .
\end{aligned}$$

Since $\sum_{k=1}^{q} \sigma_k u_k u_k^T$ is the best rank q approximation of K, the second norm in the last line above is given by σ_{q+1}; see [12]. The last part follows since $\left\|\widetilde{\mathcal{A}}\right\|_2^2 = \left\|\widetilde{\mathcal{A}}\widetilde{\mathcal{A}}^T\right\|_2 = \left\|\mathcal{A}(PP^T \circledast PP^T)\mathcal{A}^T\right\|_2$, using the form of PP^T given at the beginning of the subsection.

Remark 5.2 *(a) While the Schur complement matrix (22) is positive definite, the matrix $\widehat{M}$ may not be positive definite.*

(b) For the numerical experiments in Section 7, we take q in Theorem 5.1 as follows:

$$q \;=\; \min\left\{15, \min\{k \,:\, |\sigma_{k+1}| \leq 10^{-8}|\sigma_1|\}\right\}.$$

(c) Note that to solve (28), the easiest (though not necessarily the cheapest) mean is to compute the full eigenvalue decomposition of the $n \times n$ symmetric matrix K. If one is interested only in approximating K by the sum of a few rank-one matrices, then one can use variants of the Lanczos method to compute a partial eigenvalue decomposition

of K.

(d) The construction of $\widehat{M}$ can be made more efficient than computing each constituent term $\mathcal{A}(V_k \circledast V_k)\mathcal{A}^T$ separately. For example, the inner product with A_i in $\widehat{M}_{ij} = A_i \bullet (\sum_{k=1}^{q} \alpha_k V_k A_j V_k)$ need only be done once instead of q times. However, in the interest of keeping our implementation simple, we did not optimize the computational efficiency of $\widehat{M}$ in Section 7.

5.2 Preconditioners for M, part II

For the special convex quadratic SDP with $Q = U \circledast U$, the middle operator in the Schur complement matrix M involves inverting an operator of the form $H = G_1 \circledast G_1 + G_2 \circledast G_2$, where G_1, G_2 are given symmetric positive definite matrices. Given that it is easy to invert an operator of the form $V \circledast V$, it is natural for us to consider approximating a sum of symmetrized Kronecker products by a single term. Recall that a symmetrized Kronecker product $U \circledast U$ is an operator on $\mathcal{S}^n$, but it has a matrix representation using the operation **svec**. Let $\mathrm{Mat}(U \circledast U)$ in $\mathcal{S}^{\bar{n}}$ be defined by $\mathrm{Mat}(U \circledast U)\mathbf{svec}(V) = \mathbf{svec}(U \circledast U(V))$. Then our problem is

$$\min_{V \in \mathcal{S}^n} \left\| \sum_{j=1}^{2} \mathrm{Mat}(G_j \circledast G_j) - \mathrm{Mat}(V \circledast V) \right\|_F^2. \tag{30}$$

The above problem can be replaced by a simpler one which we will derive next. By noting that a symmetrized Kronecker product matrix $\mathrm{Mat}(G \circledast G)$ is related to a standard Kronecker product by the formula $\mathrm{Mat}(G \circledast G) = \Pi^T (G \otimes G)\Pi$, where the constant matrix $\Pi \in I\!R^{n^2 \times \bar{n}}$ has orthonormal columns (see the Appendix of [30]), we have

$$\begin{aligned} \left\| \sum_{j=1}^{2} \mathrm{Mat}(G_j \circledast G_j) - \mathrm{Mat}(V \circledast V) \right\|_F &= \left\| \Pi^T \left(\sum_{j=1}^{2} G_j \otimes G_j - V \otimes V \right) \Pi \right\|_F \\ \leq \quad \|\Pi\|_2^2 \left\| \sum_{j=1}^{2} G_j \otimes G_j - V \otimes V \right\|_F. & \end{aligned}$$

Note that $\|\Pi\|_2 = 1$. Thus instead of (30), we can consider solving the following problem:

$$\min_{V \in \mathcal{S}^n} \left\| \sum_{j=1}^{2} G_j \otimes G_j - V \otimes V \right\|_F^2. \tag{31}$$

