On Primal-Dual Interior Point Methods
for
Semidefinite Programming

97-12

Ming Gu*

March 24, 1997

Abstract

The development of primal-dual interior-point methods for semidefinite programming problems has attracted much recent attention. In this paper, we discuss issues related to such methods based on the Monteiro and Zhang (MZ) family of search directions. This family includes a number of important search directions, such as the AHO direction of Alizadeh, Haeberly and Overton, two HKM directions proposed by several groups of researchers, and the NT direction of Nesterov and Todd.

These search directions are computed via the solution of a linear equation whose coefficient matrix is called the *Schur complement.* It is known that Schur complements associated with the HKM and NT directions are symmetric positive definite, making these directions about half as cheaper to compute as the AHO direction, whose associated Schur complement is in general non-symmetric.

Motivated by work of Todd, Toh and Tütüncü, we parameterize a subset in the MZ family to be referred to as the *TTT family.* The associated Schur complements for directions in the TTT family are symmetric positive definite. We discuss interesting properties of directions in this family and show that these directions include the HKM and the NT directions and can be computed as efficiently.

The AHO direction, the HKM directions and the NT direction can all be computed in several different ways. Although mathematically equivalent, it has been observed that these different ways often lead to *drastically* different accuracy in the numerical solution to the semidefinite programming problem. It has also been observed that, while more expensive at each iteration, the method based on the AHO direction tends to produce more accurate solutions than other methods.

We analyze the computation of the AHO direction and directions in the TTT family in finite precision arithmetic. Our analysis provides a theoretical explanation of the accuracy difference for different ways of computing the AHO direction and other directions. Our analysis also explains why the AHO direction tends to produce more accurate solutions. Several researchers have discussed the computation of the HKM and NT search directions via the solution of a least squares problem. Our analysis indicates that this approach, while more expensive than the Schur complement approach, does not in general provide more accuracy.

Most importantly, our analysis indicates that, with the Schur complement approach, methods based on the AHO direction and the TTT family of directions could be numerically stable if certain coefficient matrices associated with the search directions are well-conditioned, but unstable otherwise. We present results from our numerical experiments that support our analysis.

*Department of Mathematics, University of California, Los Angeles, CA 90095-1555. This research was supported in part by NSF Career Award CCR-9702866.

Contents

1 Introduction

The semidefinite programming problem (SDP) is the following convex optimization problem:

$$\begin{array}{ll} \min_{X\in \mathbf{S}^n} & C \bullet X \\ \text{subject to} & A_k \bullet X = b_k, \quad k = 1,\cdots,m, \\ & X \succeq 0, \end{array} \tag{1.1-a}$$

where $\mathbf{S}^n$ is the vector space of real symmetric n-by-n matrices; $A \bullet B$ is an inner product satisfying

$$A \bullet B \overset{\text{def}}{=} \mathbf{tr}(A^T \cdot B) = \sum_{i,j=1}^{n} A_{ij} \cdot B_{ij} \quad \text{for} \quad A, B \in \mathbf{R}^{n\times n} \ ;$$

$C \in \mathbf{S}^n$; and $A_k \in \mathbf{S}^n$ for $k = 1, \cdots, m$. By $X \succeq 0$ we mean that X is positive semidefinite. The *dual* problem to (1.1-a) is of the form

$$\begin{array}{ll} \max_{y\in\mathbf{R}^m, Z\in \mathbf{S}^n} & b^T y \\ \text{subject to} & \sum_{k=1}^m y_k A_k + Z = C, \\ & Z \succeq 0, \end{array} \tag{1.1-b}$$

where $b = (b_1, \cdots, b_m)^T \in \mathbf{R}^m$. It is known that (1.1-b) can be reformulated as an SDP.

The SDP arises in many areas of science and engineering and includes the linear programming problem (LP) as a special case (see Vandenberghe and Boyd [28]). The recent book by Boyd, Ghaoui, Feron, and Balakrishnan [4] and survey articles by Alizadeh [2], Lewis and Overton [14], and Vandenberghe and Boyd [28] contain many applications in system and control theory, combinatorial optimization and eigenvalue optimization.

Let **svec** be an isometry identifying $\mathbf{S}^n$ with $\mathbf{R}^{n(n+1)/2}$, so that

$$K \bullet L = (\mathbf{svec}(K))^T \cdot \mathbf{svec}(L)$$

for all K, $L \in \mathbf{S}^n$; and let **smat** be the inverse of **svec**. The optimality conditions for problem (1.1) are

$$\begin{aligned} \mathcal{A}\,\mathbf{svec}(X) &= b & \text{(1.2-a)} \\ \mathbf{smat}\left(\mathcal{A}^T\, y\right) + Z &= C & \text{(1.2-b)} \\ X\, Z &= 0\,, & \text{(1.2-c)} \end{aligned}$$

where X and $Z \succeq 0$; $\mathcal{A} = (\mathbf{svec}(A_1), \cdots, \mathbf{svec}(A_m))^T$; and $y = (y_1, \cdots, y_m)^T$. Throughout this paper, we assume that $\mathbf{rank}(\mathcal{A}) = m$ and that equations (1.2) have a *unique* solution (X^*, Z^*, y^*). Hence, (X^*, Z^*, y^*) is a feasible solution to (1.1) that further satisfies the complementarity condition (1.2-c), which requires that X^* and Z^* be commutable. Consequently, there exist an orthogonal matrix Q^* and non-negative diagonal matrices Λ_1^* and Λ_2^* with dimensions r-by-r and $(n-r)$-by-$(n-r)$, respectively, for an integer r, such that

$$X^* = Q^*\ \mathbf{diag}(\Lambda_1^*, 0)\ (Q^*)^T \quad \text{and} \quad Z^* = Q^*\ \mathbf{diag}(0, \Lambda_2^*)\ (Q^*)^T\ .$$

1.1 Interior-Point Methods for the SDP

Interior point methods typically produce a sequence of iterates which converge to the optimal solution. From a computational point of view, these methods differ mainly in the search directions used to determine the iterates.

Interior point methods for SDP were originally proposed by Alizadeh [1] and Nesterov and Nemirovskii [22] and have been an active area of research since then. Since it is generally accepted that primal-dual methods are the most successful interior-point methods for LP, most of these methods and search directions are based on the primal-dual formulation. Alizadeh [2] argued that many known interior point algorithms for LP can be transformed into algorithms for SDP in a mechanical way.

Most of the interior point methods for SDP are path-following methods, meaning that they generate a sequence of iterates approximating the so-called *central path* and converging to the primal and dual solutions. For SDP, the points on the central path satisfy the primal feasibility equation (1.2-a), the dual feasibility equation (1.2-b), and the following complementarity condition relaxed from (1.2-c):

$$X \cdot Z - \mu I = 0 \, . \tag{1.3}$$

It turns out that (1.3) is much more complex than its LP counterpart, making SDP more difficult.

It is well-known that under certain conditions the solution to (1.2-a), (1.2-b) and (1.3) is unique and converges to the optimal solution of (1.1) as μ goes to 0 (see Nesterov and Todd [23]). Helmberg, Rendl, Vanderbei and Wolkowicz [11], and Kojima, Shindoh and Hara [13] independently apply Newton's method to equations (1.2-a), (1.2-b) and (1.3) to obtain a method to be referred to as the XZ method. Since the matrix $X \cdot Z$ is in general not symmetric, the XZ method produces non-symmetric search directions. In fact, the X iterates are in general not symmetric and have to be explicitly symmetrized at every iteration. In our numerical experiments, we observed that the XZ method sometimes converges extremely slowly. On the other hand, this symmetrization issue does not arise in LP, since the LP counterparts of X and Z are diagonal matrices and hence $X \cdot Z$ is always symmetric. See Wright [31, Ch. 5] for a detailed discussion on primal-dual path-following methods for LP.

Several methods have been introduced in the literature to ensure a symmetric search direction. For example, Zhang [32] defines a symmetrization operator

$$\mathbf{H}_P(M) \stackrel{\text{def}}{=} \frac{1}{2}\left(P\ M\ P^{-1} + \left(P\ M\ P^{-1}\right)^T\right)$$

for any given nonsingular matrix P, and shows that equation (1.3) is equivalent to

$$\mathbf{H}_P(X\ Z) - \mu\ I = 0 \tag{1.4}$$

for symmetric X and Z. Applying Newton's method to equations (1.2-a), (1.2-b) and (1.4) results in a family of symmetric search directions parameterized by P, usually referred to as the Monteiro and Zhang (MZ) family.

The MZ family includes a number of important symmetric search directions that were introduced earlier. The AHO method introduced by Alizadeh, Haeberly and Overton [3] is based on a direction that corresponds to $P = I$. Taking $P^T\,P = X^{-1}$ and $P^T\,P = Z$ result in the two directions suggested by Monteiro [18]. These directions are also equivalent to two special directions of the family of directions introduced by Kojima, Shindoh and Hara [13]; and the direction corresponding to $P^T\,P = Z$ was also suggested by Helmberg, Rendl, Vanderbei, and Wolkowicz [11]. We refer to methods based on these two directions as the HKM methods to reflect their history of discovery[1]. The NT method, suggested by Nesterov and Todd [23, 24], corresponds to a search direction defined by any P that satisfies

$$P^T\ P = R^{-1}\ \left(R\ Z\ R^T\right)^{\frac{1}{2}}\ R^{-T}\ , \tag{1.5}$$

where $R \in \mathbf{R}^{n \times n}$ is any matrix such that $R^T\,R = X$. Another family of symmetric search directions has been recently introduced by Monteiro and Tsuchiya [19].

Polynomial iteration complexity has been established for primal-dual path-following algorithms based on any direction in these three families. See Kojima, Shindoh, and Hara [13], Monteiro [18, 17], Monteiro and Tsuchiya [19, 20], Monteiro and Zhang [21], and Zhang [32].

[1] They are called the H..K..M directions in [27].

1.2 Computational Issues

Since we are primarily concerned with computational issues in this paper, from now on we will not make clear distinctions between interior-point methods and their search directions.

The reason interior point methods attract so much attention is because they appear to have remarkable computational promise. Recently, attempts have been made to implement interior point methods for SDP. Alizadeh, Haeberly and Overton [3] implemented and compared the XZ method, the AHO method and the NT method; and Todd, Toh and Tütüncü [27] implemented and compared the AHO method, the NT method, and the HKM method corresponding to $P^T P = Z$. An SDP solver coded in C^{++} was developed by Fujisawa, Kojima and Nakata and is available in the public domain [9]. This solver includes three types of search directions: the AHO direction, the NT direction, and the HKM direction with $P^T P = Z$. Mehrotra predictor-corrector modifications [15] were exploited in these studies to accelerate convergence.

