Open access • Journal Article • DOI:10.1137/S105262349630060X

On the Nesterov--Todd Direction in Semidefinite Programming — Source link

Michael J. Todd, Kim-Chuan Toh, Reha Tütüncü

Institutions: Cornell University, National University of Singapore

Published on: 01 Mar 1998 - Siam Journal on Optimization (Society for Industrial and Applied Mathematics)

Topics: Semidefinite programming, Interior point method, Semidefinite embedding, Second-order cone programming and Quadratically constrained quadratic program

ON THE NESTEROV-TODD DIRECTION IN SEMIDEFINITE PROGRAMMING *

M. J. TODD †, K. C. TOH ‡, AND R. H. TÜTÜNCÜ §

Abstract. We study different choices of search direction for primal-dual interior-point methods for semidefinite programming problems. One particular choice we consider comes from a specialization of a class of algorithms developed by Nesterov and Todd for certain convex programming problems. We discuss how the search directions for the Nesterov-Todd (NT) method can be computed efficiently and demonstrate how they can be viewed as Newton directions. This last observation also leads to convenient computation of accelerated steps, using the Mehrotra predictor-corrector approach, in the NT framework. We also provide an analytical and numerical comparison of several methods using different search directions, and suggest that the method using the NT direction is more robust than alternative methods.

Key words. semidefinite programming, Newton direction, predictor-corrector interior-point method.

AMS subject classifications. 90C05

1. Introduction. This paper studies interior-point methods for semidefinite programming (SDP) problems based on the search direction described by Nesterov and Todd [17, 18]. For simplicity, we call this direction the Nesterov-Todd (NT) direction. We consider the SDP given in the following standard form:

$$(SDP) \qquad \begin{array}{lll} \min_X & C \bullet X & \\ & A_i \bullet X = b_i, & i = 1, \ldots, m \\ & X \succeq 0, & \end{array} \tag{1}$$

where each $A_i \in S\mathbb{R}^{n\times n}$, $b \in \mathbb{R}^m$, $C \in S\mathbb{R}^{n\times n}$ are given, and $X \in S\mathbb{R}^{n\times n}$. Here $S\mathbb{R}^{n\times n}$ denotes the space of $n \times n$ symmetric matrices, and $X \succeq 0$ indicates that X is positive semidefinite. We assume that the set $\{A_i\}$ is linearly independent. The dual problem associated with (SDP) is:

$$(SDD) \qquad \begin{array}{ll} \max\limits_{y,S} & b^T y \\ & \sum\limits_{i=1}^{m} y_i A_i + S = C \\ & S \succeq 0, \end{array} \tag{2}$$

where $y \in \mathbb{R}^m$ and $S \in S\mathbb{R}^{n\times n}$.

† Corresponding author. 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-9505155 and ONR through grant N00014-96-1-0050.,

‡Department of Mathematics, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119620 (mattohkc@math.nus.sg). Research supported by a graduate scholarship from the National University of Singapore.,

§ Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213, USA (reha@nirvana.math.cmu.edu). Research supported in part by NSF through grant DMS-9505155 and ONR through grant N00014-96-1-0050 while the author was a graduate student at Cornell University.

SDPs have a wide range of applications in both continuous and combinatorial optimization (we refer the reader to [22] for an extensive list of applications). Interior-point methods for their solution were pioneered by Alizadeh [1] and Nesterov and Nemirovsky [15, 16] independently. These methods were primal; more efficient algorithms use a primal-dual approach. For a comprehensive list of publications concerned with both algorithms and applications and related software, see the semidefinite programming home pages maintained by Alizadeh [2] and Helmberg [6].

Our goal is to develop efficient algorithms (in terms of both number of iterations and number of arithmetic operations per iteration) for a wide range of classes of SDP problems. We will see that methods based on the NT direction compare favorably with other interior-point methods. One issue that is not addressed here is the exploitation of problem structure for large-scale problems. All the methods considered here use direct methods that do not preserve sparsity to solve the linear systems that arise in each iteration, while indirect methods may be preferable generally. However, if the $\{A_i\}$ and C have a common block-diagonal structure, then so do all feasible X and S, and all the algorithms here preserve and exploit this structure. In particular, the linear systems that arise can all be solved block-wise.

As in linear programming (LP), most interior-point algorithms for SDP can be viewed as damped and centered Newton methods. At each iteration, the search direction is computed by applying Newton's method to a system of nonlinear equations defining the so-called central path. However, unlike the case for LP, even for primal-dual methods, there are several ways to write these nonlinear equations. Indeed, for some choices, the direction that results from the Newton equation does not give symmetric matrices as the search directions for X and S, and hence cannot be implemented without further modification. One way to proceed is to symmetrize the complementarity equation in the system defining the central path with respect to an invertible matrix P, as was done by Monteiro [14] and Y. Zhang [23]. The resulting Newton system gives rise to different search directions for different matrices P. We give a simple proof that the symmetrized Newton system has a unique solution under certain conditions related to those appearing in the work of Shida, Shindoh and Kojima [19]. Based on this uniqueness result, we show that the NT direction can be expressed as the solution of a symmetrized Newton system. (We note that recently, independently of our work, Sturm and S. Zhang [20] described how to generate the NT direction as a Newton search direction for primal-dual path-following algorithms for SDP using a symmetric primal-dual transformation, similar to Section 6 in [17].)

In LP, the most computationally successful interior-point methods for LP have been primal-dual methods using Mehrotra's [12] predictor-corrector steps. These methods obtain higher-order directions from the nonlinear equations formed at each iteration. Because of our interpretation of the NT direction as a Newton direction, we can follow this approach also. In contrast, from the equation defining the NT direction in [17, 18], it is not clear how we can incorporate higher-order (quadratic) terms into the NT direction.

One of the main contributions of this paper is the implementation of such a predictor-corrector algorithm based on the NT direction, where the direction is computed from the solution of a linear least-squares problem via a QR factorization. For simplicity, we will refer to this method as Algorithm NT-PC-QR. We discuss issues related to efficient implementation, which are briefly described next. Firstly, in solving for the NT direction of an SDP, one needs to compute a certain scaling matrix W. A direct computation would involve two different matrix square roots. Here

we show that W can be computed with two Cholesky factorizations and one singular value decomposition (SVD). Secondly, we show how to use these to compute the search direction efficiently. Finally, computation of the quadratic correction requires the solution of a Lyapunov equation, but this can be solved explicitly and simply by re-using the Cholesky factors and SVD already generated.

For the purpose of comparison, we also implemented three other methods using exactly the same algorithmic framework as Algorithm NT-PC-QR, but computing the search direction from a Schur complement equation via a Cholesky or an LU factorization. (This approach is like solving the normal equations of the linear least-squares problem mentioned above.) The first is based on the direction of Alizadeh, Haeberly, and Overton [3]; the second on one of the H..K..M directions; and the last on the NT direction. The second direction has a convoluted history: it apparently first appeared in a preprint of Helmberg, Rendl, Vanderbei, and Wolkowicz [7] in late 1993; was rediscovered independently as one member of a family of directions introduced in early 1994 by Kojima, Shindoh, and Hara [10]; and was most recently derived (as well as a "dual" direction, also a member of the KSH family) in a different way by Monteiro [14] in 1995. This last derivation fits most conveniently with our discussions, but we have chosen a name that suggests all those who obtained it. These three methods are referred to as Algorithm AHO-PC-Sch, H..K..M-PC-Sch and NT-PC-Sch, respectively.

After the release of the first version of this paper, we observed that with careful implementations, the three algorithms that are based on the Schur complement approach can perform much better than we initially expected, obtaining duality gaps below 10^{-10} in the problems we tested.

Our numerical results indicate that the methods (both the QR and Schur complement approaches) based on the NT direction are fast and robust. For most problems, the Schur complement approach is as stable as the QR approach. However, it is known that the latter approach can be much better conditioned than the former. Conceivably, there may be problems where the QR approach can perform much better than the Schur complement approach because of this.

A brief summary of how the NT methods compare to the methods based on the AHO and H..K..M directions is as follows. The NT methods are almost as efficient as Algorithm AHO-PC-Sch in terms of the number of iterations required to attain a given accuracy, typically taking only two or three more iterations. Algorithm H..K..M-PC-Sch usually takes almost the same number of iterations as the NT methods, but occasionally it stagnates, taking very small steps, and then it requires several more steps. On the other hand, each iteration of the NT methods is about 50% cheaper than those for Algorithm AHO-PC-Sch and only slightly more expensive than those of Algorithm H..K..M-PC-Sch. Overall, the NT methods seem to be more robust than the other two algorithms in the sense that problems of stagnation appear less likely to occur. Finally, we must mention that Algorithm AHO-PC-Sch seems to be the best of the tested methods in terms of reducing the duality gap, which is typically 10 to 100 times smaller than that obtained from methods based on the H..K..M and NT directions.

This paper is organized as follows. In Section 2, we introduce the NT direction for SDP and study some of its symmetry and invariance properties. In Section 3, we describe the symmetrized Newton system and prove that it has a unique solution under a certain positive semidefiniteness condition. Then we apply this uniqueness result to show that the NT direction can be viewed as the search direction of a certain

symmetrized Newton system. We address implementation issues in Section 4: we give an efficient way to compute the scaling matrix W, show how the Schur complement equation can be expressed as a linear least-squares problem, and derive the Mehrotra-type corrector [12] for the NT direction. Results of numerical experiments are given in Section 5. Finally, we give an appendix containing some frequently used properties of the standard and symmetrized Kronecker products in the context of SDP.

Throughout this paper, $\mathbb{R}^n$ denotes the n-dimensional Euclidean space and $\|\cdot\|$ denotes the Euclidean norm. The standard inner product between two real $n \times n$ matrices G and K is the number $G \bullet K := \mathbf{Tr}\, G^T K$, the trace of the matrix $G^T K$. A matrix $G \in \mathbb{R}^{n\times n}$ (not necessarily symmetric) is said to be positive definite (resp., positive semidefinite) if $v^T G v > 0$, ($v^T G v \geq 0$, resp.) for all $v \neq 0$. However, all positive (semi)definite matrices considered here are symmetric except in the statement and proof of Theorem 3.1, so we shall assume that positive (semi)definiteness implies symmetry unless explicitly stated otherwise. We write $G \succ 0$ ($G \succeq 0$) if G is positive definite (positive semidefinite). For $G \succeq 0$, $G^{1/2}$ denotes the unique symmetric positive semidefinite square root of G. Finally, $\mathcal{N}(G)$ and $\mathcal{R}(G)$ denote the null space and range space of the matrix G, respectively.

2. The Nesterov-Todd direction and some symmetry and invariance issues. In two recent papers [17, 18], Nesterov and Todd study interior-point methods for convex programming problems expressed in conic form. They develop efficient primal-dual algorithms for a class of problems for which the cone and its associated barrier are self-scaled. This class includes semidefinite programming. All the primal-dual methods in [17, 18] use a direction that, for the case of SDP, is (up to a scalar multiple) the solution for some σ to the following system of equations:

$$\begin{array}{rcl} A_i \bullet \Delta X & = & 0, \quad i = 1, \ldots, m, \\ \sum_{i=1}^m \Delta y_i A_i + \Delta S & = & 0, \\ W^{-1}\Delta X W^{-1} + \Delta S & = & \sigma\mu X^{-1} - S, \end{array} \tag{3}$$

where W is the scaling matrix defined below and $\mu := X \bullet S/n$. The unique solution to (3) is an affine-scaling direction for $\sigma = 0$, and a centering direction for $\sigma = 1$ — see [18].