The problem (31) is a special case of a more general problem studied in [20] for approximating a sum of Kronecker products by a single Kronecker product, namely,

$$\min_{U,V \in I\!R^{n \times n}} \left\| \sum_{j=1}^{q} G_j \otimes K_j - U \otimes V \right\|_F^2, \tag{32}$$

where G_j, K_j are $n \times n$ matrices (not necessarily symmetric positive definite). It is shown in [20] that the optimal solution of (32) takes the form $U = \sum_{j=1}^{q} \alpha_j G_j$ and $V = \sum_{j=1}^{q} \beta_j K_j$, with the coefficients α_j, β_j being the optimal solution of the following equivalent non-convex minimization problem:

$$\min_{\alpha,\beta \in I\!R^q} f(\alpha, \beta) \ := \ \mathrm{Tr}(\boldsymbol{GK}) - 2\alpha^T \boldsymbol{GK}\beta + (\alpha^T \boldsymbol{G}\alpha)(\beta^T \boldsymbol{K}\beta), \tag{33}$$

where $\boldsymbol{G}, \boldsymbol{K} \in \mathcal{S}_+^q$ are defined by $\boldsymbol{G}_{ij} = \mathrm{Tr}(G_i^T G_j)$, $\boldsymbol{K}_{ij} = \mathrm{Tr}(K_i^T K_j)$. For simplicity, we assume that $\{G_1, \ldots, G_q\}$ and $\{K_1, \ldots, K_q\}$ are linearly independent sets. Under this assumption, $\boldsymbol{G}$ and $\boldsymbol{K}$ are positive definite.

In [20], the optimal coefficients α and β in (32) are found by using an optimization software such as multilevel coordinate search. Here we show that the optimal solution can instead be found analytically.

Proposition 5.1 *The optimal objective value of (32) is given by*

$$f(\alpha^*, \beta^*) = Tr(\boldsymbol{GK}) - \lambda_{\max}(\boldsymbol{GK}),$$

where $\lambda_{\max}(\boldsymbol{GK})$ is the largest eigenvalue of $\boldsymbol{GK}$; (α^*, β^*) is a corresponding left and right eigenvector pair of $\boldsymbol{GK}$. Let $R = \sum_{j=1}^{q} G_j \otimes K_j$. We have the following inequality for the relative error:

$$\frac{f(\alpha^*, \beta^*)}{\|R\|_F} \ = \ 1 - \frac{\lambda_{\max}(\boldsymbol{GK})}{Tr(\boldsymbol{GK})} \ \leq \ 1 - \frac{1}{q}.$$

Proof. We observe that f is coercive (so it has a minimizer) and that

$$\nabla f = 2 \begin{bmatrix} (\beta^T \boldsymbol{K}\beta)\boldsymbol{G}\alpha - \boldsymbol{GK}\beta \\ (\alpha^T \boldsymbol{G}\alpha)\boldsymbol{K}\beta - \boldsymbol{KG}\alpha \end{bmatrix}.$$

Thus the critical points of $f(\alpha, \beta)$ are given by the solutions of the following equations:

$$\begin{aligned} \boldsymbol{K}\beta &= (\beta^T \boldsymbol{K}\beta)\alpha \\ \boldsymbol{G}\alpha &= (\alpha^T \boldsymbol{G}\alpha)\beta, \end{aligned}$$

after making use of the fact that $\boldsymbol{G}$ and $\boldsymbol{K}$ are nonsingular. Now it is easy to see from the above equations that

$$\boldsymbol{KG}\alpha = (\alpha^T \boldsymbol{G}\alpha)(\beta^T \boldsymbol{K}\beta)\alpha \tag{34}$$

$$\boldsymbol{GK}\beta = (\alpha^T \boldsymbol{G}\alpha)(\beta^T \boldsymbol{K}\beta)\beta. \tag{35}$$

This shows that the critical points $(\bar{\alpha}, \bar{\beta})$ are the left and right eigenvector pairs of $\boldsymbol{GK}$. The corresponding objective value can be shown to be given by

$$f(\bar{\alpha}, \bar{\beta}) \ = \ \mathrm{Tr}(\boldsymbol{GK}) - (\bar{\alpha}^T \boldsymbol{G}\bar{\alpha})(\bar{\beta}^T \boldsymbol{K}\bar{\beta}),$$

where the term $(\bar{\alpha}^T \boldsymbol{G}\bar{\alpha})(\bar{\beta}^T \boldsymbol{K}\bar{\beta})$ is an eigenvalue of $\boldsymbol{GK}$. Since the eigenvalues of $\boldsymbol{GK}$ are all real and non-negative, to minimize the objective function in (33), it is clear

that the left and right eigenvector pair of $\boldsymbol{GK}$ corresponding to the largest eigenvalue $\lambda_{\max}(\boldsymbol{GK})$ must be chosen.

To prove the second statement, note that $\|R\|_F = \mathrm{Tr}(\boldsymbol{GK}) = \left\|\boldsymbol{G}^{1/2}\boldsymbol{K}^{1/2}\right\|_F^2$. We also have $\lambda_{\max}(\boldsymbol{GK}) = \left\|\boldsymbol{G}^{1/2}\boldsymbol{K}^{1/2}\right\|_2^2$. Thus by a standard inequality between the matrix 2-norm and Frobenius norm, the required inequality is established. □

Although it seems plausible that the optimal solution of (32) could very well also be the optimal solution of (30), we note that simple examples demonstrate that this is not necessarily true.