These studies revealed a number of computational issues for SDP that are surprisingly more complicated than those for LP. It is observed that for these interior point methods, some implementations were capable of yielding solutions in relatively good agreement with the true optimal solution, whereas others, being slightly different but mathematically equivalent in exact arithmetic, yielded very limited accuracy in the computed solution and sometimes even failed to converge. It is also observed that the AHO method of [3] appeared to be the most accurate among the methods tested [3, 27].

The search directions of these methods are usually solved via a Schur complement equation (see §2.1). The Schur complement is non-symmetric for the AHO method but symmetric positive definite for the other methods tested. As a consequence, one AHO iteration is roughly twice as costly as an iteration generated by these other methods [27].

Todd, Toh and Tütüncü [27] showed that for a subfamily of search directions in the MZ family, the Schur complement is symmetric positive definite, and the Schur complement equation can be expressed as the normal equation of a linear least squares (LS) problem and thus can be solved instead as an LS problem. Throughout this paper we refer to this subfamily as the TTT family. It includes the two HKM directions and the NT direction. Zhang [32] also discussed the LS approach for the HKM direction corresponding to $P^T P = Z$. Although their numerical results indicated that the two approaches seemed to be compatible in terms of accuracy, Todd, Toh and Tütüncü [27] argued that the LS approach could perform much better than the Schur complement approach in certain cases since the condition number of the coefficient matrix involved in the LS problem is the square root of that of the Schur complement.

Computational issues have been discussed earlier for other barrier methods and interior-point methods in optimization. For example, Ponceleón [25] analyzed linear systems arising from barrier methods for quadratic programming. Forsgren, Gill and Shinnerl [8] analyzed linear systems arising from interior methods for constrained optimization. And Wright [29, 30] analyzed interior-point methods for LP and linear complementarity problems.

1.3 Main Results

In this paper, we explicitly parameterize the TTT family of search directions. We discuss interesting properties of directions in the TTT family. We show that these directions include the NT and the HKM directions and can be computed as efficiently.

We analyze the accuracy of the AHO method and methods based on directions in the TTT family in finite precision arithmetic. We explain why some implementations of these methods are more accurate than others and why the LS approach in general does not perform better than the Schur complement approach. Most importantly, we show that, with the Schur complement approach, methods based on the AHO direction and the TTT family of directions could be numerically stable if certain coefficient matrices associated with the search direction are well-conditioned, but unstable otherwise. We present results from our numerical experiments that support our analysis.

Our error analysis is on the accuracy in the computed search direction for *one* step of the interior point methods at a point (X, Z, y) that is "close" to the optimal solution of (1.1). We avoid the discussion on the iteration complexity of these methods in finite precision partly because in practice most interior point methods with known polynomial iteration complexity do not perform as efficiently as those without. Wright [29, 30] took a somewhat similar approach in his finite precision analysis of interior-point methods for LP and linear complementarity problems.

This paper is organized as follows. In §2 we discuss the Schur complement equation and the parameterization of the TTT family. In §3 we discuss the AHO method and analyze it in finite precision. In §4 we develop methods of the TTT family, relate them to the NT and HKM methods, and analyze them in finite precision. In §5 we present results from our numerical experiments that support our analysis. And in §6 we discuss some extensions and future work.

1.4 Notation and Conventions

Throughout this paper, the symmetrized Kronecker product of G and K is a square matrix of order $n(n+1)/2$; its action on $\mathbf{svec}(H)$, where $H \in \mathrm{S}^n$, is given by

$$(G \otimes_s K)\ \mathbf{svec}(H) \overset{\mathrm{def}}{=} \frac{1}{2}\mathbf{svec}\left(K\ H\ G^T + G\ H\ K^T\right) . \tag{1.6}$$

The appendices in [3] and [27] contain some frequently used properties of the symmetrized Kronecker products in the context of SDP. They include

$$G \otimes_s K = K \otimes_s G\ , \quad (G \otimes_s K)^T = G^T \otimes_s K^T \quad \text{and} \quad (G \otimes_s G)^{-1} = G^{-1} \otimes_s G^{-1}\ .$$
$$(G \otimes_s K)\ (H \otimes_s H) = (G\ H) \otimes_s (K\ H) \quad \text{and} \quad (H \otimes_s H)\ (G \otimes_s K) = (H\ G) \otimes_s (H\ K)\ .$$

We will need the following vector from time to time:

$$\mathbf{e} \overset{\mathrm{def}}{=} \mathbf{svec}\begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix} \in \mathbf{R}^{n(n+1)/2}\ .$$

A *flop* is a floating point operation $\alpha \circ \beta$, where α and β are floating point numbers and $\circ$ is one of $+, -, \times$, and $\div$. In our error analysis, we take the usual model of arithmetic:

$$\mathbf{fl}(\alpha \circ \beta) = (\alpha \circ \beta)\ (1 + \xi)\ , \tag{1.7}$$

where $\mathbf{fl}(\alpha \circ \beta)$ is the floating point result of the operation $\circ$ and $|\xi| \leq \epsilon$, with ϵ being the machine precision. For simplicity, we ignore the possibility of overflow and underflow.

Let α and β be numbers. We write $\alpha = O(\beta)$ if $|\alpha| \leq c\ |\beta|$ for a modest positive *constant* c. We say that a matrix or a vector is $O(\alpha)$ if its norm is $O(\alpha)$. We write $\alpha = \Omega(\beta)$ if $\alpha = O(\beta)$ and $\beta = O(\alpha)$.

For any matrix X, $|X|$ is the matrix with entries $(|X|)_{ij} = |X_{ij}|$; and $|X| \leq |Y|$ means that $|X_{ij}| \leq |Y_{ij}|$ holds for all i and j. $\sigma_{\max}(X)$ and $\sigma_{\min}(X)$ are the largest and smallest singular value of X, respectively; and $\kappa(X) = \sigma_{\max}(X)/\sigma_{\min}(X) \geq 1$ is its condition number. For any *symmetric* matrix X, $\lambda_{\max}(X)$ and $\lambda_{\min}(X)$ are its largest and smallest eigenvalue, respectively. We say that X is positive definite if it is both symmetric and positive definite.

2 The Schur Complement and the TTT Family

Primal-dual methods are Newton-like methods applied to optimality equations (1.2). At a given point (X, Z, y), where X and Z are positive definite, the search direction (dX, dZ, dy) satisfies

$$\mathcal{A}\,\mathbf{svec}(dX) = r_p \tag{2.1-a}$$

$$\mathcal{A}^T\, dy + \mathbf{svec}(dZ) = r_d\,, \tag{2.1-b}$$

$$\text{where} \quad r_p = b - \mathcal{A}\cdot\mathbf{svec}(X) \quad \text{and} \quad r_d = \mathbf{svec}\left(C - Z - \mathbf{smat}\left(\mathcal{A}^T\cdot y\right)\right)\ .$$

For the MZ family, equation (1.4) is linearized to give

$$\mathbf{H}_P\,(dX\ Z + X\ dZ) = \mu I - \mathbf{H}_P\,(X\ Z)\ . \tag{2.1-c}$$

2.1 The Schur Complement

To solve for the search direction (dX, dZ, dy), we write equations (1.1) in a single 3×3 block equation (cf. Todd, Toh and Tütüncü [27] and Zhang [32])

$$\mathcal{J}\, d\mathcal{X} = \mathcal{R}\,, \qquad \mathcal{J} = \begin{pmatrix} \mathcal{E} & \mathcal{F} & 0 \\ 0 & \mathcal{I} & \mathcal{A}^T \\ \mathcal{A} & 0 & 0 \end{pmatrix},\ d\mathcal{X} = \begin{pmatrix} \mathbf{svec}(dX) \\ \mathbf{svec}(dZ) \\ dy \end{pmatrix},\ \mathcal{R} = \begin{pmatrix} r_c \\ r_d \\ r_p \end{pmatrix}, \tag{2.2}$$

where $\mathcal{I}$ is the identity matrix of appropriate dimension and

$$\mathcal{E} = \left(P^{-T}\ Z\right)\otimes_s P\,, \quad \mathcal{F} = (P\ X)\otimes_s P^{-T} \quad \text{and} \quad r_c = \mathbf{svec}\,(\mu I - \mathbf{H}_P\,(X\ Z))\ .$$

A straightforward way to compute the search direction $d\mathcal{X}$ is to solve (2.2) as a dense linear system of equations. However, this approach is too expensive for large SDPs. To compute $d\mathcal{X}$ more efficiently by taking advantage of the block structure in (2.2), we perform a block LU factorization on (2.2) to get

$$\begin{pmatrix} \mathcal{I} & 0 & 0 \\ 0 & \mathcal{I} & 0 \\ \mathcal{A}\,\mathcal{E}^{-1} & -\mathcal{A}\,\mathcal{E}^{-1}\,\mathcal{F} & \mathcal{I} \end{pmatrix}\cdot\begin{pmatrix} \mathcal{E} & \mathcal{F} & 0 \\ 0 & \mathcal{I} & \mathcal{A}^T \\ 0 & 0 & \mathcal{M} \end{pmatrix}\cdot d\mathcal{X} = \mathcal{R}\,, \tag{2.3}$$

where

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

is the Schur complement. Todd, Toh and Tütüncü [27] showed that $\mathcal{E}$ is non-singular under the assumption that both X and Z are positive definite. Apply forward block substitution to (2.3) to get

$$\begin{pmatrix} \mathcal{E} & \mathcal{F} & 0 \\ 0 & \mathcal{I} & \mathcal{A}^T \\ 0 & 0 & \mathcal{M} \end{pmatrix}\cdot\begin{pmatrix} \mathbf{svec}(dX) \\ \mathbf{svec}(dZ) \\ dy \end{pmatrix} = \begin{pmatrix} r_c \\ r_d \\ r_p + \mathcal{A}\,\mathcal{E}^{-1}\,(\mathcal{F}\,r_d - r_c) \end{pmatrix}, \tag{2.4}$$

and apply block backward substitution to (2.4) to get (cf. Zhang [32])

$$\mathcal{M}\ dy = r_p + \mathcal{A}\,\mathcal{E}^{-1}\,(\mathcal{F}\,r_d - r_c) \tag{2.5-a}$$

$$dZ = \mathbf{smat}\left(r_d - \mathcal{A}^T\ dy\right) \tag{2.5-b}$$

$$dX = \mathbf{smat}\left(\mathcal{E}^{-1}\,(r_c - \mathcal{F}\ \mathbf{svec}(dZ))\right)\ . \tag{2.5-c}$$