We start by describing how (3) arises. In fact, it is a special case for semidefinite programming of a system of equations defined for convex programming problems in conical form, where the cone, say K, has a *self-scaled* barrier function F. In our case, the cone K is the set of positive semidefinite matrices of order n, and the barrier function is defined by

$$F(X) := -\ln\det(X). \tag{4}$$

We will not give the definition of self-scaled barriers here (for details, see [17]), but note that a key consequence is that, for every X in the interior of the primal cone K and every S in the interior of the corresponding dual cone K^* there is a unique *scaling* point W in the interior of K with

$$F''(W)X = S. \tag{5}$$

The theory also implies that, with $T := -F'(W)$, the dual function F_*, which is a self-scaled barrier for the dual cone K^*, satisfies $F_*''(T) = [F''(W)]^{-1}$ and $F_*''(T)S = X$, and we therefore have a dual scaling point. Further, we have $F''(W)F_*'(S) = F'(X)$.

In the case of semidefinite programming, $F_*(S) = -\ln\det(S) - n$, $F'(X) = -X^{-1}$ and similarly $F'_*(S) = -S^{-1}$, and $F''(W)$ is the operator taking X to $W^{-1}XW^{-1}$ and similarly for $F''_*(T)$. The equation (5) then becomes

$$W^{-1}XW^{-1} = S, \tag{6}$$

or equivalently $WSW = X$, from which we easily see that

$$W = X^{1/2}(X^{1/2}SX^{1/2})^{-1/2}X^{1/2} = S^{-1/2}(S^{1/2}XS^{1/2})^{1/2}S^{-1/2}. \tag{7}$$

The scaling matrix W is related to various notions of geometric mean for symmetric positive definite matrices. Indeed, (6) shows that W is the metric geometric mean of X and S^{-1} (and W^{-1} that of S and X^{-1}), and

$$W^{-1/2}XW^{-1/2} = W^{1/2}SW^{1/2} \tag{8}$$

is the spectral geometric mean of X and S [4, 8].

Thus system (3) amounts to feasibility equations for the primal and dual directions, together with the equation

$$F''(W)\Delta X + \Delta S = -S - \sigma\mu F'(X). \tag{9}$$

From the results above, this is equivalent to

$$\Delta X + F''_*(T)\Delta S = -X - \sigma\mu F'_*(S). \tag{10}$$

One nice feature of the Nesterov-Todd approach is that it provides general methods applying to any convex programming problem in conic form, as long as the cone is self-scaled; such cones include the nonnegative orthant and the second-order cone in addition to the cone of positive semidefinite matrices. Thus the convergence proofs and complexity estimates apply to this general class of problems. In contrast, the other methods considered here are specific to SDP, and require new convergence proofs. (However, the algorithms we implement here employ second-order directions and long steps for which the Nesterov-Todd convergence proofs do not apply.)

In the literature on interior-point methods for semidefinite programming, one key point is the need for well-defined search directions, with ΔX and ΔS symmetric. This will be discussed further in the next section. We point out that the general development of [17, 18] leads to nonsingular linear systems defined on the appropriate linear spaces, namely the set of symmetric matrices of order n for ΔX and ΔS in our case, so that (3) has a unique solution with the directions lying in the correct spaces.

There is also an issue of primal-dual symmetry. Since (SDP) can be converted into (SDD) and vice versa, there is a natural symmetry between primal and dual. We would like our algorithms also to be symmetric between primal and dual. The equations (9) and (10) above show that (apart from the lack of symmetry in the feasibility equations, which follows from the way in which the constraints are written) the Nesterov-Todd direction does possess this primal-dual symmetry. Some of the methods we discuss below do not share this property.

There is a third property which we believe is desirable for interior-point methods: scale invariance. For linear programming, it is well known that, if the primal variables are measured in different units, then the search direction scales appropriately. That is, if x is replaced by $D^{-1}x$ where D is diagonal and positive definite, and correspondingly A becomes AD, c becomes Dc, and s becomes Ds, then the search direction

for any algorithm at the scaled trial point $(D^{-1}x, y, Ds)$ in the scaled problem is $(D^{-1}\Delta x, \Delta y, D\Delta s)$, where $(\Delta x, \Delta y, \Delta s)$ is the search direction at the unscaled point (x, y, s) in the original problem.

We will say that a search direction for semidefinite programming enjoys *scale invariance* if a similar property holds. Now we let U be any invertible $n \times n$ matrix, and consider the "scaling" where X is replaced by $\bar{X} := U^{-1}XU^{-T}$ and correspondingly each A_i by $\bar{A}_i := U^T A_i U$, C by $\bar{C} := U^T CU$ and S by $\bar{S} := U^T SU$. This defines a new, scaled problem: if (X, y, S) is feasible in the original problem and its dual, $(\bar{X}, y, \bar{S})$ is feasible in the scaled problem with the same objective function values. If the search direction at the unscaled point (X, y, S) in the original problem is $(\Delta X, \Delta y, \Delta S)$, we require the search direction at the scaled point $(\bar{X}, y, \bar{S})$ in the scaled problem to be $(\overline{\Delta X}, \Delta y, \overline{\Delta S}) := (U^{-1}\Delta XU^{-T}, \Delta y, U^T \Delta SU)$.

It is not hard to show that the Nesterov-Todd direction is scale-invariant. In fact, this follows from the coordinate-free representation of the directions, but a direct proof is easy and instructive. From equation (6), it is easy to check that the scaling point corresponding to $U^{-1}XU^{-T}$ and $U^T SU$ is $U^{-1}WU^{-T}$ where W is that corresponding to X and S. Now it is merely a matter of substituting the scaled values of $\{A_i\}$, ΔX, ΔS and W into (3) to confirm that the solutions scale appropriately.

Note that if we choose U above to be $W^{1/2}$, then both X and S are scaled to the same point by (8). This is parallel to the situation in linear programming. In this scaled space, the search directions for X and S are just Euclidean projections of the scaled right-hand side onto the appropriate orthogonal subspaces; see Section 6 of [17] for a general view of this. This scaling is the starting point for the development in Sturm and S. Zhang [20].

3. The central path and the Newton equation.

3.1. The Newton equation and symmetric search directions. For the SDP problem, the central path is defined as the set of solutions to the following equations:

$$\begin{array}{rcl} A_i \bullet X & = & b_i, \quad \text{for } i = 1, \ldots, m, \\ \sum_{i=1}^m y_i A_i + S & = & C, \\ XS & = & \nu I, \end{array} \tag{11}$$

for all $\nu > 0$ (together with the requirement that $X \succ 0$, $S \succ 0$). Given an iterate (X, y, S) (not necessarily feasible) for the primal and dual semidefinite programming problems, many interior-point algorithms generate the next iterate by taking a single Newton step applied to the nonlinear system of equations (11) or a related system. Much of the recent debate on interior-point methods for SDP has focused on the issue of the linearization of the third equation in this system. Suppose we try to approximate the point on the central path corresponding to a value $\nu = \sigma\mu$, where $\mu := X \bullet S/n$ and $\sigma \in [0, 1]$. If the solution is $(X + \Delta X, y + \Delta y, S + \Delta S)$, then we have

$$A_i \bullet \Delta X = b_i - A_i \bullet X, \quad \text{for } i = 1, \ldots, m, \tag{12-a}$$

$$\sum_{i=1}^m \Delta y_i A_i + \Delta S = C - S - \sum_{i=1}^m y_i A_i, \tag{12-b}$$

$$\Delta XS + X\Delta S = \sigma\mu I - (XS + \Delta X \Delta S). \tag{12-c}$$

The linearization of (12-c) gives:

$$\Delta X\, S + X\, \Delta S = \sigma\mu I - XS. \tag{12-d}$$

A key requirement for interior-point methods for semidefinite programming is to generate symmetric ΔX and ΔS matrices. By virtue of the equation (12-b), any ΔS that satisfies the above system has to be symmetric. However, this is not true in general for ΔX. Even when the system (12-a), (12-b), and (12-d) is feasible, there may not be any solutions with symmetric ΔX (see [10] for an example). Different recipes have been suggested to symmetrize equations (12-c) and (12-d). We follow Y. Zhang's approach [23], that unifies these symmetrizations by viewing them as parametrized transformations of (12-c) and (12-d).

In [23], Zhang defines the following linear transformation for a given invertible matrix P:

$$H_P(M) := \frac{1}{2}\left[PMP^{-1} + P^{-T}M^TP^T\right], \tag{12-e}$$

and observes that if P is invertible and M is similar to a (symmetric) positive definite matrix, then

$$H_P(M) = \nu I \Leftrightarrow M = \nu I.$$

This observation indicates that the last equation in (11) can be replaced by

$$H_P(XS) = \nu I \tag{12-f}$$

without affecting the definition of the central path, since XS is similar to the positive definite matrix $S^{1/2}XS^{1/2}$. Using this characterization of the central path, (12-c) and (12-d) are replaced by

$$H_P(\Delta X\, S \,+\, X\, \Delta S) = \sigma\mu I - H_P(XS + \Delta X \Delta S) \tag{12-c$'$}$$

and

$$H_P(\Delta X\, S \,+\, X\, \Delta S) = \sigma\mu I - H_P(XS). \tag{12-d$'$}$$

The resulting system (12-a), (12-b), and (12-d$'$) can be viewed as defined by a linear transformation from $\mathbb{R}^m \times S\mathbb{R}^{n\times n} \times S\mathbb{R}^{n\times n}$ to itself, and thus under suitable conditions will produce a unique solution $(\Delta y, \Delta X, \Delta S)$ with symmetric ΔX and ΔS. Therefore, we will restrict ΔX to be symmetric in the rest of our discussion. Note that the equation (12-d$'$) can then be written in full detail as

$$\begin{aligned} &P(\Delta X\, S \,+\, X\, \Delta S)P^{-1} + \\ &P^{-T}(S\,\Delta X \,+\, \Delta S\, X)P^T = 2\sigma\mu I - PXSP^{-1} - P^{-T}SXP^T. \end{aligned} \tag{13}$$

Alizadeh, Haeberly, and Overton's linearization [3] corresponds to $P = I$, and Monteiro's two linearizations to $P = X^{-1/2}$ and $P = S^{1/2}$ [14]. In fact, the second of Monteiro's linearizations gives the same direction as that introduced by Helmberg, Rendl, Vanderbei, and Wolkowicz [7], and both are included in a class developed by Kojima, Shindoh, and Hara [10]; see also Lin and Saigal [11]. However, the motivation of [7, 10] is very different, and to view all these directions in a unified way, we follow Monteiro's approach. We denote these the AHO direction and the H..K..M directions. In Subsection 3.4, we shall see that the NT direction also results from such a linearization for a suitable P.

3.2. Symmetrized Kronecker product notation. To compute the Newton step $(\Delta X, \Delta y, \Delta S)$, it is easier to express the linear systems of equations (12-a), (12-b) and (12-d′) in the standard matrix-vector form by using symmetrized Kronecker products. First we define an operator taking symmetric matrices into vectors: if U is an $n \times n$ symmetric matrix, $\mathbf{svec}(U)$ is defined by

$$\mathbf{svec}(U) := (u_{11}, \sqrt{2}\,u_{21}, \ldots, \sqrt{2}\,u_{n1}, u_{22}, \sqrt{2}\,u_{32}, \ldots, \sqrt{2}\,u_{n2}, \ldots, u_{nn})^T. \tag{14}$$

The factor $\sqrt{2}$ is introduced so that **svec** is an isometry between $S\mathbb{R}^{n\times n}$ and $\mathbb{R}^{n(n+1)/2}$ with their respective standard inner products. We denote the inverse map of **svec** by **smat**.