Proposition 5.2 *Suppose G_j and K_j are symmetric positive definite for $j = 1, \ldots, q$. Then the optimal solution pair U^* and V^* in (32) are also symmetric positive definite.*

Proof. Since $G_j, K_j,\ j = 1, \ldots, q$ are positive definite, $\boldsymbol{G}$ and $\boldsymbol{K}$ are non-negative matrices. This implies that $\boldsymbol{GK}$ is a non-negative matrix. By the Perron-Frobenius Theorem [17, p. 500], $\lambda_{\max}(\boldsymbol{GK})$ is algebraically simple and there exists a corresponding left and right eigenvector pair α^*, β^* for which the vectors are non-negative. In this case, the matrices $U^* = \sum_{j=1}^q \alpha_j^* G_j$, $V^* = \sum_{j=1}^q \beta_j K_j$ are symmetric positive definite. □

For the problem (31), the optimal solution is given by $V^* = \sum_{j=1}^2 \alpha_j^* G_j$, with α^* being a unit eigenvector corresponding to the largest eigenvalue of $\boldsymbol{G}^2$. Note that by Proposition 5.2, V^* is symmetric positive definite. Since $V^* \circledast V^*$ is an approximation of $\mathcal{H}$, we can naturally precondition the Schur complement matrix in (22) using the following matrix:

$$\widehat{M} \;=\; \mathcal{A}[(V^*)^{-1} \circledast (V^*)^{-1}]\mathcal{A}^T. \tag{36}$$

While the preconditioner we have constructed in (27) may not be positive definite, the above preconditioner is guaranteed to be positive definite.

6 Linear semidefinite programming with a simple upper bound

Consider the following semidefinite program with a simple upper bound:

$$\begin{aligned} &\min_X \; C \bullet X \\ &\mathcal{A}(X) \;=\; b, \quad 0 \;\preceq\; X \;\preceq\; U, \end{aligned} \tag{37}$$

where $U \succeq 0$ is a given matrix. An example of (37) comes from minimizing the sum of the largest q eigenvalues of an affine function of symmetric matrices, namely,

$$\min \left\{ \sum_{k=1}^{q} \lambda_k(\mathcal{A}^T y - C) \;:\; y \in I\!R^m \right\}. \tag{38}$$

In [1, (4.7)], it is shown that (38) is equivalent to a linear SDP with a simple upper bound:

$$\min \{C \bullet X \,:\, \mathcal{A}(X) = 0,\ I \bullet X = q,\ 0 \preceq X \preceq I\}. \tag{39}$$

To derive the KKT conditions for (37), it is convenient to express (37) in the following standard form:

$$\min_X \left\{ C \bullet X \,:\, \begin{bmatrix} \mathcal{A} \\ -I \end{bmatrix} X + \begin{bmatrix} 0 \\ -I \end{bmatrix} V \;=\; \begin{bmatrix} b \\ -U \end{bmatrix}, \quad X, V \succeq 0 \right\}. \tag{40}$$

We see that converting the problem (37) to the standard form introduces $\bar{n}$ extra equality constraints in the primal problem. Thus it is extremely expensive to solve (37) by treating it as a standard linear SDP.

The dual problem corresponding to (40) is given by

$$\max \left\{ b^T y - U \bullet Z \,:\, \mathcal{A}^T y - Z + S \;=\; C, \quad S, Z \succeq 0 \right\}. \tag{41}$$

The perturbed KKT conditions for (40) and (41) are given by

$$\begin{aligned} \mathcal{A}X &= b \\ X + V &= U \\ \mathcal{A}^T y - Z + S &= C \\ XS &= \nu I \\ VZ &= \nu I. \end{aligned} \tag{42}$$

The symmetrized Newton equation corresponding to the above system is given by

$$\begin{aligned} \mathcal{A}\Delta X &= r^p := b - \mathcal{A}X \\ \Delta X + \Delta V &= R^u := U - X - V \\ \mathcal{A}^T \Delta y - \Delta Z + \Delta S &= R^d := C - \mathcal{A}^T y + Z - S \\ \mathcal{E}_1 \Delta X + \mathcal{F}_1 \Delta S &= R_1^c := \sigma\mu I - H_1(XS) \\ \mathcal{E}_2 \Delta V + \mathcal{F}_2 \Delta Z &= R_2^c := \sigma\mu I - H_2(VZ), \end{aligned} \tag{43}$$