We now briefly discuss how to solve (2.5) efficiently, under the assumption that P is a general matrix and no information about its possible relation to (X, Z, y) is known. The matrix-vector products $\mathcal{F}\,u$ for $u = r_d$ and $u = \mathbf{svec}(dZ)$ on the right hand sides of (2.5) are

$$\mathcal{F}\;u = \frac{1}{2}\mathbf{svec}\left((P\;X)\;\mathbf{smat}(u)\;P^{-1} + P^{-T}\;\mathbf{smat}(u)\;(P\;X)^T\right)\;. \tag{2.6}$$

Note that $\mathcal{E}^{-1}$ appears in $\mathcal{M}$ and on the right hand sides of (2.5). Since $\mathcal{E}$ is an $n(n+1)/2$-by-$n(n+1)/2$ matrix, explicitly computing $\mathcal{E}$ can be very expensive. However, the expressions $\mathcal{E}^{-1}\;(\mathcal{F}\,r_d - r_c)$ and $\mathcal{E}^{-1}\;(r_c - \mathcal{F}\,\mathbf{svec}(dZ))$ in (2.5) can be computed through two linear systems of equations of the form

$$\mathcal{E}\;u = v \tag{2.7-a}$$

with right hand sides $v = \mathcal{F}\,r_d - r_c$ and $v = r_c - \mathcal{F}\,\mathbf{svec}(dZ)$, respectively. The Schur complement matrix $\mathcal{M}$ can be computed in a similar way: we first explicitly compute the $n(n+1)/2$-by-m matrix $\mathcal{F}\,\mathcal{A}^T$, and then compute $\mathcal{E}^{-1}\,\mathcal{F}\,\mathcal{A}^T$ by solving m linear systems of equations of the form (2.7-a). To solve (2.7-a), we first rewrite it in matrix form as

$$P\;U\;Z\;P^{-1} + P^{-T}\;Z\;U\;P^T = 2\,V\;, \tag{2.7-b}$$

where $U = \mathbf{smat}(u)$ and $V = \mathbf{smat}(v)$ (see (1.6) and (2.2)). By setting

$$\widetilde{U} = P\;U\;P^T \quad \text{and} \quad \widetilde{Z} = P^{-T}\;Z\;P^{-1}\;,$$

we can rewrite (2.7-b) as

$$\widetilde{U}\;\widetilde{Z} + \widetilde{Z}\;\widetilde{U} = 2\,V\;.$$

This last equation is a Lyapunov equation with a positive definite coefficient matrix $\widetilde{Z}$. Hence the solution $\widetilde{U}$ can be efficiently computed via the eigendecomposition of $\widetilde{Z}$ (see §3 and §4) and U can be computed from $\widetilde{U}$ as $U = P^{-1}\,\widetilde{U}\,P^{-T}$. Most of the work in solving (2.5) is in the formation and factorization of $\mathcal{M}$.

For the AHO method in §3 and the methods in the TTT family in §4, additional information about P and its relation to the current iterate (X, Z, y) is known. We will discuss more efficient solutions to (2.5) for these methods in §3 and §4, respectively.

2.2 The TTT Family

The Schur complement matrix $\mathcal{M}$ is not symmetric in general. Todd, Toh and Tütüncü [27] considered the family of search directions for which $\mathcal{E}^{-1}\mathcal{F}$ is symmetric. We refer to this family as the TTT family. Lemma 2.1 below combines some of the results in Theorem 3.1 and Proposition 3.1 of [27].

Lemma 2.1 (Todd, Toh and Tütüncü [27]) *Matrix $\mathcal{E}^{-1}\,\mathcal{F}$ is symmetric if and only if $P\,X\,Z\,P^{-1}$ is. Assume that $\mathcal{E}^{-1}\,\mathcal{F}$ is symmetric and $\mathbf{H}_P\,(X\;Z)$ is positive semidefinite, then the Schur complement matrix $\mathcal{M}$ is symmetric positive definite and the system of equations (2.2) has a unique solution.*

In the following we parameterize the TTT family. Let

$$X = R^T\;R \quad \text{and} \quad Z = H^T\;H \tag{2.8}$$

be decompositions of X and Z, respectively. They can be computed via the Cholesky factorizations or the eigendecompositions of X and Z. Let

$$R\;H^T = W\;\Sigma\;V^T \tag{2.9}$$

be the SVD of $R\,H^T$. Assume that $R\,H^T$ has k distinct singular values $\sigma_1 < \cdots < \sigma_k$ and that W and V are chosen such that $\Sigma = \mathbf{diag}\,(\sigma_1 I_1, \cdots, \sigma_k I_k)$ is a block diagonal matrix with distinct diagonal blocks.

By Lemma 2.1, $\mathcal{E}^{-1}\,\mathcal{F}$ is symmetric if and only if

$$P\,X\,Z\,P^{-1} = P^{-T}\,Z\,X\,P^T\,,$$

which simplies to

$$\left(P^T\,P\right)\,\left(R^T\,R\right)\,\left(H^T\,H\right) = \left(H^T\,H\right)\,\left(R^T\,R\right)\,\left(P^T\,P\right)\,.$$

This last equation can be rewritten as

$$\left(H^{-T}\,P^T\,P\,H^{-1}\right)\cdot\left(H\,R^T\,R\,H^T\right) = \left(H\,R^T\,R\,H^T\right)\cdot\left(H^{-T}\,P^T\,P\,H^{-1}\right)\,.$$

Plugging in (2.9) and simplifying, we have

$$\left(P\,H^{-1}\,V\right)^T\cdot\left(P\,H^{-1}\,V\right)\cdot\Sigma^2 = \Sigma^2\cdot\left(P\,H^{-1}\,V\right)^T\cdot\left(P\,H^{-1}\,V\right)\,.$$

In other words, the two positive definite matrices $(P\,H^{-1}\,V)^T\cdot(P\,H^{-1}\,V)$ and Σ^2 commute. It follows that there exists a non-singular block diagonal matrix $B = \mathbf{diag}(B_1, \cdots, B_k)$, where the dimension of B_j is that of I_j for $j = 1, \cdots, k$, such that

$$\left(P\,H^{-1}\,V\right)^T\cdot\left(P\,H^{-1}\,V\right) = B^T\,B\,.$$

Consequently,

$$\begin{aligned} P &= S\,B\,\widetilde{H}\,, && \text{where } S \in \mathbf{R}^{n\times n} \text{ is an orthogonal matrix and } \widetilde{H} = V^T\,H; && \text{(2.10-a)}\\ &= S\,\widetilde{B}\,\widetilde{R}^{-T}\,, && \text{where } \widetilde{R} = W^T\,R \text{ and } \widetilde{B} = B\,\Sigma. && \text{(2.10-b)}\end{aligned}$$

Equation (2.10-b) is equivalent to (2.10-a) because equation (2.9) implies

$$\widetilde{R}\,\widetilde{H}^T = \Sigma\,. \tag{2.11}$$

The amount of freedom in choosing P depends on the distribution of the singular values of $R\,H^T$. For example, if X and Z are *on* the central path (see (1.3)), then all the singular values are identical, and B, and hence P, can be chosen to be any non-singular matrix. In general, all the singular values are distinct, and B must be a non-singular diagonal matrix. The search direction in (2.2) is determined by B only; the orthogonal matrix S in (2.10) leaves (2.2) invariant.

The two HKM search directions [11, 13, 18] $P^T\,P = X^{-1}$ and $P^T\,P = Z$ are members in the TTT family with $B = \Sigma^{-1}$ and $B = I$, respectively; and the NT direction [23, 24] (see (1.5)) is a member in the TTT family with $B = \Sigma^{-\frac{1}{2}}$. Let P satisfy (2.10-a). Then

$$\begin{aligned} P\,X\,Z\,P^{-1} &= \left(S\,B\,V^T\,H\right)\,\left(R^T\,R\,H^T\,H\right)\,\left(H^{-1}\,V\,B^{-1}\,S^T\right)\\ &= S\,B\,V^T\,V\,\Sigma^2\,V^T\,V\,B^{-1}\,S^T = S\,\Sigma^2\,S^T\,.\end{aligned}$$

Hence

$$\mathbf{H}_P\,(X\,Z) = S\,\Sigma^2\,S^T \tag{2.12}$$

is a symmetric positive definite matrix. It follows from Lemma 2.1 that $\mathcal{M}$ is positive definite and (2.2) has a unique solution for any member of the TTT family.

Let $B_j = S_j D_j V_j^T$ be the SVD of B_j for $j = 1, \cdots, k$. Then SVD (2.9) can be rewritten as

$$R\,H^T = W\,\Sigma\,V^T = (W\,\mathbf{diag}(V_1, \cdots, V_k))\;\Sigma\;(V\,\mathbf{diag}(V_1, \cdots, V_k))^T\,.$$

Similarly, equations (2.10) can be rewritten as

$$\begin{aligned} P &= (S\ \mathbf{diag}(S_1,\cdots,S_k))\cdot \mathbf{diag}(D_1,\cdots,D_k)\cdot \widetilde{H}\ , & \widetilde{H} = (V\ \mathbf{diag}(V_1,\cdots,V_k))^T\ H \\ &= (S\ \mathbf{diag}(S_1,\cdots,S_k))\cdot (\mathbf{diag}(D_1,\cdots,D_k)\ \Sigma)\cdot \widetilde{R}^{-T}, & \widetilde{R} = (W\ \mathbf{diag}(V_1,\cdots,V_k))^T\ R. \end{aligned}$$

These relations show that the block diagonal matrix B can always be made diagonal with a proper choice of the singular vector matrices of $R\,H^T$ in (2.9). We will discuss interior-point algorithms based on the TTT family of search directions in more detail in §4.