The symmetrized Kronecker product of any two $n \times n$ matrices G and K (not necessarily symmetric) is a square matrix of order $n(n+1)/2$; its action on a vector $u := \mathbf{svec}(U)$, where $U \in S\mathbb{R}^{n\times n}$, is given by

$$(G \otimes_s K)\,\mathbf{svec}(U) := \frac{1}{2}\,\mathbf{svec}(KUG^T + GUK^T). \tag{15}$$

The operator **svec** and the symmetrized Kronecker product were first used by Alizadeh et al. [3], but with the latter restricted to the case where G and K were both symmetric. We refer the reader to the appendix for some properties of the symmetrized Kronecker product, as well as corresponding properties for, and relations to, the standard Kronecker product.

Using (14) and (15), we now can write the system (12-a), (12-b) and (12-d′) in a 3×3 block equation

$$\begin{pmatrix} 0 & \mathcal{A} & 0 \\ \mathcal{A}^T & 0 & \mathcal{I} \\ 0 & E & F \end{pmatrix} \begin{pmatrix} \Delta y \\ \mathbf{svec}(\Delta X) \\ \mathbf{svec}(\Delta S) \end{pmatrix} = \begin{pmatrix} r_p \\ \mathbf{svec}(R_d) \\ \mathbf{svec}(R_c) \end{pmatrix}, \tag{16}$$

where $\mathcal{I}$ is the identity matrix of order $n(n+1)/2$,

$$E := P \otimes_s P^{-T}S, \quad F := PX \otimes_s P^{-T}, \tag{17}$$

$$r_p := b - \mathcal{A}\,\mathbf{svec}(X), \quad R_d := C - S - \sum_{i=1}^{m} y_i A_i, \quad R_c := \sigma\mu I - H_P(XS). \tag{18}$$

Here, $\mathcal{A}$ is a matrix of dimension $m \times n(n+1)/2$ defined by

$$\mathcal{A}^T := [\mathbf{svec}(A_1) \cdots \mathbf{svec}(A_m)]\,. \tag{19}$$

The assumption that the $\{A_i\}$ are linearly independent implies that $\mathcal{A}$ has full row rank.

3.3. Uniqueness of symmetric search directions. For the rest of our presentation it is essential to understand the conditions under which the system (16) has a unique solution with symmetric ΔX and ΔS. In their recent paper [19], Shida, Shindoh, and Kojima established the existence and uniqueness of the solutions of (16) for some choices of the matrix P, as well as some other systems that may not be expressible in the form (16). They show that when $P = X^{-1/2}$, the system (16) always has a unique solution as long as $X \succ 0$ and $S \succ 0$, but that this is not true for $P = I$: in this case the additional the condition that $XS + SX \succeq 0$ is derived as a sufficient condition for (16) to have a unique solution. This last condition turns

out to be sufficient but not necessary. Before generalizing their result to arbitrary invertible matrices P, we observe that the matrices E and F are nonsingular when X and S are positive definite. Indeed,

$$E = P \otimes_s P^{-T} S = (I \otimes_s P^{-T} S P^{-1})(P \otimes_s P), \tag{20}$$

and

$$F = PX \otimes_s P^{-T} = (P X P^T \otimes_s I)(P^{-T} \otimes_s P^{-T}), \tag{21}$$

where the equalities follow from the properties of the symmetrized Kronecker product listed in the appendix. Now, E and F are nonsingular since both factors in (20) and (21) are nonsingular (see the appendix).

THEOREM 3.1. *Suppose X and S are positive definite. Then the system of equations (16) has a unique solution $(\Delta y, \Delta X, \Delta S) \in \mathbb{R}^m \times S\mathbb{R}^{n\times n} \times S\mathbb{R}^{n\times n}$ if $E^{-1}F$ is positive definite (not necessarily symmetric). In particular, this condition holds when X, S, and $H_P(XS)$ are positive semidefinite.*

Proof. For this proof, positive definiteness does not imply symmetry; if matrices are symmetric, we will explicitly say so.

It suffices to show that the 3×3 block system (with a square matrix)

$$\begin{pmatrix} 0 & \mathcal{A} & 0 \\ \mathcal{A}^T & 0 & \mathcal{I} \\ 0 & E & F \end{pmatrix} \begin{pmatrix} \Delta y \\ \mathbf{svec}(\Delta X) \\ \mathbf{svec}(\Delta S) \end{pmatrix} = 0 \tag{22}$$

only has the trivial solution.

Since E is invertible by (20), we can use block Gaussian elimination to reduce (22) to a Schur complement equation:

$$(\mathcal{A} E^{-1} F \mathcal{A}^T)\, \Delta y = 0. \tag{23}$$

Since $E^{-1}F$ is assumed to be positive definite and $\mathcal{A}$ has full row rank, this implies that $\mathcal{A}E^{-1}F\mathcal{A}^T$ is positive definite and thus $\Delta y = 0$ is the only solution to (23).

Now the second block equation in (22) gives $\Delta S = -\mathbf{smat}(\mathcal{A}^T \Delta y) = 0$, and the third block equation gives $\Delta X = -E^{-1}F\,\Delta S = 0$. This shows that the solution to (16) exists and is unique.

To show that the condition $H_P(XS)$ is positive semidefinite implies that $E^{-1}F$ is positive definite, consider an arbitrary $n(n+1)/2$-dimensional nonzero vector g. We shall prove that $g^T E^{-1} F g > 0$. Let $k := E^{-T} g$ and $K := \mathbf{smat}(k)$, so that k is also nonzero. We have

$$\begin{aligned} g^T E^{-1} F g &= k^T F E^T k = k^T (PX \otimes_s P^{-T})(P^T \otimes_s S P^{-1})\, k \\ &= \frac{1}{2}\, k^T (PXP^T \otimes_s P^{-T} S P^{-1})\, k + \frac{1}{2}\, k^T (PXSP^{-1} \otimes_s I)\, k \\ &> \frac{1}{2}\, k^T (PXSP^{-1} \otimes_s I)\, k \\ &= \frac{1}{4}\, (\mathbf{svec} K)^T \mathbf{svec}(PXSP^{-1}K + KP^{-T}SXP^T) \\ &= \frac{1}{4}\, \mathbf{Tr}\left[K(PXSP^{-1} + P^{-T}SXP^T)K\right] \\ &= \frac{1}{2}\, \mathbf{Tr}\left[K\, H_P(XS)\, K\right] \;\geq\; 0. \end{aligned}$$

The strict inequality above holds because $PXP^T \otimes_s P^{-T}SP^{-1}$ is positive definite, a consequence of the fact that PXP^T and $P^{-T}SP^{-1}$ are symmetric positive definite (see the appendix). The last inequality follows from the assumption that $H_P(XS)$ is positive semidefinite. This completes the proof. □

We note that, if $P = X^{-1/2}$ or, more generally, $P = (X^{1/2}SX^{1/2})^{\alpha}X^{-1/2}$ for some α, then $H_P(XS) = X^{1/2}SX^{1/2}$, which is positive definite whenever X and S are positive definite. Similarly, when $P = S^{1/2}$, or $P = (S^{1/2}XS^{1/2})^{\alpha}S^{1/2}$, we have $H_P(XS) = S^{1/2}XS^{1/2}$, which again is positive definite whenever X and S are. When $P = I$, the sufficient condition given in Theorem 3.1 (i.e. that $H_P(XS)$ is positive semidefinite) coincides with the condition in [19]. So, as in [19], we see that the system (16) always has a unique solution with symmetric ΔX and ΔS for Monteiro's choices of P. This also holds for Alizadeh et al.'s choice of P if $XS + SX$ is positive semidefinite, but there are examples where the coefficient matrix in (16) is singular. One such example is given by

$$\mathcal{A} = [-1 \quad 2 \quad 0], \quad X = \begin{bmatrix} 1 & \sqrt{2} \\ \sqrt{2} & 3 \end{bmatrix}, \quad S = \begin{bmatrix} 1 & 0 \\ 0 & 11 \end{bmatrix}.$$

Indeed, in this case the Schur complement matrix in (23) is the 1×1 zero matrix.

The following theorem describes a class of matrices P that satisfy the sufficient condition of Theorem 3.1.

THEOREM 3.2. *Suppose X and S are positive definite matrices and P is an invertible matrix. Let E and F be as defined in (17). Then the following are equivalent:*

(a) PXP^T and $P^{-T}SP^{-1}$ commute;

(b) $PXSP^{-1}$ is symmetric;

(c) FE^T is symmetric; and

(d) $E^{-1}F$ is symmetric.

Condition (a) implies that $H_P(XS)$ is positive definite. Consequently, any of the conditions above imply that $E^{-1}F$ and FE^T are both positive definite.

Proof. The equivalence of (a) and (b) is immediate: PXP^T and $P^{-T}SP^{-1}$ commute if and only if

$$PXSP^{-1} = (PXP^T)(P^{-T}SP^{-1}) = (P^{-T}SP^{-1})(PXP^T) = P^{-T}SXP^T. \tag{24}$$

On the other hand,

$$FE^T = (PX \otimes_s P^{-T})(P^T \otimes_s SP^{-1}) = \frac{1}{2}(PXP^T \otimes_s P^{-T}SP^{-1} + PXSP^{-1} \otimes_s I).$$

The first term in the last equality above is symmetric, since PXP^T and $P^{-T}SP^{-1}$ are. Therefore FE^T is symmetric if and only if the second term is. But this holds if and only if $PXSP^{-1}$ is symmetric. This shows the equivalence of (b) and (c). Finally, the equivalence of (c) and (d) is obvious: $E^{-1}F = E^{-1}(FE^T)E^{-T}$ is symmetric if and only if FE^T is.

Now suppose (a) holds (thus (b) holds). Then

$$H_P(XS) = PXSP^{-1} = (PXP^T)(P^{-T}SP^{-1}),$$

which is the product of two commuting positive definite matrices. This implies that $H_P(XS)$ is positive definite. Consequently, by Theorem 3.1, $E^{-1}F$ and hence FE^T are positive definite. □

Direction	Primal-dual symmetry	Scale invariance	Directions uniquely defined
AHO ($P = I$) [3]	yes	no	no
H..K..M ($P = X^{-1/2}$ or $S^{1/2}$) [7, 10, 14]	no	yes	yes
NT ($P = W^{-1/2}$) [17, 18]	yes	yes	yes

TABLE 1
Properties of the AHO, H..K..M, and NT directions

The first of the equivalent conditions of Theorem 3.2 holds for the two Monteiro choices for P (one of the two matrices is the identity), and also for $P = W^{-1/2}$ (the two matrices are equal from (8)); we will see much more of this choice in the rest of the paper.

Of the two H..K..M directions and the AHO direction, only the latter is primal-dual symmetric. Turning now to scale invariance, we see that the AHO direction is *not* scale-invariant; XS transforms to $U^{-1}XSU$ while SX transforms to U^TSXU^{-T}, and similarly for products involving directions.