where $\mathcal{E}_1, \mathcal{F}_1$ and $\mathcal{E}_2, \mathcal{F}_2$ are linear operators in $\mathcal{S}^n$ that depend on the symmetrization schemes chosen for X, S and V, Z, respectively. It is readily shown that the search direction $(\Delta X, \Delta V, \Delta y, \Delta S, \Delta Z)$ corresponding to the Newton equation (43) can be computed by solving the following linear system:

$$\begin{bmatrix} -(\mathcal{F}_1^{-1}\mathcal{E}_1 + \mathcal{F}_2^{-1}\mathcal{E}_2) & \mathcal{A}^T \\ \mathcal{A} & 0 \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta y \end{bmatrix} = \begin{bmatrix} h \\ r^p \end{bmatrix}$$

where

$$h \;=\; R^d - \mathcal{F}_1^{-1}R_1^c + \mathcal{F}_2^{-1}R_2^c - \mathcal{F}_2^{-1}\mathcal{E}_2 R^u.$$

By eliminating ΔX from the augmented equation above, we get the following Schur complement equation:

$$\mathcal{A}\left(\mathcal{F}_1^{-1}\mathcal{E}_1 + \mathcal{F}_2^{-1}\mathcal{E}_2\right)^{-1}\mathcal{A}^T\Delta y \;=\; r^p + \mathcal{A}\left(\mathcal{F}_1^{-1}\mathcal{E}_1 + \mathcal{F}_2^{-1}\mathcal{E}_2\right)^{-1} h. \tag{44}$$

The operator $\mathcal{F}_1^{-1}\mathcal{E}_1 + \mathcal{F}_2^{-1}\mathcal{E}_2$ in (44) is costly to invert in general. Hence constructing the Schur complement matrix in (44) can be very expensive. The reader may recall that in contrast, for the case of linear programming, having an upper bound vector does not add any extra computational cost because the middle matrix in (44) is the sum of two diagonal matrices.

The middle matrix in (44) has exactly the same structure as that in (20) if the NT scalings are chosen. In this case, $\mathcal{F}_1^{-1}\mathcal{E}_1 + \mathcal{F}_2^{-1}\mathcal{E}_2 = W_1^{-1} \circledast W_1^{-1} + W_2^{-1} \circledast W_2^{-1}$. Thus the Schur complement matrix in (44) can be reduced to exactly the same form as the matrix M in (22), and the preconditioners proposed in the last section can be used when solving (44) by an iterative solver.

The problem (37) is an example of problems where the decision variables are constrained to lie at the intersection of two cones. For such problems, the Schur complement equation arising from an interior-point iteration will have the form (44). Andersen studies such structures for the intersection of a linear and a second-order cone [5]. Another example appears in [14] where the authors study a problem very similar to (37). This problem comes from a robust optimization formulation of a convex quadratic programming problem and the upper bound is expressed using componentwise inequalities rather than the semidefinite inequality. In this case, the matrix $\mathcal{F}_2^{-1}\mathcal{E}_2$ is diagonal and the situation is similar to the QSDP problem with a diagonal $\mathcal{Q}$ operator.

7 Numerical experiments

To evaluate the performance of our interior-point algorithm, we consider the following classes of test problems:

E1. Quadratic SDP arising from the nearest correlation matrix matrix problem where $\mathcal{Q}(X) = X$. We use the linear map $\mathcal{A}(X) = \mathrm{diag}(X)$, and b is the vector of all ones; see (24). We generate the matrix $-C$ in the same way as Higham did in [16, p.340]. It is generated from the MATLAB function `gallery('randcorr',...)` with a random symmetric perturbation of Frobenius norm 10^{-4} added.

E2. Same as E1 but the matrix C is generated as follows: `T = 2*rand(n)-1; C = -0.5*(T+T');` Such a matrix is considered in the numerical experiments in [23].

E3. Same as E1 but the matrix C is generated as in [23] as follows: `T = [ones(n/2), zeros(n/2); zeros(n/2), eye(n/2)]; C = - T - 1e4*diag(2*rand(n,1)-1);`

E4. Same as E1 but $\mathcal{Q}(X) = UXU$, with $U \succ 0$ generated randomly as follows: `[Q,R] = qr(randn(n)); beta = 10^(-4/(n-1)); U = Q*diag(beta.^[0:n-1])*Q'`. The matrix C in E1 is replaced by UCU.

E5. Same as E2 but $\mathcal{Q}(X) = UXU$, with U generated in the same way as in E4. The matrix C in E2 is replaced by UCU.

E6. Same as E3 but $\mathcal{Q}(X) = UXU$, with U generated in the same way as in E4. The matrix C in E3 is replaced by UCU.