Now we briefly turn our attention to *scale-invariance*, an interesting property of interior point methods discussed by Nesterov and Todd [23, 24], Sturm and Zhang [26], and Todd, Toh and Tütüncü [27]. Let $U \in \mathbf{R}^{n\times n}$ be non-singular. "Scale" problem (1.1) by replacing X, C, Z, and each A_k by

$$\widetilde{X} = U^{-1}\,X\,U^{-T}\ ,\quad \widetilde{C} = U^T\,C\,U\ ,\quad \widetilde{Z} = U^T\,Z\,U \quad \text{and} \quad \widetilde{A}_k = U^T\,A_k\,U\ . \tag{2.13}$$

This "scaling" defines a new problem. If (X, Z, y) is feasible in problem (1.1), then $(\widetilde{X}, \widetilde{Z}, y)$ is feasible in the scaled problem with the same objective function values. Let the search direction (dX, dZ, dy) for (1.1) at a given point (X, Z, y) be defined as a function of (X, Z, y):

$$(dX, dZ, dy) = (\mathbf{dX}(X,Z,y), \mathbf{dZ}(X,Z,y), \mathbf{dy}(X,Z,y))\ .$$

This direction is called scale-invariant if the search direction for the scaled problem at $(\widetilde{X}, \widetilde{Z}, y)$ is similarly scaled:

$$\left(\mathbf{dX}(\widetilde{X},\widetilde{Z},y), \mathbf{dZ}(\widetilde{X},\widetilde{Z},y), \mathbf{dy}(\widetilde{X},\widetilde{Z},y)\right) = \left(U^{-1}\ \mathbf{dX}(X,Z,y)\,U^{-T}, U^T\ \mathbf{dZ}(X,Z,y)\,U, \mathbf{dy}(X,Z,y)\right).$$

A similar property holds for primal-dual methods for LP. In the MZ family, we assume that $P = \mathbf{P}(X,Z)$ has been chosen as certain function of X and Z.

Lemma 2.2 (Todd, Toh and Tütüncü [27]) *Assume that the function $P = \mathbf{P}(X,Z)$ satisfies*

$$\left(\mathbf{P}\left(\widetilde{X},\widetilde{Z}\right)\right)^T\ \mathbf{P}\left(\widetilde{X},\widetilde{Z}\right) = U^T\ \left(\mathbf{P}\left(X,Z\right)\right)^T\ \mathbf{P}\left(X,Z\right)\ U\ ,$$

where $\widetilde{X}$ and $\widetilde{Z}$ are defined in (2.13) for a non-singular matrix $U \in \mathbf{R}^{n\times n}$. Also assume that $\mathbf{H}_P\,(X\ Z)$ is positive definite. Then the direction that solves (2.2) is scale-invariant.

With Lemma 2.2, Todd, Toh and Tütüncü [27] showed that the NT direction and the two HKM directions are scale-invariant.

In the following we show that *every* member of the TTT family is scale-invariant. Let X and Z be decomposed as in (2.8). Then $\widetilde{X}$ and $\widetilde{Z}$ can be decomposed as

$$\widetilde{X} = \left(R\,U^{-T}\right)^T\ \left(R\,U^{-T}\right) \quad \text{and} \quad \widetilde{Z} = (H\ U)^T\ (H\ U)\ ,$$

respectively. Since $\left(R\,U^{-T}\right)\ (H\,U)^T = R\,H^T$, the SVD (2.9) for $(\widetilde{X},\widetilde{Z})$ is the same SVD for (X,Z). By equations (2.10), we have

$$R\ \left(\mathbf{P}\left(X,Z\right)\right)^T\ \mathbf{P}\left(X,Z\right)\ R^T = \left(R\ U^{-T}\right)\left(\mathbf{P}\left(\widetilde{X},\widetilde{Z}\right)\right)^T\ \mathbf{P}\left(\widetilde{X},\widetilde{Z}\right)\ \left(R\,U^{-T}\right)^T = W\ \widetilde{B}^T\ \widetilde{B}\ W^T\ .$$

Lemma 2.2 follows this relation and equation (2.12).

3 Analysis of the AHO Method

3.1 The AHO Method

The AHO method of [3] is a special case of (2.2) with $P = I$:

$$\mathcal{J}\, d\mathcal{X} = \mathcal{R}\,, \qquad \text{where} \quad \mathcal{J} = \begin{pmatrix} \mathcal{E} & \mathcal{F} & 0 \\ 0 & \mathcal{I} & \mathcal{A}^T \\ \mathcal{A} & 0 & 0 \end{pmatrix} \quad \text{and} \quad \mathcal{R} = \begin{pmatrix} r_c \\ r_d \\ r_p \end{pmatrix}, \tag{3.1}$$

with $\mathcal{E} = Z \otimes_s I$, $\mathcal{F} = X \otimes_s I$ and $r_c = \mathbf{svec}\left(\mu I - \frac{X\,Z + Z\,X}{2}\right)$.

The matrix-vector product (2.6) is

$$\mathcal{F}\,u = \frac{1}{2}\mathbf{svec}\,(X\;\mathbf{smat}(u) + \mathbf{smat}(u)\;X)\ .$$

Hence $\mathcal{F}\,u$ can be computed with just one matrix-matrix product, which costs about $2n^3$ flops (see Golub and van Loan [10, Ch. 1]). The matrix form of (2.7) is simply

$$U\,Z + Z\,U = 2\,V\,, \tag{3.2}$$

which is already a Lyapunov equation. To solve it, eigendecompose Z to get $Z = Q\,\Lambda\,Q^T$, where $Q \in \mathbf{R}^{n\times n}$ is orthogonal and $\Lambda = \mathbf{diag}(\lambda_1,\cdots,\lambda_n) \succeq 0$ is diagonal. This computation requires about $9n^3$ flops or less (see Demmel [7, Ch. 5] and Golub and van Loan [10, Ch. 8]). The solution to (3.2) is

$$U = Q\,\bar{U}\,Q^T \quad \text{where} \quad \bar{U} = \left(\frac{2\cdot\bar{V}_{ij}}{\lambda_i + \lambda_j}\right) \quad \text{with} \quad \bar{V} = Q^T\,V\,Q\,. \tag{3.3}$$

The cost for computing $\bar{V}$ from V is about $3n^3$ flops, taking into account symmetry in $\bar{V}$; the cost for computing $\bar{U}$ from $\bar{V}$ is about n^2 flops; and the cost for computing U from $\bar{U}$ is about $3n^3$ flops.

There are $m + 2$ equations of the form (3.2) in (2.5), all of which can be solved via the same eigendecomposition of Z. The total cost for eigendecomposing Z and solving these equations is about $6mn^3$ flops. Adding up the costs for computing $\mathcal{F}\,\mathcal{A}^T$ and computing $\mathcal{M}$ from $\mathcal{E}^{-1}\,\mathcal{F}\,\mathcal{A}^T$, the total cost for computing $\mathcal{M}$ is about $m^2n^2 + 8mn^3$ flops. If we assume that Gaussian elimination with partial pivoting, which is usually stable and costs about $2/3m^3$ flops, is used to factorize $\mathcal{M}$, then the total cost for solving (2.5) is about $2/3m^3 + m^2n^2 + 8mn^3$ flops. Algorithm 3.1 below describes the AHO method.

Algorithm 3.1 AHO Method

1. Choose $0 \le \sigma < 1$ and determine (dX, dZ, dy) from (3.1), using $\mu = \sigma \cdot \frac{X \bullet Z}{n}$.
2. Choose steplengths α and β and update the iterates by

$$(X, Z, y) \leftarrow (X + \alpha\,dX, Z + \beta\,dZ, y + \beta\,dy)\ .$$

The steplength rule is given by choosing a parameter $\tau \in (0,1)$ and defining, via the factorizations (2.8),

$$\alpha = \min\left(1, -\frac{\tau}{\lambda_{\min}\left(R^{-T}\,dX\,R^{-1}\right)}\right) \quad \text{and} \quad \beta = \min\left(1, -\frac{\tau}{\lambda_{\min}\left(H^{-T}\,dZ\,H^{-1}\right)}\right)\ . \tag{3.4}$$

The computation of α and β involves the computation of the factorizations (2.8) and the eigenvalues of $R^{-T}\,dX\,R^{-1}$ and $H^{-T}\,dZ\,H^{-1}$. The total cost for this computation is about $24n^3$ or less. Hence Algorithm 3.1 costs about $2/3m^3 + m^2n^2 + 8mn^3$ flops per step.

Mehrotra's predictor-corrector (PC) rule [15] is a very powerful technique to accelerate convergence and has been adopted for many of the interior point methods (see [31]). Alizadeh, Haeberly and Overton [3] extended this rule to the AHO method to get

Algorithm 3.2 AHO Method with Mehrotra PC Rule

1. Determine (dX, dZ, dy) from (3.1), using $\mu = 0$.
2. Choose steplengths α and β using (3.4), and define
$$\mu = \frac{X \bullet Z}{n} \cdot \left(\frac{(X + \alpha\, dX) \bullet (Z + \beta\, dZ)}{X \bullet Z} \right)^3 , \; r_c = \mathbf{svec}\left(\mu I - \frac{X\, Z + Z\, X + dX\, dZ + dZ\, dX}{2} \right) .$$
3. Redetermine (dX, dZ, dy) from (3.1), and α and β from (3.4). Update the iterates by
$$(X, Z, y) \leftarrow (X + \alpha\; dX, Z + \beta\; dZ, y + \beta\; dy) \; .$$

Similar to Algorithm 3.1, one step of Algorithm 3.2 costs about $2/3m^3 + m^2n^2 + 8mn^3$ flops. Alizadeh, Haeberly and Overton [3] recommend that Algorithm 3.1 be used with $\sigma = 0.25$ til the residual norms $\|r_p\|_2$ and $\|r_d\|_2$ become relatively small, in which case Algorithm 3.2 should be used. The value of τ is usually chosen to be between 0.9 and 0.999.