To see that the other two directions are scale-invariant, we note first that the solution to (16) corresponding to P is the same as that corresponding to QP for any orthogonal Q; indeed, the direction depends only on P^TP, as is easily seen by pre- and post-multiplying (13) by P^T and P^{-T} respectively. Next we use the following result:

LEMMA 3.3. *Suppose $B = CC^T$ is positive definite. Then $Q := C^{-1}B^{1/2}$ is orthogonal.*

Proof. We see that $QQ^T = C^{-1}BC^{-T} = C^{-1}CC^TC^{-T} = I$. □

THEOREM 3.4. *Suppose P is defined from X and S in such a way that, if X and S are transformed into $\bar{X} := U^{-1}XU^{-T}$ and $\bar{S} := U^TSU$, then P is transformed into $\bar{P}$ with $\bar{P}^T\bar{P} = U^TP^TPU$. Suppose also that $H_P(XS)$ is positive definite. Then the direction that solves (16) is scale-invariant. In particular this is true for the directions defined by $P = X^{-1/2}$, $P = S^{1/2}$, and $P = W^{-1/2}$.*

Proof. By letting $B := \bar{P}^T\bar{P} = U^TP^TPU$ and applying the lemma twice, we find that $\bar{P} = QPU$ for some orthogonal Q. It is also easy to see that $H_{\bar{P}}(\bar{X}\bar{S})$ is positive definite too, so that the original and the scaled system both have unique solutions by Theorem 3.1. From our remarks above, the resulting search direction in the scaled problem is the same as if we used $\bar{P} = PU$; but it is easy to check that the scaled direction $(\overline{\Delta X}, \Delta y, \overline{\Delta S}) := (U^{-1}\Delta XU^{-T}, \Delta y, U^T\Delta SU)$ satisfies the resulting system (16). As noted above the three choices of P given satisfy condition (a) of Theorem 3.2, and therefore the condition that $H_P(XS)$ is positive definite. □

This theorem implies that the two H..K..M directions as well as the direction corresponding to $P = W^{-1/2}$ are scale-invariant. In the next subsection we will show that this last direction is actually the Nesterov-Todd direction. Table 1 summarizes the symmetry properties of the aforementioned directions.

3.4. The Nesterov-Todd direction as a Newton direction. In this subsection, we show that the system (3) defining the NT direction is actually equivalent to (16) for some choices of P. These systems are not quite comparable: (16) allows for

infeasible iterates, while (3) assumes that feasibility is maintained. Let us denote by (3') the system (3) where the two zeroes on the right-hand side are replaced by r_p and R_d respectively. We note that, independently of our work, Sturm and S. Zhang [20] recently derived the Nesterov-Todd direction as a Newton direction for path-following algorithms using the symmetric transformation given in (8).

We first note that if an invertible matrix P satisfies $P^TP = W^{-1}$, then

$$P^TPXSP^{-1}P^{-T} = W^{-1}XSW = SW\,SW = SWW^{-1}X = SX. \tag{25}$$

Above we used the fact that

$$W^{-1}X = SW, \tag{26}$$

which follows from (6). Now, (25) indicates that

$$PXSP^{-1} = P^{-T}SXP^T.$$

Hence, using Theorem 3.2, we conclude that $H_P(XS)$ is positive definite.

There are several matrices P that satisfy $P^TP = W^{-1}$, for example $P = W^{-1/2}$ and $P = (X^{1/2}SX^{1/2})^{1/4}X^{-1/2}$. We will give another useful choice in the next subsection. We are now ready to show that the NT direction is a Newton direction.

THEOREM 3.5. *Let P be an invertible matrix such that $P^TP = W^{-1}$. Then the solutions to (16) and (3') are identical.*

Proof. It suffices to show the equivalence of (13) and the last equation in (3). We observe that (13) can be rewritten as

$$W^{-1}(X\,\Delta S + \Delta X\,S)W + \Delta S\,X + S\,\Delta X = 2\sigma\mu I - 2SX \tag{27}$$

by premultiplying it with P^T, postmultiplying it with P^{-T}, and using $P^TP = W^{-1}$. On the other hand, the last equation of (3) is equivalent to each of the following two equations:

$$W^{-1}\,\Delta X\,W^{-1}\,X + \Delta S\,X = \sigma\mu I - SX, \tag{28}$$

$$S\,\Delta X + S\,W\,\Delta S\,W = \sigma\mu S\,W\,X^{-1}\,W - S\,W\,S\,W. \tag{29}$$

For the first equality above, we postmultiplied the last equation of (3) by X, and for the second, we premultiplied it with SW, and postmultiplied it with W. Now adding up these two equations, and using (26) as well as its equivalent variants $SWX^{-1}W = I$, $WSW = X$, we get (27). This proves that the solution to (3') will also satisfy (16). However, (16) has a unique solution as indicated by the remarks above and Theorem 3.1, and therefore these two systems are equivalent. □

4. Implementation. This section describes in detail how to perform the necessary computations at each iteration, and how to define and compute a second-order corrector modification (based on Mehrotra's corrector for LP [12]) of the Nesterov-Todd direction.

4.1. Computation of W. The main purpose here is to show that W can be computed using two Cholesky factorizations and one singular value decomposition (SVD).

Let the Cholesky factorizations of the positive definite matrices X and S be

$$X = LL^T, \qquad S = RR^T, \tag{30}$$

and let $UDV^T = R^T L$ be the SVD of $R^T L$. Define $Q := L^{-1}X^{1/2}$. By Lemma 3.3, Q is an orthogonal matrix. It is easily seen that

$$X^{1/2}SX^{1/2} \;=\; Q^T(L^TR)\,(R^TL)Q \;=\; (Q^TV)\,D^2\,(V^TQ).$$

Note that VD^2V^T is the eigenvalue decomposition of $L^TRR^TL = L^TSL$, so we could use this instead of the SVD. But the condition number of this matrix is the square of that of R^TL, so this may lead to a less stable algorithm. Since Q^TV is orthogonal, we have

$$(X^{1/2}SX^{1/2})^{-1/2} = (Q^TV)D^{-1}(V^TQ). \tag{31}$$

From (31), W can be computed easily:

$$W \;=\; LVD^{-1}V^TL^T \;=\; GG^T, \tag{32}$$

where

$$G := LVD^{-1/2}. \tag{33}$$

We will use the matrix G in the computation of the direction $(\Delta X, \Delta y, \Delta S)$. If we wish to use the second-order corrector modification, we will also need $G^{-1} = D^{1/2}V^TL^{-1}$, but this is easy to compute. For later use, note the following equation:

$$G^TSG \;=\; G^{-1}XG^{-T} \;=\; D, \tag{34}$$

i.e., G scales X and S to the same diagonal matrix, cf. (8).

We remark that $G^{-T}G^{-1} = W^{-1}$ from (32), so that $P := G^{-1}$ satisfies $P^TP = W^{-1}$ and hence is yet another choice for P that yields the NT direction. In this case, not only do $P^{-T}SP^{-1}$ and PXP^T commute (see Theorem 3.2), they are actually the same diagonal matrix.

Note that, if we use the "reverse" Cholesky factorization of S, so that L is lower but R upper triangular in (30), then we save work in computing the lower triangular matrix R^TL. Also, we do not need to keep the orthogonal matrix U; V and D are all we need.

4.2. Computation of the NT direction. Using Theorem 3.5 and the choice $P = G^{-1}$, we can obtain the NT direction as the solution to

$$\begin{pmatrix} 0 & \mathcal{A} & 0 \\ \mathcal{A}^T & 0 & \mathcal{I} \\ 0 & E & F \end{pmatrix} \begin{pmatrix} \Delta y \\ \mathbf{svec}(\Delta X) \\ \mathbf{svec}(\Delta S) \end{pmatrix} = \begin{pmatrix} r_p \\ \mathbf{svec}(R_d) \\ \mathbf{svec}(R_c) \end{pmatrix}, \tag{35}$$

where

$$E \;=\; G^{-1} \otimes_s G^TS, \quad F \;=\; G^{-1}X \otimes_s G^T,$$

$$r_p \;:=\; b - \mathcal{A}\,\mathbf{svec}(X), \quad R_d \;:=\; C - S - \sum_{i=1}^{m} y_iA_i, \quad R_c \;:=\; \sigma\mu I \;-\; D^2.$$

(We used $H_{G^{-1}}(XS) = (G^{-1}XSG + G^TSXG^{-T})/2 = D^2$ from (34).) Here, $\mathcal{A}$ is the matrix defined by (19).

The system (35) involves $m + n(n+1)$ linear equations. A standard method to solve this large system efficiently is to use block Gaussian elimination to reduce it to a Schur complement equation involving only Δy:

$$(\mathcal{A}E^{-1}F\mathcal{A}^T)\,\Delta y = r_p \,+\, \mathcal{A}E^{-1}F\,\mathbf{svec}(R_d) \,-\, \mathcal{A}E^{-1}\,\mathbf{svec}(R_c). \tag{36}$$

In our case, E^{-1} is not easy to obtain, but we only need $E^{-1}F$ and $E^{-1}\,\mathbf{svec}(R_c)$. It is easy to check directly, by premultiplying by E, that

$$E^{-1}F = W \otimes_s W, E^{-1}\,\mathbf{svec}(R_c) = \mathbf{svec}(\sigma\mu S^{-1} - X). \tag{37}$$

So $E^{-1}F$ is symmetric and positive definite, and it can be factorized as $E^{-1}F = KK^T$, where $K = G \otimes_s G$, obtained from the identity $W \otimes_s W = (G \otimes_s G)(G \otimes_s G)^T$. We will call such a factorization of $E^{-1}F$ a Cholesky-like factorization since the factors are transposes of one another, although they may not be triangular.

Using the Cholesky-like factorization of $E^{-1}F$, we have

$$\mathcal{A}E^{-1}F\mathcal{A}^T \;=\; \mathcal{B}\,\mathcal{B}^T,$$

and the equation (36) can be rewritten as

$$\mathcal{B}\,\mathcal{B}^T\Delta y = r_p \,+\, \mathcal{B}\,h, \tag{38}$$

where

$$\mathcal{B}^T := K^T\mathcal{A}^T \;=\; \big[\mathbf{svec}(G^TA_1G) \,\cdots\, \mathbf{svec}(G^TA_mG)\big] \tag{39}$$

and

$$\begin{aligned} h &= K^T\,\mathbf{svec}(R_d) \,-\, K^{-1}\,E^{-1}\,\mathbf{svec}(R_c), \\ &= \mathbf{svec}(G^TR_dG \,-\, \sigma\mu D^{-1} \,+\, D). \end{aligned} \tag{40}$$

For the last expression, we used (37) and (34) to obtain

$$K^{-1}E^{-1}\,\mathbf{svec}(R_c) \;=\; (G^{-1} \otimes_s G^{-1})\,\mathbf{svec}(\sigma\mu S^{-1} - X) \;=\; \mathbf{svec}(\sigma\mu D^{-1} - D).$$

Note that the equation (38) actually consists of the normal equations of a linear least-squares problem. To see this, choose a matrix $X_r \in \mathcal{S}\mathbb{R}^{n\times n}$ such that $\mathcal{A}\,\mathbf{svec}(X_r) = r_p$ (such a matrix exists because $\mathcal{A}$ has full row rank). By using $r_p = \mathcal{B}K^{-1}\mathbf{svec}(X_r)$, we can rewrite (38) as

$$\mathcal{B}\mathcal{B}^T\Delta y \;=\; \mathcal{B}\,(K^{-1}\,\mathbf{svec}(X_r) + h),$$

the normal equations of the following least-squares problem:

$$\min_{\Delta y} \|\mathcal{B}^T\Delta y - K^{-1}\,\mathbf{svec}(X_r) - h\|.$$

There are two ways to compute Δy. One is to compute Δy from the Schur complement equation (36) via a Cholesky factorization of $\mathcal{A}E^{-1}F\mathcal{A}^T$. The other

is to compute Δy from (38) via a QR factorization of $\mathcal{B}^T$ [5]. In exact arithmetic, these two methods are the same. But in finite precision arithmetic, they may have different numerical stability. Let κ be the condition number of $\mathcal{A}E^{-1}F\mathcal{A}^T$. It is well known from linear perturbation theory that solving (36) via a Cholesky factorization of $\mathcal{A}E^{-1}F\mathcal{A}^T$ produces a computed Δy whose relative error is dependent on κ; whereas solving (38) via a QR factorization of $\mathcal{B}^T$ produces a computed Δy whose relative error is dependent on a number that is between $\sqrt{\kappa}$ and κ. Thus, solving (38) via the QR approach is conveivably better than using the Schur complement approach. In our experiments later, we consider both approaches.