E7. Linear SDPs with simple upper bounds (39) arising from (38). Here we take $q = 5$, the matrix C is generated randomly in MATLAB as follows: `T = randn(n); T = 0.5*(T+T'); C = T + norm(T,2)*I`; the linear map $\mathcal{A}$ is chosen to be $\mathcal{A}(X) = [A_1 \bullet X, \ldots, A_{n-1} \bullet X]$, where $A_k = e_k e_{k+1}^T + e_{k+1} e_k^T$ for $k = 1, \ldots, n-1$.

We use 4 variants of Algorithm IP-QSDP to solve each test problem, namely,

A1. Algorithm IP-QSDP with search direction computed via (22) by a direct solver;

A2. Algorithm IP-QSDP with search direction computed via (22) by PSQMR with no preconditioning.

A3. Algorithm IP-QSDP with search direction computed via (22) by PSQMR with a hybrid preconditioner chosen as follows: it is taken to be the preconditioner (27) if it is positive definite; otherwise, it is taken to be the preconditioner (36).

A4. Algorithm IP-QSDP with search direction computed via (22) by PSQMR with preconditioner (36).

We implemented the algorithms in MATLAB (version 7.0) and the experiments were conducted on a Pentium IV 3.0GHz PC with 2GB of RAM. We stopped the algorithms when the accuracy measure ϕ in (8) was less than 10^{-8}, or when the algorithms did not improve both the duality gap and infeasibilities. We set the maximum number of PSQMR steps allowed per solve to n.

The initial iterate for all the algorithms was taken to be the default starting point in [31]. For the problem sets E1–E6, it is given by

$$X^0 \;=\; \frac{n}{\sqrt{2}} I, \quad y^0 = 0, \quad S^0 \;=\; \sqrt{n} I.$$

The performance results of our algorithms are given in Table 1. The columns corresponding to "`it`" give the total number of interior-point iterations required to solve each problem, whereas the columns "`psqmr`" give the average number of PSQMR steps required to solve each of the two linear systems (22) during the computation of the predictor and corrector directions at each interior-point iteration.

It is worth making some observations we may derive from the performance table.

1. Solving (13) via a direct solver is extremely expensive. For the problem in E1-1600, it is at least 20 times more expensive than the algorithms using iterative solvers to solve (13).
2. Based on the stopping criterion we proposed in Section 4, the algorithms that use an iterative method to solve (13) took about the same number of interior-point

iterations to converge compared to the algorithm using a direct method. This indicates that the inexact search directions are computed to sufficient accuracy, and thus the residual errors do not degrade the outer iterations.

3. The test examples considered in [16] and [23] for the unweighted nearest correlation matrix problem are easy problems that can be solved by an iterative solver even without preconditioning. As we can observe from the test problems in E1–E3, the SQMR method takes an average of 2 to 4 steps to solve the Schur complement equation (13) without any preconditioning. This indicates that the coefficient matrix M is very well-conditioned throughout the entire course of interior-point iterations. With such a well-conditioned system, the preconditioners proposed in (27) and (36) cannot offer any saving in the computation time because of the overheads involved in their construction. For these easy problems, the condition number of M stays bounded even when the duality gap decreases to zero. This possibility is consistent with Corollary 4.1.

4. The conditioning of the matrix M becomes slightly worse for the weighted nearest correlation matrix problems considered in E4 and E5. This can be seen from the slight increase in the average number of SQMR steps required to solve (13). The preconditioned systems generally take fewer steps to converge, but the reduction in the number of PSQMR steps is not enough to offset the preconditioning overhead.

5. The test problems in E6 and E7 truly demonstrate the effectiveness of the preconditioners (27) and (36). For the test problem in E6-2000, the SQMR method takes an average of 80.4 steps to solve (13) whereas the corresponding numbers for the system preconditioned by (27) and (36) are 1.0 and 26.8, respectively.

6. The number of interior-point iterations required by our proposed primal-dual interior-point method grows very modestly with the problem dimension n. In all the test problems, the number of iterations required is less than 30. In contrast, for problems similar to those in E3 considered in [23], the number of iterations required to solve the Lagrangian dual problems exceeded 300.

7. If we compare the CPU time taken by Algorithm A2 to solve the problem E1-2000 with that in [23], each interior-point iteration of our algorithm is about 7.7 times slower (after taking into account the difference in the speed of the machines used). There are several reasons to explain such a big discrepancy. First, the computation in [23] is done in Fortran combined with the LAPACK library, whereas our computation is done in MATLAB. It is a well-known fact that a speed-up by a factor of 5 or 10 is not uncommon when a MATLAB computation is transplanted to Fortran. Second, each iteration in [23] is a quasi-Newton iteration which involves only gradient evaluations, whereas each iteration of our algorithm is a Newton iteration. Third, as we mentioned in Remark 5.1, each iteration of our algorithm requires 2 eigenvalue decompositions, whereas each iteration of the algorithm in [23] requires only one. This additional eigenvalue decomposition accounts for 20% of the time spent per iteration.