3.2 Preliminary Analysis

To prepare for our analysis of the AHO method in finite precision, in §3.2 we analyze the round-off errors in the solution to equation (3.2). Assume that the eigendecomposition of Z is computed as

$$Z = \widehat{Q}\, \widehat{\Lambda}\, \widehat{Q}^T + O\left(\epsilon \cdot \|Z\|_2\right) , \tag{3.5}$$

where $\widehat{Q}$ is a *nearly* orthogonal matrix satisfying $\widehat{Q}^T \widehat{Q} = I + O(\epsilon)$; and $\widehat{\Lambda} = \mathbf{diag}(\widehat{\lambda}_1, \cdots, \widehat{\lambda}_n)$ is a diagonal matrix. Then there exists an *exactly* orthogonal matrix $Q^\dagger$ such that $\widehat{Q} = Q^\dagger + O\left(\epsilon\right)$ (see Chandrasekaran and Ipsen [5]). It follows that

$$Z = Q^\dagger\, \widehat{\Lambda}\, \left(Q^\dagger\right)^T + O\left(\epsilon \cdot \|Z\|_2\right) \overset{\text{def}}{=} Z^\dagger + O\left(\epsilon \cdot \|Z\|_2\right) . \tag{3.6}$$

Note that $Z^\dagger = Q^\dagger \widehat{\Lambda} \left(Q^\dagger\right)^T$ is an exact eigendecomposition. We further let

$$\mathcal{E}^\dagger = Z^\dagger \otimes_s I = \mathcal{E} + O\left(\epsilon \cdot \|Z\|_2\right) \quad \text{and} \quad \mathcal{M}^\dagger = \mathcal{A}\, \left(\mathcal{E}^\dagger\right)^{-1}\, \mathcal{F}\, \mathcal{A}^T . \tag{3.7}$$

Lemma 3.1 below is the basis of our error analysis in §3.3. We leave its proof in Appendix A.

Lemma 3.1 *Assume that equation (2.7-a) for $\mathcal{E} = Z \otimes_s I$ is solved as in §3.1. Then*

$$\mathbf{fl}\left(\mathcal{E}^{-1}\, v\right) = \mathbf{svec}(\mathbf{fl}\left(U\right)) = (\mathcal{I} + \Delta_2) \cdot \left(\mathcal{E}^\dagger\right)^{-1}\, (\mathcal{I} + \Delta_3)\; v\; , \tag{3.8}$$

where $\Delta_2 = O(\epsilon)$ and $\Delta_3 = O(\epsilon)$ are $n(n+1)/2$-by-$n(n+1)/2$ perturbation matrices.

It is important to note that the matrix $\mathcal{E}^\dagger$ does *not* depend on v. For different right hand sides v, the corresponding numerical solutions in (3.8) will in general have different perturbation matrices Δ_2 and Δ_3, but always the same $\mathcal{E}^\dagger$.

3.3 Error Analysis for the AHO Method

In Algorithm 3.1, $d\mathcal{X}$ is computed using (2.5). Round-off errors are in general made at every step of the computation. Let

$$\widehat{d\mathcal{X}} = \begin{pmatrix} \mathbf{svec}(\widehat{dX}) \\ \mathbf{svec}(\widehat{dZ}) \\ \widehat{dy} \end{pmatrix} \quad \text{and} \quad \widehat{\mathcal{R}} = \begin{pmatrix} \widehat{r}_c \\ \widehat{r}_d \\ \widehat{r}_p \end{pmatrix}$$

be the computed versions of $d\mathcal{X}$ of $\mathcal{R}$, respectively. In our analysis, we will put every round-off error in computing $\mathcal{R}$ into $\widehat{\mathcal{R}}$, and put every round-off error in solving (2.5) from $\widehat{\mathcal{R}}$ into the backward error in $\mathcal{J}$. We will show in §3.3 that there exists a perturbation matrix $\delta\mathcal{J}$ such that

$$(\mathcal{J} + \delta\mathcal{J})\ \widehat{d\mathcal{X}} = \widehat{\mathcal{R}}\,. \tag{3.9}$$

It follows from standard perturbation theory (see, for example, Demmel [7, Ch. 2]) that

$$\frac{\left\|\widehat{d\mathcal{X}} - d\mathcal{X}\right\|_2}{\|d\mathcal{X}\|_2} \le \frac{\kappa(\mathcal{J})}{1 - \kappa(\mathcal{J})\ \frac{\|\delta\mathcal{J}\|_2}{\|\mathcal{J}\|_2}} \left(\frac{\|\delta\mathcal{J}\|_2}{\|\mathcal{J}\|_2} + \frac{\left\|\widehat{\mathcal{R}} - \mathcal{R}\right\|_2}{\|\mathcal{R}\|_2}\right). \tag{3.10}$$

Since Algorithm 3.1 is an iterative method, it usually is not necessary for $d\mathcal{X}$ to be computed very accurately for the methods to make progress. However, if the round-off errors in $\widehat{\mathcal{R}}$ are so large that $\left\|\widehat{\mathcal{R}} - \mathcal{R}\right\|_2 = \Omega\left(\|\mathcal{R}\|_2\right)$, or if the backward errors in $\mathcal{J}$ are so large that $\|\delta\mathcal{J}\|_2 = \Omega\left(\sigma_{\min}(\mathcal{J})\right)$, then the right hand side of (3.10) becomes at least $\Omega(1)$ or even undefined, implying that the computed search direction $\widehat{d\mathcal{X}}$ could be *completely* different from $d\mathcal{X}$. In such cases it is likely that Algorithm 3.1 will *not* make any progress. We make the following assumptions.

Assumption 3.1 *The matrices A_k have been scaled so that $\|\mathcal{A}\|_2 = \Omega(1)$.*

Assumption 3.2 *$\mathcal{J}$ is numerically non-singular, i.e., $\sigma_{\min}(\mathcal{J}) \gg \epsilon \cdot \|\mathcal{J}\|_2$.*

Assumption 3.3 *The current iterate (X, Z, y) is close to the exact solution (X^*, Z^*, y^*) and*

$$\|b\|_2 \le O\left(\|\mathcal{A}\|_2\, \|X\|_2\right) \quad \textit{and} \quad \|C\|_2 \le O\left(\|Z\|_2 + \|\mathcal{A}\|_2\, \|y\|_2\right).$$

Assumption 3.4 *The Schur complement $\mathcal{M}$ is explicitly computed and the equation (2.5-a) is then solved by a backward stable method.*

We first consider round-off errors in $\widehat{\mathcal{R}}$. The computation of $\mathcal{R}$ involves a number of simple matrix-matrix and matrix-vector products as well as matrix and vector additions. By standard error analysis (see Golub and van Loan [10, Ch. 2]) and Assumption 3.3, we have

$$\widehat{\mathcal{R}} = \mathcal{R} + \begin{pmatrix} O\left(\epsilon \cdot \|X\|_2\, \|Z\|_2\right) \\ O\left(\epsilon \cdot \|Z\|_2 + \epsilon \cdot \|\mathcal{A}\|_2\, \|y\|_2\right) \\ O\left(\epsilon \cdot \|\mathcal{A}\|_2\, \|X\|_2\right) \end{pmatrix} = \mathcal{R} + O\left(\epsilon \cdot \|\mathcal{J}\|_2\, \|\mathcal{X}\|_2\right). \tag{3.11}$$

We now consider round-off errors on the right hand side of equation (2.4). Let A be a matrix and x be a vector. Then the round-off errors in the matrix-vector product $A\,x$ satisfy (see Higham [12, Ch. 3])

$$\mathbf{fl}(A\,x) = (A + \delta A)\ x\,, \quad \text{where} \quad |\delta A| \le O(\epsilon) \cdot |A|\,. \tag{3.12}$$

Since $\mathcal{F}\,\widehat{r}_d$ is a matrix-vector product, by equation (3.12) we have

$$\mathbf{fl}\,(\mathcal{F}\,\widehat{r}_d) = (\mathcal{F} + \delta_1\mathcal{F})\;\widehat{r}_d\,, \quad \text{where} \quad \delta_1\mathcal{F} = O\,(\epsilon\cdot\|\mathcal{F}\|_2)\;.$$

By our model of arithmetic (1.7), every component in $\mathbf{fl}\,(\mathcal{F}\,\widehat{r}_d) - \widehat{r}_c$ is computed to high relative accuracy. So there exists an $n(n+1)/2$-by-$n(n+1)/2$ diagonal perturbation matrix $\Delta_4 = O(\epsilon)$ such that

$$\mathbf{fl}\,(\mathcal{F}\,\widehat{r}_d - \widehat{r}_d) = (\mathcal{I} + \Delta_4)\;((\mathcal{F} + \delta_1\mathcal{F})\;\widehat{r}_d - \widehat{r}_d)\;.$$

According to (3.8), there exist $n(n+1)/2$-by-$n(n+1)/2$ perturbation matrices Δ_5 and Δ_6 such that

$$\begin{aligned}\mathbf{fl}\left(\mathcal{E}^{-1}\,(\mathcal{F}\,\widehat{r}_d - \widehat{r}_d)\right) &= (\mathcal{I} + \Delta_6)\;\left(\mathcal{E}^{\dagger}\right)^{-1}\;(\mathcal{I} + \Delta_5)\;\mathbf{fl}\,(\mathcal{F}\,\widehat{r}_d - \widehat{r}_d) \\ &= (\mathcal{I} + \Delta_6)\;\left(\mathcal{E}^{\dagger}\right)^{-1}\;(\mathcal{I} + \Delta_5)\;(\mathcal{I} + \Delta_4)\;((\mathcal{F} + \delta_1\mathcal{F})\;\widehat{r}_d - \widehat{r}_d)\;.\end{aligned}$$

Putting this together, we can write

$$\begin{aligned}&\mathbf{fl}\left(\widehat{r}_p + \mathcal{A}\,\mathcal{E}^{-1}\;(\mathcal{F}\,\widehat{r}_d - \widehat{r}_c)\right) \\ &\quad = (\mathcal{I} + \Delta_7)\;\left(\widehat{r}_p + \mathbf{fl}\left(\mathcal{A}\,\mathcal{E}^{-1}\;(\mathcal{F}\,\widehat{r}_d - \widehat{r}_d)\right)\right) = (\mathcal{I} + \Delta_7)\;\left(\widehat{r}_p + (\mathcal{A} + \delta_1\mathcal{A})\;\mathbf{fl}\left(\mathcal{E}^{-1}\,(\mathcal{F}\,\widehat{r}_d - \widehat{r}_d)\right)\right) \\ &\quad = (\mathcal{I} + \Delta_7)\;\left(\widehat{r}_p + (\mathcal{A} + \delta_1\mathcal{A})\;(\mathcal{I} + \Delta_6)\;\left(\mathcal{E}^{\dagger}\right)^{-1}\;(\mathcal{I} + \Delta_8)\;((\mathcal{F} + \delta_1\mathcal{F})\;\widehat{r}_d - \widehat{r}_c)\right)\,, \qquad (3.13)\end{aligned}$$

where $\Delta_7 = O(\epsilon) \in \mathbf{R}^{m\times m}$ is a diagonal perturbation matrix; $\delta_1\mathcal{A} = O(\epsilon\cdot\|\mathcal{A}\|_2)$ is an m-by-$n(n+1)/2$ perturbation matrix; and $\Delta_8 = (\mathcal{I} + \Delta_5)(\mathcal{I} + \Delta_4) - \mathcal{I} = O(\epsilon)$.