Assuming that the step Δy has been computed from (38) via either QR or Cholesky factorization, we can compute ΔS simply from the second block equation of (35), namely,

$$\Delta S = R_d - \mathbf{smat}(\mathcal{A}^T \Delta y). \tag{41}$$

and ΔX from the third block equation in (35):

$$\Delta X = \mathbf{smat}[E^{-1}\,\mathbf{svec}(R_c) - E^{-1}F\,\mathbf{svec}(\Delta S)]. \tag{42}$$

We have shown that Δy in the NT direction can be computed via a QR factorization of $\mathcal{B}^T$. The same can be done for the H..K..M direction corresponding to $P = S^{1/2}$, where now $E^{-1}F = X \otimes_s S^{-1}$. To show this, as before, the main task is getting a Cholesky-like factorization of $E^{-1}F$. By using (34), we have

$$\begin{aligned} E^{-1}F \;=\; X \otimes_s S^{-1} &= GDG^T \otimes_s GD^{-1}G^T \\ &= (G \otimes_s G)\,(D \otimes_s D^{-1})\,(G^T \otimes_s G^T). \end{aligned} \tag{43}$$

The middle term $D \otimes_s D^{-1}$ in (43) is a diagonal matrix with positive entries of the form $(d_i d_j^{-1} + d_j d_i^{-1})/2$, where the d_i's are the diagonal entries of the matrix D. So the square root of $D \otimes_s D^{-1}$, say M, exists and can be computed trivially. It is now easy to see that we have the Cholesky-like factorization $E^{-1}F = KK^T$, where $K := (G \otimes_s G)M$. Y. Zhang [24] also observed the same Cholesky-like factorization of $E^{-1}F$ for the H..K..M direction, and Kojima [9] noted this for the linear system involving Kronecker (rather than symmetric Kronecker) products where ΔX is explicitly symmetrized afterwards.

In fact, the QR approach can be applied almost as easily to the direction that solves (16) whenever $E^{-1}F$ is symmetric. This is because $E^{-1}F$ can be expressed in the form

$$E^{-1}F = (P^{-1}Q \otimes_s P^{-1}Q)\,\tilde{D}\,(P^{-1}Q \otimes_s P^{-1}Q)^T,$$

where $\tilde{D}$ is a positive definite diagonal matrix and Q is an orthogonal matrix whose columns form an orthonormal basis that simultaneously diagonalizes PXP^T and $P^{-T}SP^{-1}$. Note that such a basis exists because by Theorem 3.2, the latter two matrices commute if $E^{-1}F$ is symmetric. (If $P = G^{-1}$, so that we have the NT direction, then Q and $\tilde{D}$ are identity matrices.)

Finally, we note that Δy in the AHO direction cannot be computed via the QR approach because the corresponding matrix $E^{-1}F$ is not symmetric.

4.3. A second-order correction to the NT direction. To derive a second-order correction to the NT direction, we suppose that we have approximations δX

and δS to the search directions, and substitute them on the right-hand side of (12-c′). Then, we get

$$E\,\mathbf{svec}(\Delta X) \;+\; F\,\mathbf{svec}(\Delta S) \;=\; \mathbf{svec}(R_s),$$

where $E = P \otimes_s P^{-T}S$, $F = PX \otimes_s P^{-T}$ and

$$R_s \;=\; \sigma\mu I - H_P(XS + \delta X\,\delta S)$$

(subscript s for second-order). Then

$$R_s = R_c + R_q,$$

where R_q is the "quadratic correction" $-H_P(\delta X\,\delta S)$. It is easily seen that ΔX and ΔS depend on P only through the product P^TP. For the NT direction we can therefore again choose $P = G^{-1}$.

If we then proceed exactly as in the previous subsection, we see that we only have to replace $E^{-1}\mathbf{svec}(R_c)$ by $E^{-1}\mathbf{svec}(R_s)$ in (40) and (42). Thus in the second expression for h in (40), $-\sigma\mu D^{-1} + D$ must be replaced by $-\sigma\mu D^{-1} + D - R_{NT}$, where the correction R_{NT} for the NT direction is given by

$$\mathbf{svec}(R_{NT}) \;:=\; K^{-1}E^{-1}\,\mathbf{svec}[-H_{G^{-1}}(\delta X\delta S)].$$

Thus R_{NT} is the solution to $EK\,\mathbf{svec}(R_{NT}) = \mathbf{svec}[-H_{G^{-1}}(\delta X\delta S)]$. But $K = G\otimes_s G$ and $E = G^{-1} \otimes_s G^TS$, so this simplifies to the Lyapunov equation

$$(G^TSG)R_{NT} + R_{NT}(G^TSG) \;=\; -G^{-1}(\delta X\,\delta S)G - G^T(\delta S\,\delta X)G^{-T}.$$

But by (34), G^TSG is the diagonal matrix D, and we can therefore write down the solution explicitly:

$$R_{NT} \;=\; -\,(G^{-1}(\delta\,X\delta S)G + G^T(\delta S\,\delta X)G^{-T})./(de^T + ed^T), \tag{44}$$

where "./" means entrywise division, $d := \mathrm{diag}(D)$, and $e := (1,\ldots,1)^T$. If we wished to use the Schur complement equation (36), the term $E^{-1}\mathbf{svec}(R_c)$ on the right-hand side would be replaced by

$$E^{-1}\mathbf{svec}(R_s) \;=\; \mathbf{svec}(\sigma\mu S^{-1} - X) + (G \otimes_s G)\mathbf{svec}(R_{NT}). \tag{45}$$

We also make this replacement in (42).

Note that the derivation of a second-order correction for the NT direction was possible only after the observation that the NT direction can be viewed as a Newton direction (cf. Section 3.4); neither the original description (3) of the NT direction in [17, 18], nor the derivation by Sturm and S. Zhang [20] directly lead to a such a correction.

4.4. A Mehrotra predictor-corrector algorithm based on the NT direction. The results derived in the previous subsections are the pieces involved in various steps of a Mehrotra-type predictor-corrector algorithm. Now we are ready to describe such a method based on the NT direction.

Algorithm NT-PC-QR. Suppose given an initial iterate (X^0, y^0, S^0) with X^0, S^0 positive definite. Choose $\tau \in (0,1)$ and $expon \in \{1,2,3\}$.

For $k = 0, 1, \ldots,$

let the current and the next iterate be (X, y, S) and (X^+, y^+, S^+) respectively. Let $\mu := X \bullet S/n$.

- (Predictor step)
  Compute the Newton step $(\delta X, \delta y, \delta S)$ from (38), (41) and (42), with $\sigma := 0$ in (40).
- Determine the parameter σ:

$$\sigma \;:=\; \left[\frac{(X+\alpha\delta X)\bullet(S+\beta\delta S)}{X\bullet S}\right]^{expon}. \tag{46}$$

  Here

$$\alpha \;:=\; \min\left(1, \frac{-\tau}{\lambda_{\min}(X^{-1}\delta X)}\right), \quad \beta \;:=\; \min\left(1, \frac{-\tau}{\lambda_{\min}(S^{-1}\delta S)}\right) \tag{47}$$

  are step lengths chosen to ensure that $X + \alpha\delta X$ and $S + \beta\delta S$ are positive definite (if the minimum eigenvalue in either expression is positive, we ignore the corresponding term).
- (Corrector step)
  With σ and μ determined as above, compute the Newton step $(\Delta X, \Delta y, \Delta S)$ from (38), (41) and (42) with the matrix $-\sigma\mu D^{-1} + D$ in (40) replaced by the matrix $-\sigma\mu D^{-1} + D - R_{NT}$, and the vector $E^{-1}\mathbf{svec}(R_c)$ in (42) replaced by $E^{-1}\mathbf{svec}(R_s)$ in (45), where R_{NT} is given by (44).
- Update (X, y, S) to (X^+, y^+, S^+) by

$$X^+ \;=\; X \;+\; \alpha\,\Delta X, \quad y^+ \;=\; y \;+\; \beta\,\Delta y, \quad S^+ \;=\; S \;+\; \beta\,\Delta S,$$

  where α and β are determined from (47) with δX, δS replaced by ΔX, ΔS, so that X^+ and S^+ are positive definite.

Remarks.

- In LP, the step-length parameter τ is fixed at a value very close to one, typically .99995. This choice seems to be too aggressive for Algorithm NT-PC-QR and experiments have shown that $\tau = 0.98$ is usually a good parameter value. In our experiments, we also allow the step-length parameter τ to be chosen adaptively based on the step-lengths taken in the previous iteration, i.e., by setting

$$\tau^{(k+1)} \;=\; 0.9 + 0.09\min(\alpha^{(k)}, \beta^{(k)}), \tag{48}$$

  where $\alpha^{(k)}$ and $\beta^{(k)}$ are the primal and dual step-lengths in the kth corrector step respectively.
- In our experiments with Algorithm NT-PC-QR, the parameter *expon* used for choosing σ in (46) is chosen to be one. This is more conservative than using an exponent of three as in LP.
- When solving (38) in the predictor step by Householder QR factorization, the Householder vectors are saved for later use in the corrector step, as is normally done for the LU factorization of the Schur complement matrix. In this way, the corrector step can be computed with just $O(mn^2)$ flops.