8. It is clear from (27) and (36) that the overhead incurred in constructing the first

preconditioner will be more than that for the second. This is reflected in the CPU times in Table 1. For example, Algorithm A4 takes less time to solve E5-2000 (similarly for E7-2000) than Algorithm A3, even though the former required more PSQMR steps per solve. Generally, both preconditioners are quite effective on all the problem classes considered. However, the first preconditioner is much more effective than the second for the more difficult problem class E6.

9. If we compare the CPU times taken to solve the three larger problems within each problem class, we see that the growth rates are at most $n^{2.65}$, $n^{1.94}$, and $n^{2.18}$ (the exponents are crudely estimated from regressions with $\log_{10}(n)$ against n), for Algorithms A2, A3, and A4, respectively. These exponents are much smaller than the exponent of between 3 to 4 that is expected from the direct-solver based Algorithm A1.

8 Conclusion and future research

We considered a primal-dual path-following Mehrotra-type predictor-corrector method for solving convex quadratic SDP problems. For the special case when the derivative of the quadratic term $\mathcal{Q}(X)$ has the form UXU, we suggested computing the search direction at each iteration based on the Schur complement equation, using the PSQMR iterative solver with one of two appropriately constructed preconditioners. Numerical experiments on a variety of QSDPs with matrices of dimension up to 2000 showed that our methods are quite efficient and robust. We also extended our methods to solving linear SDP problems with upper bound constraints on the primal matrix variable.

We observed from our computations that the matrix $\widehat{M}$ in (27) is usually an extremely good approximation to M for the value q selected based on Remark 5.2(b). An interesting idea that is worth exploring is to use the direction Δy computed from $\widehat{M}\Delta y = h$ for each IPM iteration, instead of the direction computed from (13) itself. In that case, Δy can be computed from the approximate system straightforwardly by a direct solver.

As mentioned in the introduction, our ultimate goal is to solve problems with a general positive semidefinite $\mathcal{Q}$, in which case the search direction at each IPM iteration has to be computed from the augmented equation (11). Our hope is to solve this equation efficiently via an iterative solver with appropriately constructed preconditioners.

Through the years and in the current paper, our interest in the robust and efficient implementation of interior-point methods for various classes of conic optimization problems and our continued work on SDPT3 were motivated, and in many cases inspired, by the friendly but strong competition we received from Jos Sturm. His hugely successful software SeDuMi [29] continually raised the bar for us and other developers of conic optimization software. We dedicate this work to his memory.

Table 1: Performance of the Algorithms A1–A4 on the problem sets E1–E7.