Similar to (3.13), the (i,j) entry of the computed Schur complement $\mathcal{M}$ can be written as

$$\begin{aligned}&\mathbf{svec}\,(A_i + \delta_{i,j}A_i)^T\;(I + \Delta_{i,j})\;\left(\mathcal{E}^{\dagger}\right)^{-1}\;(I + \bar{\Delta}_{i,j})\;(\mathcal{F} + \delta_{i,j}\mathcal{F})\;\mathbf{svec}(A_j) \\ &\quad = \mathbf{svec}\,(A_i)^T\;\left(\mathcal{E}^{\dagger}\right)^{-1}\;\mathcal{F}\,\mathbf{svec}\,(A_j) + O\left(\epsilon\cdot\|\mathcal{A}\|_2^2\;\|\left(\mathcal{E}^{\dagger}\right)^{-1}\|_2\;\|\mathcal{F}\|_2\right) \\ &\quad = \left(\mathcal{M}^{\dagger}\right)_{i,j} + O\left(\epsilon\cdot\|\mathcal{A}\|_2^2\;\|\left(\mathcal{E}^{\dagger}\right)^{-1}\|_2\;\|\mathcal{F}\|_2\right)\;.\end{aligned}$$

In other words, the computed $\mathcal{M}$ can be written as $\mathcal{M}^{\dagger} + O\left(\epsilon\cdot\|\mathcal{A}\|_2^2\;\|\left(\mathcal{E}^{\dagger}\right)^{-1}\|_2\;\|\mathcal{F}\|_2\right)$. By Assumption 3.4, the backward errors committed during the solution of (2.5-a) are bounded by $O\,(\epsilon\cdot\|\mathcal{M}\|_2)$, which is also bounded by $O\left(\epsilon\cdot\|\mathcal{A}\|_2^2\;\|\left(\mathcal{E}^{\dagger}\right)^{-1}\|_2\;\|\mathcal{F}\|_2\right)$. Putting all these errors together, we have

$$\left(\mathcal{M}^{\dagger} + \delta\mathcal{M}^{\dagger}\right)\;\widehat{dy} = \mathbf{fl}\left(\widehat{r}_p + \mathcal{A}\,\mathcal{E}^{-1}\;(\mathcal{F}\,\widehat{r}_d - \widehat{r}_c)\right),\;\; \delta\mathcal{M}^{\dagger} = O\left(\epsilon\cdot\|\mathcal{A}\|_2^2\;\|\left(\mathcal{E}^{\dagger}\right)^{-1}\|_2\;\|\mathcal{F}\|_2\right). \qquad (3.14)$$

With similar analysis, the round-off errors in equations (2.5-b) and (2.5-c) can be written as

$$\begin{aligned}\widehat{dZ} &= \mathbf{smat}\left((\mathcal{I} + \Delta_9)\;\left(\widehat{r}_d - (\mathcal{A} + \delta_2\mathcal{A})^T\;\widehat{dy}\right)\right) \\ \widehat{dX} &= \mathbf{smat}\left((\mathcal{I} + \Delta_{11})\;\left(\mathcal{E}^{\dagger}\right)^{-1}\;(\mathcal{I} + \Delta_{10})\;\left(\widehat{r}_c - (\mathcal{F} + \delta_2\mathcal{F})\;\mathbf{svec}\left(\widehat{dZ}\right)\right)\right)\;.\end{aligned}$$

We now rewrite these equations in a form similar to (2.4) to get

$$\begin{pmatrix} \mathcal{E}^{\dagger} + \delta\mathcal{E}^{\dagger} & \mathcal{F} + \delta_2\mathcal{F} & 0 \\ 0 & (\mathcal{I} + \Delta_9)^{-1} & (\mathcal{A} + \delta_2\mathcal{A})^T \\ 0 & 0 & \mathcal{M}^{\dagger} + \delta\mathcal{M}^{\dagger} \end{pmatrix} \widehat{d\mathcal{X}} = \begin{pmatrix} \widehat{r}_c \\ \widehat{r}_d \\ \mathbf{fl}\,(\widehat{r}_p + \mathcal{A}\,\mathcal{E}^{-1}\;(\mathcal{F}\,\widehat{r}_d - \widehat{r}_c)) \end{pmatrix}, \qquad (3.15)$$

where (see (3.7))

$$\delta\mathcal{E}^\dagger = (\mathcal{I}+\Delta_{10}I)^{-1}\cdot\mathcal{E}^\dagger\cdot(I+\Delta_{11}I)^{-1} - \mathcal{E}^\dagger = O\left(\epsilon\cdot\|Z\|_2\right) .$$

With (3.13), we rewrite equation (3.15) in a form similar to (2.3):

$$\begin{pmatrix} \mathcal{I} & 0 & 0 \\ 0 & \mathcal{I} & 0 \\ \mathcal{L}_{3,1} & \mathcal{L}_{3,2} & (\mathcal{I}+\Delta_7)^{-1} \end{pmatrix} \begin{pmatrix} \mathcal{E}^\dagger+\delta\mathcal{E}^\dagger & \mathcal{F}+\delta_2\mathcal{F} & 0 \\ 0 & (\mathcal{I}+\Delta_9)^{-1} & (\mathcal{A}+\delta_2\mathcal{A})^T \\ 0 & 0 & \mathcal{M}^\dagger+\delta\mathcal{M}^\dagger \end{pmatrix} \widehat{d\mathcal{X}} = \begin{pmatrix} \widehat{r}_c \\ \widehat{r}_d \\ \widehat{r}_p \end{pmatrix} . \tag{3.16}$$

where

$$\begin{aligned} \mathcal{L}_{3,1} &= (\mathcal{A}+\delta_1\mathcal{A})\ (\mathcal{I}+\Delta_6)\ \left(\mathcal{E}^\dagger\right)^{-1}\ (\mathcal{I}+\Delta_8) = \mathcal{A}\ \left(\mathcal{E}^\dagger\right)^{-1} + O\left(\epsilon\cdot\|\mathcal{A}\|_2\ \|\left(\mathcal{E}^\dagger\right)^{-1}\|_2\right) \\ \mathcal{L}_{3,2} &= -\mathcal{L}_{3,1}\cdot(\mathcal{F}+\delta_1\mathcal{F}) = -\mathcal{A}\ \left(\mathcal{E}^\dagger\right)^{-1}\ \mathcal{F} + O\left(\epsilon\cdot\|\mathcal{A}\|_2\ \|\left(\mathcal{E}^\dagger\right)^{-1}\|_2\ \|\mathcal{F}\|_2\right) . \end{aligned}$$

Comparing equation (3.16) with (3.9), we see that the backward error matrix $\delta\mathcal{J}$ satisfies

$$\begin{aligned} \delta\mathcal{J} &= \begin{pmatrix} \mathcal{I} & 0 & 0 \\ 0 & \mathcal{I} & 0 \\ \mathcal{L}_{3,1} & \mathcal{L}_{3,2} & (\mathcal{I}+\Delta_7)^{-1} \end{pmatrix} \begin{pmatrix} \mathcal{E}^\dagger+\delta\mathcal{E}^\dagger & \mathcal{F}+\delta_2\mathcal{F} & 0 \\ 0 & (\mathcal{I}+\Delta_9)^{-1} & (\mathcal{A}+\delta_2\mathcal{A})^T \\ 0 & 0 & \mathcal{M}^\dagger+\delta\mathcal{M}^\dagger \end{pmatrix} - \mathcal{J} \\ &= \begin{pmatrix} \mathcal{E}^\dagger+\delta\mathcal{E}^\dagger-\mathcal{E} & \delta_2\mathcal{F} & 0 \\ 0 & (\mathcal{I}+\Delta_9)^{-1}-\mathcal{I} & \delta_2\mathcal{A}^T \\ \mathcal{L}_{3,1}\ \mathcal{E}^\dagger-\mathcal{A} & \mathcal{L}_{3,1}\ \left(\mathcal{F}-\mathcal{F}\ (\mathcal{I}+\Delta_9)^{-1}\right) & (\mathcal{I}+\Delta_7)^{-1}\ \mathcal{M}^\dagger+\mathcal{L}_{3,2}\ \mathcal{A}^T \end{pmatrix} \\ &\quad + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \mathcal{L}_{3,1}\cdot\delta\mathcal{E}^\dagger & \mathcal{L}_{3,1}\ \left(\delta_2\mathcal{F}-\delta_1\mathcal{F}\ (\mathcal{I}+\Delta_9)^{-1}\right) & (\mathcal{I}+\Delta_7)^{-1}\ \delta\mathcal{M}^\dagger+\mathcal{L}_{3,2}\cdot\delta_2\mathcal{A}^T \end{pmatrix} \\ &= O\left(\epsilon\cdot\|\mathcal{J}\|_2\right) + O\left(\epsilon\cdot(1+\|\mathcal{A}\|_2)^2\ (\|\mathcal{E}\|_2+\|\mathcal{F}\|_2)\ \|\left(\mathcal{E}^\dagger\right)^{-1}\|_2\right) . \end{aligned} \tag{3.17}$$

The first term in (3.17), which includes backward errors in the first two block rows of $\mathcal{J}$, is a small backward perturbation to $\mathcal{J}$; but the second term, which includes backward errors in the last block row, could be very large. In fact, relation (3.10) implies that there maybe little accuracy in the computed search direction $\widehat{d\mathcal{X}}$ if the second term is so large that

$$\epsilon\cdot(1+\|\mathcal{A}\|_2)^2\ (\|\mathcal{E}\|_2+\|\mathcal{F}\|_2)\ \|\left(\mathcal{E}^\dagger\right)^{-1}\|_2 = \Omega\left(\sigma_{\min}(\mathcal{J})\right) .$$

Since

$$\lambda_{\min}(Z) = \lambda_{\min}(Z^\dagger) + O(\epsilon\cdot\|Z\|_2) = \left\|\left(\mathcal{E}^\dagger\right)^{-1}\right\|_2^{-1} + O(\epsilon\cdot\|Z\|_2) ,$$

it follows that Algorithm 3.1 could stop making further progress if

$$\frac{\lambda_{\min}(Z)}{\|Z\|_2+\|X\|_2} = \frac{\lambda_{\min}(Z)}{\|\mathcal{E}\|_2+\|\mathcal{F}\|_2} = \Omega\left(\epsilon\cdot\kappa(\mathcal{J})\right) . \tag{3.18}$$

Putting it another way, we might expect Algorithm 3.1 to stop making progress as soon as it reaches an iterate (X, Z, y) that satisfies (3.18). Since the optimal solution Z^* is in general singular (see (1.2)), if $\mathcal{J}$ is well-conditioned at every iteration, then (3.18) does not put a limit on the amount of accuracy in the numerical solution. On the other hand, if $\mathcal{J}$ is ill-conditioned, then (3.18) indicates that Algorithm 3.1 could stop making progress well before some eigenvalues of Z become sufficiently small, making Algorithm 3.1 numerically unstable. Since equation (3.11) indicates that the right hand side of (2.2) is always computed very accurately, the backward errors in $\mathcal{J}$ appear to be the only source of potential numerical instability in Algorithm 3.1.