- In computing α in (47), it is cheaper to compute the minimum eigenvalue of the symmetric matrix $L^{-1}\delta X L^{-T}$, where $X = LL^T$ is the Cholesky factorization of X. This matrix has the same spectrum as that of the nonsymmetric matrix $X^{-1}\delta X$. We do the same when computing β.
- Actually, it is more efficient to do most of the computations with "G"-scaled quantities. At the beginning of the iteration, we compute G and hence $\mathcal{B}$ and $\bar{R}_d := G^T R_d G$. Hence we calculate h from (40) and thus δy from (38). We then obtain

$$\overline{\delta S} := G^T \delta S\, G = \bar{R}_d - \mathbf{smat}(\mathcal{B}^T \delta y)$$

and

$$\begin{aligned}\overline{\delta X} := G^{-1}\delta X\, G^{-T} &= \mathbf{smat}(K^{-1}E^{-1}\mathbf{svec}(R_c)) - \overline{\delta S} \\ &= \sigma\mu D^{-1} - D - \bar{R}_d + \mathbf{smat}(\mathcal{B}^T\delta y).\end{aligned}$$

Next we compute α and β from (47), but using $\lambda_{\min}(D^{-1/2}\,\overline{\delta X}\,D^{-1/2})$ and similarly for S; note that in transformed space we want $D + \alpha\,\overline{\delta X} \succ 0$, since X and S both transform to $\bar{X} = \bar{S} = D$. Then we obtain σ from (46), but using barred quantities. For the corrector step, $-\sigma\mu D^{-1} + D$ in (40) is replaced by

$$-\sigma\mu D^{-1} + D + (\overline{\delta X}\,\overline{\delta S} + \overline{\delta S}\,\overline{\delta X})./(de^T + ed^T).$$

Then proceed as above to obtain Δy, $\overline{\Delta X}$, and $\overline{\Delta S}$. Calculate new step lengths α and β. Finally we set

$$X^+ = G(D + \alpha\,\overline{\Delta X})G^T, \quad y^+ = y + \beta\,\Delta y, \quad S^+ = S + \beta\,\Delta S,$$

where ΔS is computed from (41) to maintain dual feasibility.
- The stopping criterion we use in our algorithm is as follows: we stop the iteration if $\sigma > 1$. This usually happens when the duality gap is in the range of the achievable accuracy of the algorithm. However, we note that more precise stopping criteria need to be devised. We also stop the iteration if the step lengths α and β in the corrector step are smaller than 10^{-6}.
- Assuming that the Householder vectors are saved when solving (38) in the predictor step, the complexity of each iteration of this algorithm is $3mn^3 + \frac{1}{2}m^2n^2 + O(n^3 + mn^2)$ floating point operations.

5. Numerical experiments. Our purpose here is to compare Algorithm NT-PC-QR with methods using the same algorithmic framework, but computing the directions from the Schur complement equation (36) via a Cholesky or an LU factorization of $\mathcal{A}E^{-1}F\mathcal{A}^T$. In the latter methods, we consider three different search directions (we only consider one of the H..K..M directions, the one that coincides with the direction of Helmberg, Rendl, Vanderbei, and Wolkowicz [7] and the one that is easier to compute — see Y. Zhang [23]):

1. AHO direction ($P = I$);
2. H..K..M direction ($P = S^{\frac{1}{2}}$);
3. NT direction ($P = G^{-1}$).

These three algorithms will be referred to as Algorithm AHO-PC-Sch, H..K..M-PC-Sch and NT-PC-Sch, respectively.

Our code is based on one of Alizadeh, Haeberly, and Overton for feasible iterates; we are grateful to them for making it available to us. We also mention here that they have had success with the larger step-length parameter value of 0.999.

We have already discussed how $(\Delta X, \Delta y, \Delta S)$ is computed for Algorithm NT-PC-QR. Now we briefly describe how it is computed for the other three methods. For simplicity, we only discuss the predictor step. The step δy is first computed from the Schur complement equation (36) via a Cholesky or LU factorization of the $m \times m$ Schur complement matrix $\mathcal{A}E^{-1}F\mathcal{A}^T$, according to whether the matrix is known to be symmetric positive definite or not. Once δy is obtained, δS is computed from (41) and δX from (42). It is clear that the main work involved in these methods is the computation of the coefficient matrix and the right-hand side vector of the Schur complement equation. In Table 5, we summarize some of the formulae that are needed in this computation. The complexity of each iteration of these methods is also included, ignoring lower order terms like $O(n^3 + mn^2)$.

directions	AHO ($P = I$)	H..K..M ($P = S^{\frac{1}{2}}$)	NT ($P = G^{-1}$)
E	$I \otimes_s S$	$S^{\frac{1}{2}} \otimes_s S^{\frac{1}{2}}$	$G^{-1} \otimes_s G^T S$
F	$X \otimes_s I$	$S^{\frac{1}{2}} X \otimes_s S^{-\frac{1}{2}}$	$G^{-1} X \otimes_s G^T$
$E^{-1}F$	$(I \otimes_s S)^{-1}(X \otimes_s I)$	$X \otimes_s S^{-1}$	$W \otimes_s W$
R_c	$\sigma\mu I - (XS + SX)/2$	$\sigma\mu I - S^{\frac{1}{2}} X S^{\frac{1}{2}}$	$\sigma\mu I - D^2$
R_q	$-(\delta X\, \delta S + \delta S\, \delta X)/2$	$-H_{S^{\frac{1}{2}}}(\delta X\, \delta S)$	see (45)
complexity	$8mn^3 + m^2n^2$	$4mn^3 + 0.5m^2n^2$	$3mn^3 + 0.5m^2n^2$

TABLE 2

Summary of the formulae involved in the computation of the coefficient matrix and the right-hand side of the Schur complement equation for the AHO, H..K..M, and NT directions. We count one addition and one multiplication each as one flop.

For all the problems below, the starting points are feasible and so feasibility is maintained throughout in exact arithmetic. Our formulae for ΔS guarantee that dual feasibility is preserved to high accuracy, but the same is not true for primal feasibility. Hence we adjust $\mathbf{svec}(\Delta X)$ (before updating to X^+) in each iteration to regain feasibility by projection onto the null space of $\mathcal{A}$, i.e, replacing $\mathbf{svec}(\Delta X)$ by $[I - \mathcal{A}^T(\mathcal{A}\mathcal{A}^T)^{-1}\mathcal{A}]\mathbf{svec}(\Delta X)$. This step is inexpensive. To compute the projection, we only need to compute the Cholesky factorization of the $m \times m$ matrix $\mathcal{A}\mathcal{A}^T$ once at the beginning of the algorithm. (We use the Cholesky factorization here instead of a QR factorization to avoid using the **smat** and **svec** operations, which are expensive in MATLAB, in the Schur complement methods; for Algorithm NT-PC-QR an alternative would be to use the QR factorization of $K^T\mathcal{A}^T$ obtained in the first iteration to do the primal feasibility correction in all iterations.)

We compare the performance of Algorithm NT-PC-QR and the three methods just mentioned on the following classes of semidefinite programming problems.

(1) Random SDP. The starting point does not necessarily lie on the central path.

(2) Norm minimization problem:

$$\min_{x\in R^m} \|A_0 + \sum_{k=1}^{m} x_k A_k\|, \tag{49}$$

where the A_k, $k = 0, \ldots, m$, are real $N \times N$ matrices and the norm is the matrix 2-norm. It is well known that this problem can be expressed as an SDP involving $m + 2$ symmetric matrices of dimension $n \times n$, where $n = 2N$ [22]. For this problem, a feasible starting point is readily available, and we choose (X_0, y_0, S_0) to be feasible, but not necessarily lying on the central path.

(3) Chebyshev approximation problem for a matrix:

$$\min_{p} \|p(A)\|, \tag{50}$$

where the minimization is over the class of monic polynomials of degree m and the norm is the matrix 2-norm. Here A is a real $N \times N$ matrix. The Chebyshev problem (50) is a special case of the norm minimization problem and thus can be expressed as an SDP.

We should note that since the power basis $\{I, A, \ldots, A^m\}$ is highly ill-conditioned in general, it should be replaced by a better conditioned alternative for numerical stability. In our actual computation, we use the orthonormal basis $\{Q_1, \ldots, Q_{m+1}\}$, with respect to the inner product $G \bullet K$, obtained from the power basis via a modified Gram-Schmidt procedure. For details on how this change of basis transforms the original problem, see [21]. Again, we choose a feasible starting point for this problem.

(4) Max-Cut problem:

$$\begin{aligned} &\min\ L \bullet X \\ &\text{s.t.}\quad \operatorname{diag}(X) = e/4, \quad X \succeq 0, \end{aligned} \tag{51}$$

where $L = A - \operatorname{Diag}(Ae)$, e is the vector of all ones and A is the weighted adjacency matrix of a graph [7]. In our experiments, we only consider unweighted graphs where each edge is present independently with probability one half. We choose the following feasible starting point, where $\operatorname{abs}(L)$ denotes the matrix of absolute values of the entries of L:

$$X_0 \;=\; \operatorname{diag}(e/4), \quad y_0 \;=\; -1.1\operatorname{abs}(L)\, e, \quad S_0 \;=\; L - \operatorname{diag}(y_0).$$

(5) ETP (Educational testing problem):

$$\begin{aligned} &\max\ e^T d \\ &\text{s.t.}\quad A - \operatorname{Diag}(d) \succeq 0, \quad d \geq 0, \end{aligned} \tag{52}$$

where A is an $N \times N$ positive definite matrix. This problem can readily be expressed as an SDP of the form (2), involving symmetric matrices of dimension $n \times n$, where $n = 2N$. Again, a feasible starting point is chosen for this problem.

In our experiments, all the computations were performed in MATLAB [13]. For each class of the SDPs mentioned above, we solved ten instances with random data. That is, the given matrices are random, with entries chosen from the normal distribution with zero mean and unit variance. For the ETP problem, A is the matrix

product of such a random matrix and its transpose so that the resulting matrix is symmetric positive definite.

For each set of 10 instances, we compare the algorithms using the adaptive step-length strategy given in (48) and also using a fixed step-length parameter with $\tau = 0.98$. The parameter *expon* used in our experiments is as follows:

$$expon = \begin{cases} 3 & \text{for the AHO method,} \\ 1 & \text{for the H..K..M method,} \\ 1 & \text{for the NT methods.} \end{cases}$$

For each class, we plot the convergence curves of the duality gap of one of the instances. The plots are shown in Figures 1–5. The dashed curve corresponds to the AHO direction, the dotted curve corresponds to the H..K..M direction, and the solid curves correspond to the NT direction. For the most part, the two solid curves coincide with one another, indicating that the performances of Algorithm NT-PC-QR and NT-PC-Sch are about the same on most problems. This also shows that for the NT direction, solving the Schur complement equation by Cholesky factorization is as good as solving by QR factorization. Since the behavior of these two algorithms depends mainly on the properties of the NT direction rather than on how the direction is computed, we will not distinguish them in the subsequent discussion, but refer to them collectively as the NT methods.

In Tables 3 and 4, corresponding to the adaptive step-length parameter given in (48) and the fixed step-length parameter of 0.98, respectively, we give the average number of iterations and CPU time for each method to reduce the duality gap by a factor of 10^{10}. (Note that this is much better than a factor of $\epsilon_{\text{mach}}^{-1/2}$, which informal reasoning might suggest as the limit for methods based on the Schur complement approach.) The times given exhibit mild fluctuations due to other jobs being present on the machines being used, but the trends are clear.

A summary of our observations is given below.