		A1			A2				A3				A4			
	n	it	ϕ	Time (s)	it	ϕ	Time (s)	psqmr	it	ϕ	Time (s)	psqmr	it	ϕ	Time (s)	psqmr
E1	200	14	7.1e-09	36.9	14	6.9e-09	8.6	3.6	14	7.1e-09	11.1	1.0	14	7.2e-09	9.7	3.5
	400	15	3.4e-09	327.8	15	3.4e-09	44.7	3.1	15	3.4e-09	54.6	1.0	15	3.3e-09	46.0	3.0
	800	15	7.8e-09	3605.4	15	7.8e-09	291.2	2.8	15	7.8e-09	345.2	1.0	15	7.7e-09	300.2	2.8
	1600	15	9.4e-09	46546.3	15	9.5e-09	1869.9	2.6	15	9.4e-09	2404.7	1.0	15	9.5e-09	1889.6	2.6
	2000				16	2.8e-09	4584.9	2.6	16	2.7e-09	4872.8	1.0	16	2.7e-09	4753.2	2.6
E2	200	10	4.9e-09	25.7	10	4.6e-09	5.8	2.7	10	4.9e-09	7.3	1.8	10	5.1e-09	5.8	2.1
	400	11	7.8e-09	235.1	11	7.6e-09	31.4	2.6	11	7.8e-09	40.0	1.7	11	8.0e-09	31.5	2.1
	800	11	4.7e-09	2578.8	11	5.1e-09	192.7	2.3	11	4.7e-09	259.2	1.6	11	4.3e-09	195.1	1.9
	1600				12	2.6e-09	1411.5	2.2	12	2.2e-09	1961.3	1.7	12	2.0e-09	1424.0	1.8
	2000				11	5.9e-09	3073.0	2.0	11	5.4e-09	3396.8	1.4	11	5.6e-09	3147.1	1.7
E3	200	13	5.9e-10	28.9	13	7.9e-10	7.1	1.8	13	5.9e-10	8.4	1.1	13	1.0e-09	7.5	1.6
	400	13	3.4e-10	293.6	13	4.9e-09	41.0	1.7	13	3.4e-10	48.5	1.1	13	3.6e-10	42.7	1.7
	800	14	2.9e-10	3516.8	14	1.6e-09	291.7	1.7	14	1.1e-09	368.4	1.1	14	3.0e-10	300.0	1.7
	1600				14	6.8e-09	1932.0	1.7	14	6.7e-09	2302.3	1.1	14	6.7e-09	1974.7	1.7
	2000				15	2.3e-10	3862.6	1.7	15	2.2e-10	4701.5	1.2	15	2.2e-10	4053.4	1.7
E4	200	11	3.7e-09	29.1	11	3.0e-09	8.9	11.3	11	3.7e-09	7.6	1.0	11	6.1e-09	7.5	6.7
	400	12	5.0e-09	281.1	12	7.6e-09	50.8	9.0	12	5.0e-09	51.7	2.4	12	6.1e-09	46.6	6.4
	800	15	3.6e-09	3922.4	15	4.3e-09	383.8	7.2	15	3.6e-09	431.2	3.4	15	9.1e-09	372.4	5.9
	1600				15	5.1e-09	2415.9	5.3	14	5.8e-09	2741.5	1.4	15	3.5e-09	2470.5	5.1
	2000				16	2.2e-09	4872.2	5.3	15	1.3e-09	5917.1	3.5	16	6.6e-10	4916.0	5.0
E5	200	13	3.6e-09	35.3	13	4.4e-09	11.7	13.1	13	3.6e-09	9.8	1.5	13	3.7e-09	8.6	5.3
	400	14	3.7e-10	328.0	14	3.9e-10	61.6	9.8	14	3.7e-10	61.3	2.5	14	3.7e-10	53.0	5.6
	800	14	7.4e-09	3614.8	14	7.0e-09	358.1	7.3	14	7.4e-09	394.8	3.1	14	7.0e-09	326.0	4.7
	1600				15	4.9e-09	2460.2	5.6	14	9.8e-09	2702.5	2.8	14	9.1e-09	2383.5	4.2
	2000				14	9.6e-09	4180.8	5.0	14	4.6e-09	5377.4	2.9	14	6.0e-09	4156.7	4.3
E6	200	21	1.6e-09	57.0	21	9.6e-09	32.1	32.9	21	1.6e-09	14.4	1.3	21	1.8e-09	21.4	16.7
	400	22	1.1e-09	510.8	24	3.3e-09	250.5	44.9	22	1.1e-09	81.2	1.0	22	4.6e-09	128.4	18.5
	800	23	4.4e-09	5888.7	26	1.5e-09	2049.9	57.0	23	4.4e-09	551.0	1.0	24	1.5e-09	947.6	20.8

Table 1: Performance of the Algorithms A1–A4 on the problem sets E1–E7.

		A1			A2				A3				A4			
	n	it	ϕ	Time (s)	it	ϕ	Time (s)	psqmr	it	ϕ	Time (s)	psqmr	it	ϕ	Time (s)	psqmr
	1600				29	1.5e-09	21704.4	81.6	26	1.4e-09	4422.5	1.0	27	1.5e-09	9293.8	30.6
	2000				30	4.2e-09	41537.6	80.4	27	7.7e-10	9014.6	1.0	28	9.3e-09	16608.7	26.8
E7	200	15	5.5e-08	58.9	17	2.9e-08	93.3	90.7	15	5.2e-08	19.9	1.1	16	3.0e-08	82.5	85.2
	400	16	3.2e-08	373.7	17	1.1e-08	563.1	149.6	15	4.3e-08	116.2	1.5	16	2.6e-08	545.6	107.9
	800	15	6.6e-08	3955.8	15	1.1e-07	4312.7	236.4	15	1.4e-07	508.8	1.3	15	5.0e-08	707.2	18.9
	1600	17	2.7e-07	56457.0	17	3.3e-07	62418.6	440.9	17	1.3e-07	5818.5	1.3	17	7.7e-08	3917.3	6.4
	2000				18	1.4e-06	136231.9	520.2	17	1.2e-07	9128.1	1.1	18	8.6e-08	7619.3	6.0

References

[1] F. Alizadeh, *Interior point methods in semidefinite programming with applications to combinatorial optimization*, SIAM J. Optimization, 5 (1995), pp. 13–51.

[2] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, *Complementarity and nondegeneracy in semidefinite programming*, Math. Programming, 77 (1997), pp. 111–128.

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

[4] A. Y. Alfakih, A. Khandani, and H. Wolkowicz, *Solving Euclidean distance matrix completion problems via semidefinite programming*, Computational Optimization and Applications, 12 (1999), pp. 13–30.