We have only analyzed Algorithm 3.1 in §3.3. For Algorithm 3.2, the coefficient matrices of the two linear systems of equations it solves are the same as that in (2.2). Repeat the arguments in §3.3, it is easy to see that Algorithm 3.2 might stop making progress as soon as it reaches an iterate (X, Z, y) that satisfies (3.18), and hence Algorithm 3.2 could be numerically unstable if $\mathcal{J}$ is ill-conditioned.

The potential numerical instability of Algorithms 3.1 and 3.2 is due to the block LU factorization procedure discussed in §2.1. As our numerical results in §5.3 indicate, this instability will disappear if the search direction is computed by solving equation (2.2) as a dense linear system of equations.

Finally, we caution that the above analysis merely identifies situations in which Algorithms 3.1 and 3.2 *could* be numerically unstable. It does *not* assert instability in these situations nor does it guarantee progress of the algorithms in other situations. Despite this serious weakness, it is clear that this analysis does provide important new insight into understanding the numerical stability of Algorithms 3.1 and 3.2 in finite precision arithmetic. In §5 we will present results from our numerical experiments that support the analysis in §3.3.

3.4 Error Analysis for a Variation of the AHO Method

Several mathematically equivalent formulas are possible for computing the search direction. For example, the expression $\mathcal{F} r_d - r_c$ in (2.5-a) can be written equivalently as

$$\mathcal{F}\, r_d - r_c = \mathbf{svec}\left(X\ \left(C - \mathbf{smat}(\mathcal{A}^T\, y)\right) + \left(C - \mathbf{smat}(\mathcal{A}^T\, y)\right)\ X - 2\mu I\right)\ . \tag{3.19}$$

However, Alizadeh, Haeberly and Overton [3] observed numerical instability leading to significant loss of primal feasibility near the exact solution with (3.19). Todd, Toh and Tütüncü [27] also observed that some mathematically equivalent formulas for computing the search direction appear to be much less numerically stable than others in the case of the NT method.

In the following we briefly explain why (3.19) leads to instability. It is clear that it does not hold in general in *finite precision.* Define

$$\eta = \mathbf{fl}\left(\mathbf{svec}\left(X\ \left(C - \mathbf{smat}(\mathcal{A}^T\, y)\right) + \left(C - \mathbf{smat}(\mathcal{A}^T\, y)\right)\ X - 2\,\mu\, I\right)\right) - (\mathcal{F}\,\widehat{r}_d - \widehat{r}_c)\ .$$

Equation (3.13) now becomes

$$\begin{aligned}\mathbf{fl}\left(\widehat{r}_p + \mathcal{A}\,\mathcal{E}^{-1}\ (\mathcal{F}\,\widehat{r}_d - \widehat{r}_c)\right) &= (\mathcal{I}+\Delta_7)\ \left(\widehat{r}_p + (\mathcal{A}+\delta_1\mathcal{A})\ (\mathcal{I}+\Delta_6)\ \left(\mathcal{E}^\dagger\right)^{-1}\ (\mathcal{I}+\Delta_8)\ (\mathcal{F}\,\widehat{r}_d - \widehat{r}_c + \eta)\right)\\ &= (\mathcal{I}+\Delta_7)\ (\widehat{r}_p + \mathcal{L}_{3,1}\,\eta + \mathcal{L}_{3,1}\ (\mathcal{F}\,\widehat{r}_d - \widehat{r}_c)),\ \ \mathcal{L}_{3,1} = (\mathcal{A}+\delta_1\mathcal{A})\ (\mathcal{I}+\Delta_6)\ \left(\mathcal{E}^\dagger\right)^{-1}\ (\mathcal{I}+\Delta_8)\,.\end{aligned} \tag{3.20}$$

And equation (3.15) now takes the form

$$\begin{pmatrix} \mathcal{E}^\dagger + \delta\mathcal{E}^\dagger & \mathcal{F}+\delta_2\mathcal{F} & 0 \\ 0 & (\mathcal{I}+\Delta_9)^{-1} & (\mathcal{A}+\delta_2\mathcal{A})^T \\ 0 & 0 & \mathcal{M}^\dagger + \delta\mathcal{M}^\dagger \end{pmatrix} \widehat{d\mathcal{X}} = \begin{pmatrix} \widehat{r}_c \\ \widehat{r}_d \\ \mathbf{fl}\left(\widehat{r}_p + \mathcal{A}\,\mathcal{E}^{-1}\ (\mathcal{F}\,\widehat{r}_d - \widehat{r}_c)\right) \end{pmatrix}, \tag{3.21}$$

where $\mathbf{fl}\left(\widehat{r}_p + \mathcal{A}\,\mathcal{E}^{-1}\ (\mathcal{F}\,\widehat{r}_d - \widehat{r}_c)\right)$ satisfies (3.20) instead of (3.13). Hence equation (3.19) amounts to a replacement of mathematically equivalent but numerically different right hand sides in the middle of a block Gaussian elimination procedure. After this replacement, equation (3.16) now becomes

$$\begin{pmatrix} \mathcal{I} & 0 & 0 \\ 0 & \mathcal{I} & 0 \\ \mathcal{L}_{3,1} & -\mathcal{L}_{3,1}\,\mathcal{F} & (\mathcal{I}+\Delta_7)^{-1} \end{pmatrix} \begin{pmatrix} \mathcal{E}^\dagger + \delta\mathcal{E}^\dagger & \mathcal{F}+\delta_2\mathcal{F} & 0 \\ 0 & (\mathcal{I}+\Delta_9)^{-1} & (\mathcal{A}+\delta_2\mathcal{A})^T \\ 0 & 0 & \mathcal{M}^\dagger + \delta\mathcal{M}^\dagger \end{pmatrix} \widehat{d\mathcal{X}} = \begin{pmatrix} \widehat{r}_c \\ \widehat{r}_d \\ \widehat{r}_p + \mathcal{L}_{3,1}\,\eta \end{pmatrix}.$$

On the other hand, similar to (3.11), we have

$$\|\eta\|_2 = O\left(\epsilon \cdot \|\mathcal{J}\|_2 \, \|\mathcal{X}\|_2\right) \quad \text{and hence} \quad \mathcal{L}_{3,1}\,\eta = O\left(\epsilon \cdot \|\mathcal{A}\|_2 \, \| \left(\mathcal{E}^{\dagger}\right)^{-1} \|_2 \, \|\mathcal{J}\|_2 \, \|\mathcal{X}\|_2\right) . \tag{3.22}$$

As before, the backward errors in the coefficient matrix of the equation above are bounded by (3.17). However, the round-off errors on the right hand side could become huge as the iterates converge. For example, assume that the current iterate (X, Z, y) is sufficiently close to (X^*, Z^*, y^*) so that

$$\lambda_{\min}(Z) \le O\left(\sqrt{\epsilon} \cdot \|\mathcal{X}\|_2\right) \quad \text{and} \quad \|\mathcal{R}\|_2 \le O\left(\sqrt{\epsilon} \cdot \|\mathcal{J}\|_2 \, \|\mathcal{X}\|_2\right) .$$

It follows from (3.22) that there might be *no* significant digits at all in the right hand side vector $\widehat{r}_p + \mathcal{L}_{3,1}\,\eta$, and $\|\widehat{r}_p + \mathcal{L}_{3,1}\,\eta\|_2$ could be significantly larger than $\|\widehat{r}_c\|_2$ and $\|\widehat{r}_d\|_2$. Hence the computed search direction could be completely in error. It follows that the AHO method with (3.19) could stop making progress when $\|\mathcal{R}\|_2 = O\left(\sqrt{\epsilon} \cdot \|\mathcal{J}\|_2 \, \|\mathcal{X}\|_2\right)$, even if $\mathcal{J}$ is well-conditioned.

Similar analysis holds for the NT method. As we will show in §4, the NT method, when implemented according to a similar block Gaussian elimination procedure, is highly accurate in general. On the other hand, if mathematically equivalent but numerically different formulas are used to replace computed quantities during the computation, as is done for the AHO method in (3.21), then the resulting method could be highly unstable. The same argument holds for all other methods in the TTT family as well.