1. For Algorithm AHO-PC-Sch, the achievable accuracy in the duality gap is typically 10 to 100 times smaller than that obtained from methods based on the H..K..M and NT directions. For the first four classes of SDPs we consider here, the duality gap can usually reach a level smaller than 10^{-10} to 10^{-12} for Algorithm AHO-PC-Sch. It is surprising that AHO-PC-Sch can achieve such an accuracy in the duality gap despite the fact that the condition number of the Schur complement matrix involved is of the order of the reciprocal of the duality gap.
2. Besides being the most accurate, Algorithm AHO-PC-Sch is also the most efficient (in terms of the number of iterations) on the sets of problems we consider here, but the NT methods are almost as efficient, typically requiring only 2 or 3 more iterations to achieve a given precision in the duality gap. The H..K..M method is also almost as efficient as the AHO method, except on the ETP problems.
3. However, each iteration of Algorithm AHO-PC-Sch is about twice as expensive as each iteration of the H..K..M or the NT methods. Among the latter three methods, each iteration of NT-PC-QR is slightly more expensive than those of NT-PC-Sch (but not always), which in turn is slightly more expensive than those of H..K..M-PC-Sch. Consequently, overall the H..K..M method takes the least average amount of CPU time to reduce the duality gap by 10^{10}, the NT methods follow closely, and the AHO method is the

slowest.

4. Though the NT methods are not the best in achieving high accuracy, they may be the most robust in the sense that problems of stagnation such as using very small step lengths seem less likely to occur. We have not observed stagnation problems with the NT methods in our numerical experiments, while the AHO method occasionally stagnates on the Chebyshev approximation problems for matrices, and the H..K..M method often encounters such difficulties on the ETP problems. This difficulty does not seem to be due to the Schur complement approach; using a QR implementation of the H..K..M method yields similar results.
5. Using an adaptive step-length parameter choice can alleviate stagnation problems that are encountered from using a fixed step-length parameter. The improvement is especially noticable for the H..K..M method in solving the ETP problems.

As we already mentioned in the introduction, the good performance of the methods that are based on the Schur complement approach was observed after the release of the first version of this paper, in which NT-PC-QR had been observed to be superior to the rest. We note that these improvements in the Schur-complement-based methods come from careful implementations; for example, the code now symmetrizes explicitly quantities that ought to be symmetric in exact arithmetic.

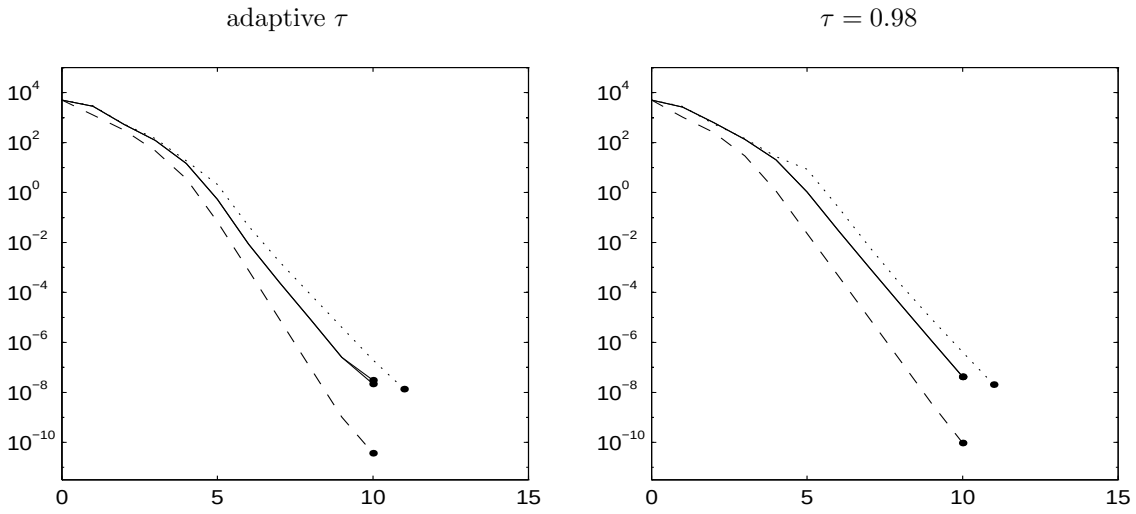

FIG. 1. *Convergence curves of the duality gaps of a random SDP problem for different methods. The dashed curve corresponds to the AHO direction, the dotted curve corresponds to the H..K..M direction, and the solid curves correspond to the NT direction.*

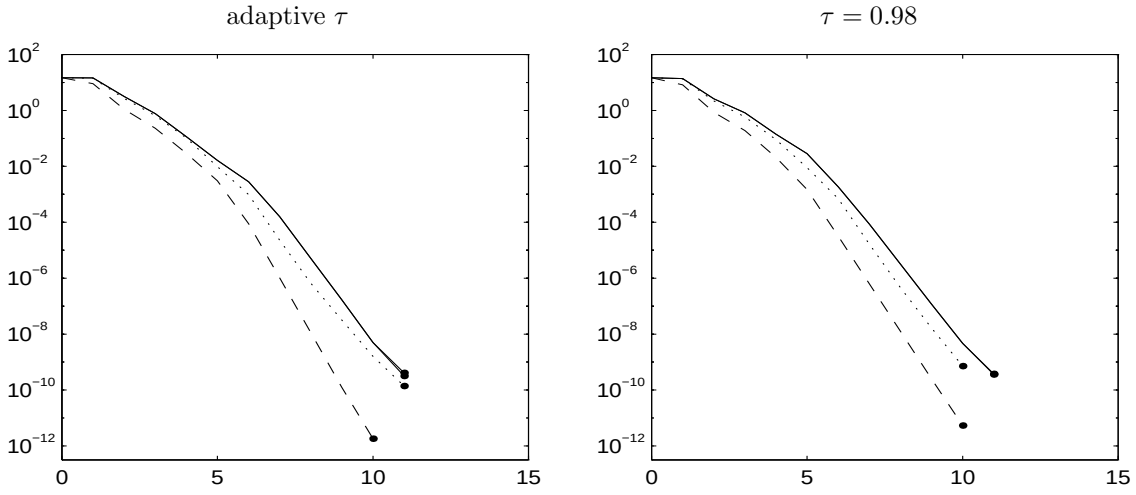

FIG. 2. *Same as Figure 1 but for a norm-minimization problem.*

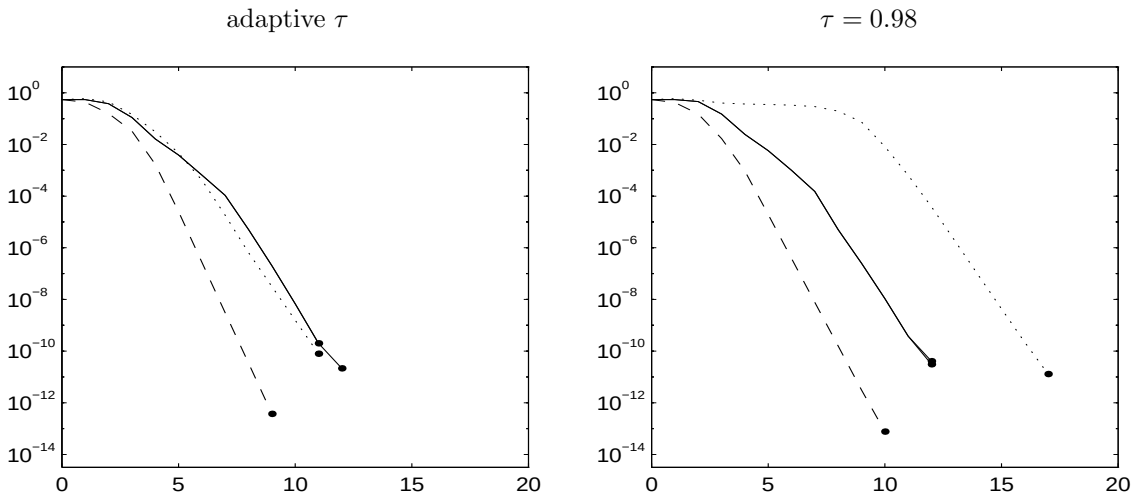

FIG. 3. *Same as Figure 1 but for a Chebyshev problem.*

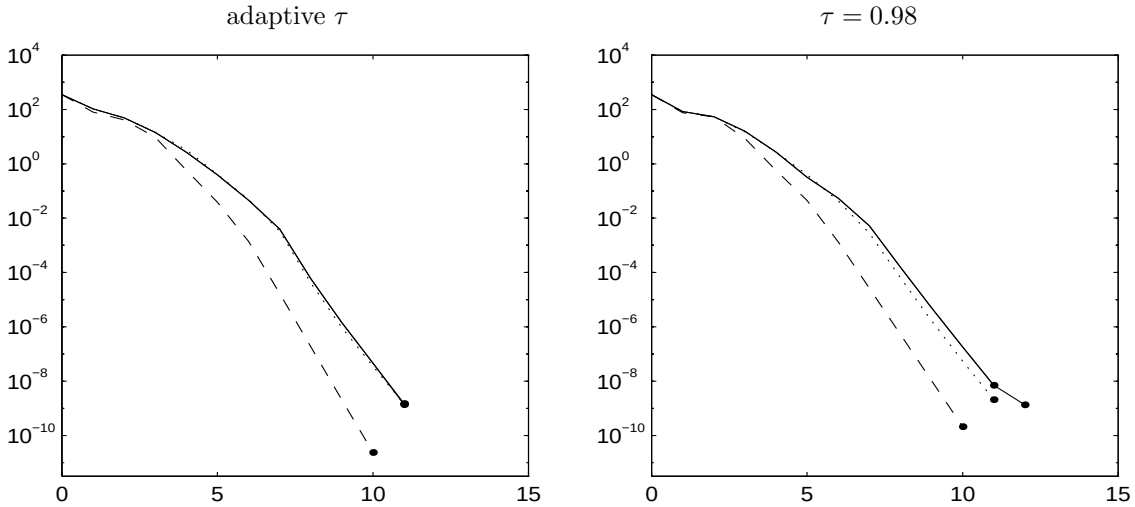

FIG. 4. *Same as Figure 1 but for a Max-Cut problem.*

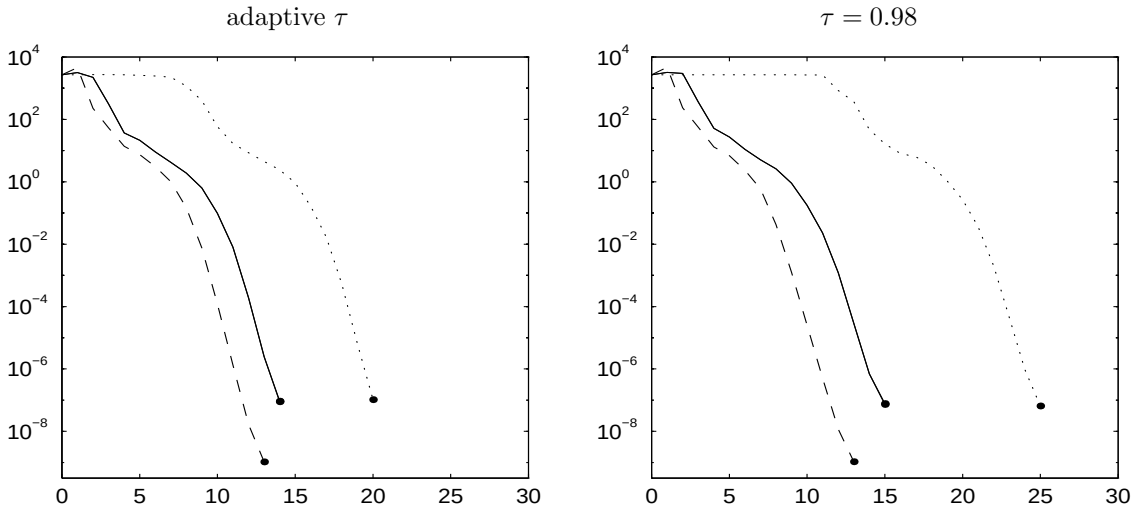

FIG. 5. *Same as Figure 1 but for an ETP problem. This class of SDP problems proves to be more difficult for the H..K..M direction.*

adaptive τ examples		Average no. of iterations to reduce the duality gap by 10^{10}				Average CPU time (secs.) to reduce the duality gap by 10^{10}			
		AHO	H..K..M	NT	NT(QR)	AHO	H..K..M	NT	NT(QR)
Random SDP	$n = 100$ $m = 25$	9.2	10.2	9.8	9.8	30.7	22.4	20.6	22.7
Norm Minimization	$n = 100$ $m = 25$	9.0	11.0	11.2	11.2	90.8	59.8	67.1	71.5
Chebyshev	$n = 100$ $m = 25$	8.7	11.0	11.1	11.1	89.0	60.2	66.5	70.8
Max-Cut	$n = 50$ $m = 50$	9.3	11.0	11.0	11.0	22.7	13.9	14.0	20.1
ETP	$n = 100$ $m = 50$	13.4	20.2	15.8	15.8	70.8	45.3	38.7	44.2

TABLE 3

Computational results on different classes of SDPs. Ten instances with random data are considered for each class of the SDP. The computations were done on a DEC AlphaStation 500 (300 MHz). An adaptive step-length parameter choice is used in the algorithms.