[5] E. D. Andersen, *On exploiting structure induced when modelling an intersection of cones in conic optimization*, Research Report, July 2004. available from `http://www.optimization-online.org`.

[6] M. F. Anjos, N. J. Higham, P. L. Takouda, and H. Wolkowicz, *A semidefinite programming approach for the nearest correlation matrix problem*, Research Report, Department of Combinatorics and Optimization, University of Waterloo, 2003.

[7] P. Apkarian, D. Noll, J.-P. Thevenet, and H. D. Tuan, *A spectral quadratic-SDP method with applications to fixed-order H_2 and H_∞ synthesis*, Research Report, Universite Paul Sabaitier, Toulouse, France, 2004.

[8] S. Boyd and L. Xiao, *Least-squares covariance matrix adjustment*, preprint, Department of Electrical Engineering, Stanford University.

[9] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[10] R. W. Freund and N. M. Nachtigal, *A new Krylov-subspace method for symmetric indefinite linear systems*, Proceedings of the 14th IMACS World Congress on Computational and Applied Mathematics, Atlanta, USA, W.F. Ames ed., July 1994, pp. 1253–1256.

[11] K. Fujisawa, M. Kojima, and K. Nakata, *Exploiting sparsity in primal-dual interior-point method for semidefinite programming*, Mathematical Programming, 79 (1997), pp. 235–253.

[12] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989.

[13] M. Halicka, E. de Klerk, and C. Roos, *Limiting behaviour of the central path in semidefinite optimization*, Optimization Methods and Software, 20 (2005), pp. 99–113.

[14] B. V. Halldórsson and R. H. Tütüncü, *An interior-point method for a class of saddle point problems*, Journal of Optimization Theory and Applications, 116 (2003) pp. 559–590.

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

[16] N. J. Higham, *Computing the nearest correlation matrix — a problem from finance*, IMA J. Numerical Analysis, 22 (2002), pp. 329–343.

[17] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.

[18] C. Kanzow, C. Nagel, H. Kato, and M. Fukushima, *Successive linearization methods for nonlinear semidefinite programming*, preprint.

[19] M. Kojima, S. Shindoh, and S. Hara, *Interior-point methods for the monotone linear complementarity problem in symmetric matrices*, SIAM J. Optimization, 7 (1997), pp. 86–125.

[20] A. N. Langville and W. J. Stewart, *A Kronecker product approximate preconditioner for SANs*, Numerical Linear Algebra with Applications, 11 (2004), pp. 723–752.

[21] Z. Lu, R. D. C. Monteiro, and J. W. O'Neal, *An iterative solver-based infeasible primal-dual path-following algorithm for convex QP*, preprint, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta.

[22] Z.-Q. Luo, J. F. Sturm, and S. Zhang, *Superlinear convergence of a symmetric primal-dual path following algorithm for semidefinite programming*, SIAM J. on Optimization, 8 (1998), pp. 59–81.

[23] J. Malick, *A dual approach to semidefinite least-squares problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 272–284.

[24] R. D. C. Monteiro, *Primal-dual path-following algorithms for semidefinite programming*, SIAM J. Optimization, 7 (1997), pp. 663–678.

[25] R. D. C. Monteiro and Y. Zhang, *A unified analysis for a class of path-following primal-dual interior-point algorithms for semidefinite programming*, Math. Prog., 81 (1998), pp. 281–299.

[26] J. W. Nie and Y. X. Yuan, *A predictor-corrector algorithm for QSDP combining Dikin-type and Newton centering steps*, Annals of Operations Research, 103 (2001), pp. 115–133.

[27] H. Qi and D. Sun, *Quadratic convergence and numerical experiments of Newton's method for computing the nearest correlation matrix* , preprint, Department of Mathematics, National University of Singapore.

[28] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.

[29] J. F. Sturm, *Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones*, Optimization Methods and Software, 11 & 12 (1999), pp. 625–653.

[30] M. J. Todd, K. C. Toh, and R. H. Tütüncü, *On the Nesterov-Todd direction in semidefinite programming*, SIAM J. Optimization, 8 (1998), pp. 769–796.

[31] M. J. Todd, K. C. Toh, and R. H. Tütüncü, *Solving semidefinite-quadratic-linear programs using SDPT3*, Math. Prog. 95 (2003), pp. 189–217.

[32] K. C. Toh, K. K. Phoon, and S. H. Chan, *Block preconditioners for symmetric indefinite linear systems*, Int. J. Numerical Methods in Engineering, 60 (2004), pp. 1361–1381.

[33] K. C. Toh, *Some new search directions for primal-dual interior point methods in semidefinite programming*, SIAM J. Optimization, 11 (2000), pp. 223–242.