$\tau = 0.98$ examples		Average no. of iterations to reduce the duality gap by 10^{10}				Average CPU time (secs.) to reduce the duality gap by 10^{10}			
		AHO	H..K..M	NT	NT(QR)	AHO	H..K..M	NT	NT(QR)
Random SDP	$n = 100$ $m = 25$	9.0	10.4	10.1	10.1	27.6	19.7	20.2	22.3
Norm Minimization	$n = 100$ $m = 25$	9.3	11.0	11.5	11.4	90.4	58.2	66.1	72.9
Chebyshev	$n = 100$ $m = 25$	9.4*	13.0	11.4	11.5	90.3*	68.2	66.4	72.7
Max-Cut	$n = 50$ $m = 50$	9.4	11.1	11.1	11.1	22.3	13.2	14.0	19.3
ETP	$n = 100$ $m = 50$	13.7	25.5	17.2	17.1	65.7	57.1	41.6	51.0

TABLE 4
Same as Table 3 but for fixed step-length parameter with $\tau = 0.98$.

* The AHO method fails on one of the instances due to step lengths going below 10^{-6}. The number reported here does not include this unsuccessful instance.

Appendix. We present some useful properties of the standard and symmetrized Kronecker products.

- Standard Kronecker product:
  1. $G \otimes K = [g_{ij}K]$.
  2. $(G \otimes K)\mathbf{vec}(H) = \mathbf{vec}(KHG^T)$ (where **vec** is defined below).
  3. $(G \otimes K)^T = G^T \otimes K^T$.
  4. $G \otimes I$ is symmetric iff G is.
  5. $(G \otimes K)^{-1} = G^{-1} \otimes K^{-1}$.
  6. $(G \otimes K)(H \otimes L) = GH \otimes KL$.
  7. If $\Lambda(G) = \{\lambda_i\}$ and $\Lambda(K) = \{\mu_j\}$, then $\Lambda(G \otimes K) = \{\lambda_i\mu_j\}$. If x_i and y_j are the eigenvectors corresponding to the eigenvalues λ_i and μ_j of G and K, then $\mathbf{vec}(y_j x_i^T)$ is the eigenvector corresponding to the eigenvalue $\lambda_i\mu_j$ of $G \otimes K$.
  8. $\mathbf{vec}(G)^T\mathbf{vec}(K) = G \bullet K$.
- Symmetrized Kronecker product:
  1. $(G \otimes_s K)\mathbf{svec}(H) = \frac{1}{2}\mathbf{svec}(KHG^T + GHK^T)$.
  2. $G \otimes_s K = K \otimes_s G$.
  3. $(G \otimes_s K)^T = G^T \otimes_s K^T$.
  4. $G \otimes_s I$ is symmetric iff G is.
  5. $(G \otimes_s G)^{-1} = G^{-1} \otimes_s G^{-1}$.
  6. $(G \otimes_s K)(H \otimes_s L) = \frac{1}{2}(GH \otimes_s KL + GL \otimes_s KH)$.
  7. $(G \otimes_s K)(H \otimes_s H) = (GH \otimes_s KH)$ and $(H \otimes_s H)(G \otimes_s K) = (HG \otimes_s HK)$.
  8. If $\Lambda(G) = \{\lambda_i\}$, then $\Lambda(G \otimes_s G) = \{\lambda_i\lambda_j\}$. If x_i and x_j are the eigenvectors corresponding to the eigenvalues λ_i, λ_j of G, then $\mathbf{svec}(\frac{1}{2}(x_i x_j^T + x_j x_i^T))$ is the eigenvector corresponding to the eigenvalue $\lambda_i\lambda_j$ of $G \otimes_s G$.
  9. Let G and K be commuting symmetric matrices with the common basis of eigenvectors $\{x_i\}$ and corresponding eigenvalues λ_i and μ_i, respectively. Then $\Lambda(G \otimes_s K) = \{\frac{1}{2}(\lambda_i\mu_j + \lambda_j\mu_i)\}$. Also, $\mathbf{svec}(\frac{1}{2}(x_i x_j^T + x_j x_i^T))$ is an eigenvector corresponding to the eigenvalue $\frac{1}{2}(\lambda_i\mu_j + \lambda_j\mu_i)$ of $G \otimes_s K$ [3].
  10. $\mathbf{svec}(G)^T\mathbf{svec}(K) = G \bullet K$.
  11. If G and K are symmetric and positive definite, then so is $G \otimes_s K$.

The properties of the symmetrized Kronecker product can be verified by relating it to the standard Kronecker product, and then using its properties. To this end, consider the $n(n+1)/2 \times n^2$ matrix U such that $U\mathbf{vec}(H) = \mathbf{svec}(H)$ and $U^T\mathbf{svec}(H) = \mathbf{vec}(H)$ for all $n \times n$ symmetric matrices H. Here

$$\mathbf{vec}(H) := (h_{11}, h_{21}, \ldots, h_{n1}, h_{12}, \ldots, h_{nn})^T.$$

If we label the rows of U in the order $(1,1), (2,1), \ldots, (n,1), (2,2), (3,2), \ldots, (n,2), (3,3), \ldots, (n,n)$ and its columns in the order $(1,1), (2,1), \ldots, (n,1), (1,2), \ldots, (n,2), (1,3), \ldots, (n,n)$, then

$$U_{(i,j),(k,l)} = \begin{cases} 1 & \text{if } i = j = k = l, \\ 1/\sqrt{2} & \text{if } i = k \neq j = l, \text{or } i = l \neq j = k, \\ 0 & \text{otherwise.} \end{cases}$$

For example, for $n = 3$,

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 0 & 0 & 0 & 1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Note that $U\,U^T = I_{n(n+1)/2}$. Although $U^T\,U \neq I_{n^2}$, we have that $U^TU\,\mathbf{vec}(H) = \mathbf{vec}(H)$ for every $n \times n$ symmetric matrix H. In fact, U^TU is the orthogonal projection matrix onto the space of symmetric matrices.

With the matrix U, $G \otimes_s K$ can be expressed in terms of the standard Kronecker products of G and K as follows:

$$G \otimes_s K = \frac{1}{2}\,U(G \otimes K + K \otimes G)U^T.$$

Some properties of the symmetrized Kronecker product can also be derived directly from the definition (15), which is listed as Property 1 above. For example, Property 6 holds because for any matrix M

$$\begin{aligned}(G \otimes_s K)(H \otimes_s L)\mathbf{svec}(M) &= \frac{1}{2}\,(G \otimes_s K)\mathbf{svec}(L\,M\,H^T + H\,M\,L^T) \\ &= \frac{1}{4}\,\mathbf{svec}(KL\,M\,H^TG^T + GH\,M\,L^TK^T \\ &\qquad + KH\,M\,L^TG^T + GL\,M\,H^TK^T) \\ &= \frac{1}{2}\,(GH \otimes_s KL + GL \otimes_s KH)\mathbf{svec}(M).\end{aligned}$$

Since M is arbitrary, the property follows.

REFERENCES

[1] F. Alizadeh, Combinatorial optimization with interior point methods and semi–definite matrices, Ph.D. thesis, University of Minnesota, Minneapolis, MN, 1991.

[2] ———, Semidefinite programming home page, `http://rutcor.rutgers.edu/~alizadeh/sdp.html`

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

[4] T. Ando, Concavity of certain maps and positive definite matrices and applications to Hadamard products, *Linear Algebra Appl.*, 26 (1979), pp. 203–241.

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

[6] C. Helmberg, Semidefinite programming home page, `http://www.zib.de/helmberg/semidef.html`

[7] ———, F. Rendl, R. Vanderbei, and H. Wolkowicz, An interior-point method for semidefinite programming, *SIAM J. Optim.*, 6 (1996), pp. 342–361.

[8] R. Horn, private communication, University of Utah, February 1996.

[9] M. Kojima, private communication, Tokyo Institute of Technology, April 1996.

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

[11] C.-J. Lin and R. Saigal, A predictor corrector method for semi-definite linear programming, manuscript (1995), Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI.

[12] S. Mehrotra, On the implementation of a primal-dual interior point method, *SIAM J. Optim.*, 2 (1992), pp. 575–601.

[13] The MathWorks, Inc., *MATLAB Reference Guide*, The MathWorks, Inc., Natick, MA, 1992.

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

[15] Yu. Nesterov and A. S. Nemirovskii, Conic formulation of a convex programming problem and duality, *Optimization Methods and Software*, 1(2) (1992), pp. 95–115.

[16] ———and ———, *Interior Point Polynomial Methods in Convex Programming : Theory and Algorithms*, SIAM Publications, SIAM, Philadelphia, PA, 1994.

[17] Yu. Nesterov and M. J. Todd, Self-scaled barriers and interior-point methods in convex programming, *Math. Oper. Res.*, 22 (1997), pp. 1–42.

[18] ———and ———, Primal-dual interior-point methods for self-scaled cones, TR–1125 (1995), School of Operations Research and *SIAM J. Optim.*, 8 (1998), pp. 324–364.

[19] M. Shida, S. Shindoh, and M. Kojima, Existence of search directions in interior-point algorithms for the SDP and the monotone SDLCP, *SIAM J. Optim.*, 8 (1998), pp. 387–396.

[20] J. F. Sturm and S. Zhang, Symmetric primal-dual path following algorithms for semidefinite programming, Report 9554/A (1995), Econometric Institute, Erasmus University, Rotterdam, The Netherlands.

[21] K. C. Toh and L. N. Trefethen, The Chebyshev polynomials of a matrix, manuscript (1996), Center for Applied Mathematics, Cornell University, Ithaca, NY (to appear in *SIAM J. Matrix Anal. Appl.*).

[22] L. Vandenberghe and S. Boyd, Semidefinite programming, *SIAM Rev.*, 38 (1996), pp. 49–95.

[23] Y. Zhang, On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming, *SIAM J. Optim.*, 8 (1998), pp. 365–386.

[24] ———, private communication, University of Maryland Baltimore County, March 1996.