4 Analysis of the TTT Methods

4.1 The TTT Methods

A search direction in the TTT family is a search direction defined by (2.2) with P satisfying one of the two mathematically equivalent equations in (2.10). We assume that a proper choice of the singular vector matrices of $R\,H^T$ in (2.9) has been made so that B is a diagonal matrix. Arrange the singular values as $0 < \sigma_1 \le \sigma_2 \le \cdots \le \sigma_n$ and let

$$B = \mathbf{diag}(\beta_1, \cdots, \beta_n) \quad \text{and} \quad \widetilde{B} = B\,\Sigma = \mathbf{diag}\left(\beta_1\,\sigma_1, \cdots, \beta_n\,\sigma_n\right). \tag{4.1}$$

We will base our development on the assumption that P is chosen using expression (2.10-a). It follows from (2.2) and (2.12) that

$$\mathcal{E} = \left(S\;B^{-1}\;\widetilde{H}\right) \otimes_s \left(S\;B\;\widetilde{H}\right) \quad \text{and} \quad \mathcal{F} = \left(S\;B\;\widetilde{H}\;X\right) \otimes_s \left(S\;B^{-1}\;\widetilde{H}^{-T}\right) . \tag{4.2-a}$$

With (2.10-b) and (2.11), these expressions can be rewritten as

$$\mathcal{E} = \left(S\;B^{-1}\,\Sigma\,\widetilde{R}^{-T}\right) \otimes_s \left(S\;B\,\Sigma\,\widetilde{R}^{-T}\right) \quad \text{and} \quad \mathcal{F} = \left(S\;\widetilde{B}\;\widetilde{R}\right) \otimes_s \left(S\;\widetilde{B}^{-1}\;\widetilde{R}\right) . \tag{4.2-b}$$

Similarly, equations (2.2), (2.10) and (2.12) imply that

$$r_c = \mathbf{svec}\left(\mu\,I - S\,\mathbf{H}_B\left(\widetilde{H}\,X\,\widetilde{H}^T\right)\,S^T\right) = \mathbf{svec}\left(S\,\mathbf{smat}\left(\widetilde{r}_c\right)\,S^T\right) , \text{ where } \widetilde{r}_c \overset{\text{def}}{=} \mathbf{svec}\left(\mu\,I - \Sigma^2\right). \tag{4.3}$$

With (4.2-b), the matrix-vector product (2.6) can be written as

$$\begin{aligned} \mathcal{F}\,u &= \frac{1}{2}\mathbf{svec}\left(S\;\left(\widetilde{B}\;\widetilde{R}\,\mathbf{smat}(u)\;\widetilde{R}^T\;\widetilde{B}^{-1} + \widetilde{B}^{-1}\;\widetilde{R}\,\mathbf{smat}(u)\;\widetilde{R}^T\;\widetilde{B}\right)\;S^T\right) \\ &= \frac{1}{2}\mathbf{svec}\left(S\;\left(D_{\mathcal{F}} \odot \left(\widetilde{R}\,\mathbf{smat}(u)\;\widetilde{R}^T\right)\right)\;S^T\right) , \end{aligned} \tag{4.4}$$

where $X \odot Y = (X_{i,j} \cdot Y_{i,j})$ is the *Hadamard product*; and $D_{\mathcal{F}} = \left(\frac{\beta_i\, \sigma_i}{\beta_j\, \sigma_j} + \frac{\beta_j\, \sigma_j}{\beta_i\, \sigma_i} \right) = \left(\frac{\beta_i^2\, \sigma_i^2 + \beta_j^2\, \sigma_j^2}{\beta_i\, \beta_j\, \sigma_i\, \sigma_j} \right)$.

We now solve equation (2.7). With (4.2-b), the left hand side of (2.7-b) can be rewritten as

$$S \left(B^{-1} \Sigma\, \widetilde{R}^{-T} U\, \widetilde{R}^{-1} \Sigma\, B + B\, \Sigma\, \widetilde{R}^{-T} U\, \widetilde{R}^{-1} \Sigma\, B^{-1} \right) S^T = S \left(\left(\frac{\beta_j\, \sigma_i\, \sigma_j}{\beta_i} + \frac{\beta_i\, \sigma_i\, \sigma_j}{\beta_j} \right) \cdot \left(\widetilde{R}^{-T} U\, \widetilde{R}^{-1} \right)_{ij} \right) S^T .$$

Hence the solution to equation (2.7) is

$$\mathcal{E}^{-1}\, v = 2\, \mathbf{svec} \left(\widetilde{R}^T \, \left(D_{\mathcal{E}} \odot \left(S^T\, \mathbf{smat}(v)\, S \right) \right)\, \widetilde{R} \right) \quad \text{where} \quad D_{\mathcal{E}} = \left(\frac{\beta_i\, \beta_j}{\sigma_i\, \sigma_j \left(\beta_i^2 + \beta_j^2 \right)} \right) . \tag{4.5}$$

With equations (4.4) and (4.5), we can rewrite the Schur complement matrix $\mathcal{M}$ as

$$\begin{aligned} \mathcal{M} &= \left(\mathbf{svec}(A_i)^T\, \mathcal{E}^{-1}\, \mathcal{F}\, \mathbf{svec}(A_j) \right) \\ &= 2\, \left(\mathbf{svec}(A_i)^T \cdot \mathbf{svec} \left(\widetilde{R}^T\, \left(D_{\mathcal{E}} \odot \left(S^T\, \mathbf{smat}\,(\mathcal{F}\, \mathbf{svec}(A_j))\;S \right) \right)\, \widetilde{R} \right) \right) \\ &= \left(\mathbf{svec}(A_i)^T \cdot \mathbf{svec} \left(\widetilde{R}^T\, \left(D_{\mathcal{E}} \odot \left(S^T \left(S \left(D_{\mathcal{F}} \odot \left(\widetilde{R}\, A_j\, \widetilde{R}^T \right) \right) S^T \right) S \right) \widetilde{R} \right) \right) \right) \\ &= \left(\mathbf{svec}(A_i)^T \cdot \mathbf{svec} \left(\widetilde{R}^T\, \left((D_{\mathcal{E}} \odot D_{\mathcal{F}}) \odot \left(\widetilde{R}\, A_j\, \widetilde{R}^T \right) \right)\, \widetilde{R} \right) \right) \\ &= \left(\left(\widetilde{R}\, A_i\, \widetilde{R}^T \right) \bullet \left((D_{\mathcal{E}} \odot D_{\mathcal{F}}) \odot \left(\widetilde{R}\, A_j\, \widetilde{R}^T \right) \right) \right) = \widetilde{\mathcal{A}}\, \mathcal{D}_{\mathcal{M}}\, \widetilde{\mathcal{A}}^T , \end{aligned} \tag{4.6}$$

where $\widetilde{\mathcal{A}} = \left(\mathbf{svec} \left(\widetilde{R}\, A_1\, \widetilde{R}^T \right), \cdots, \mathbf{svec} \left(\widetilde{R}\, A_m\, \widetilde{R}^T \right) \right)^T$; and $\mathcal{D}_{\mathcal{M}}$ is an $n(n+1)/2$-by-$n(n+1)/2$ diagonal matrix that satisfies

$$\mathcal{D}_{\mathcal{M}}\, \mathbf{e} = \mathbf{svec}\,(D_{\mathcal{E}} \odot D_{\mathcal{F}}) = \mathbf{svec} \left(\frac{\beta_i^2\, \sigma_i^2 + \beta_j^2\, \sigma_j^2}{\sigma_i^2\, \sigma_j^2\, \left(\beta_i^2 + \beta_j^2 \right)} \right)$$

for the vector $\mathbf{e}$ in §1.4. Note that the matrix $\mathcal{D}_{\mathcal{M}}$ is the only part in $\mathcal{M}$ that is affected by B. For *any* choice of B, the entries of the matrix $D_{\mathcal{E}} \odot D_{\mathcal{F}}$ are always bounded. In fact,

$$\frac{1}{\sigma_i^2 + \sigma_j^2} \le \frac{\beta_i^2\, \sigma_i^2 + \beta_j^2\, \sigma_j^2}{\sigma_i^2\, \sigma_j^2\, \left(\beta_i^2 + \beta_j^2 \right)} \le \frac{1}{\sigma_i^2} + \frac{1}{\sigma_j^2} . \tag{4.7}$$

Since the two HKM search directions [11, 13, 18] $P^T P = X^{-1}$ and $P^T P = Z$ correspond to $B = \Sigma^{-1}$ and I, respectively, and since the NT direction [23, 24] corresponds to $B = \Sigma^{-\frac{1}{2}}$ (see §2.2), their corresponding $D_{\mathcal{E}} \odot D_{\mathcal{F}}$ matrices take the form

$$\left(\frac{2}{\sigma_i^2 + \sigma_j^2} \right) , \quad \left(\frac{\sigma_i^2 + \sigma_j^2}{2\, \sigma_i^2\, \sigma_j^2} \right) \quad \text{and} \quad \left(\frac{1}{\sigma_i\, \sigma_j} \right) .$$

Now we use equations (4.4) and (4.5) to simplify the right hand side of (2.5). By (4.3) and (4.5),

$$\mathcal{E}^{-1}\, r_c = 2\, \mathbf{svec} \left(\widetilde{R}^T\, \left(D_{\mathcal{E}} \odot \left(S^T \left(\mu\, I - S\, \Sigma^2\, S^T \right) S \right) \right)\, \widetilde{R} \right) = \mathbf{svec} \left(\widetilde{R}^T\, \left(D_{\mathcal{E}} \odot \mathbf{smat}\,(\widetilde{r}_c) \right)\, \widetilde{R} \right) .$$

Combining this relation with (4.4) and (4.5), and with some algebra similar to that used to obtain (4.6),

$$\mathcal{E}^{-1}\, \left(\mathcal{F}\, r_d - r_c \right) = \mathbf{svec} \left(\widetilde{R}^T\, \left((D_{\mathcal{E}} \odot D_{\mathcal{F}}) \odot \left(\widetilde{R}\, \mathbf{smat}(r_d)\, \widetilde{R}^T \right) - D_{\mathcal{E}} \odot \mathbf{smat}\,(\widetilde{r}_c) \right)\, \widetilde{R} \right) . \tag{4.8}$$

However, we will compute $\mathcal{E}^{-1}\ (r_c - \mathcal{F}\,\mathbf{svec}(dZ))$ differently. With (4.2-a), the left hand side of (2.7-b) can be rewritten as

$$S\left(B^{-1}\,\widetilde{H}\,U\,\widetilde{H}^T\,B + B\,\widetilde{H}\,U\,\widetilde{H}^T\,B^{-1}\right)S^T = S\left(\left(\frac{\beta_j}{\beta_i}+\frac{\beta_i}{\beta_j}\right)\cdot\left(\widetilde{H}\,U\,\widetilde{H}^T\right)_{ij}\right)S^T.$$

Hence the solution to equation (2.7) can also be written as

$$\mathcal{E}^{-1}\ v = 2\,\mathbf{svec}\left(\widetilde{H}^{-1}\ \left(\bar{D}_{\mathcal{E}}\odot\left(S^T\,\mathbf{smat}(v)\,S\right)\right)\ \widetilde{H}^{-T}\right)\quad \text{where}\quad \bar{D}_{\mathcal{E}} = \left(\frac{\beta_i\,\beta_j}{\beta_i^2+\beta_j^2}\right)\ .$$

Combined with (4.3) and (4.4), and after some algebra, we obtain

$$\mathcal{E}^{-1}\ (r_c - \mathcal{F}\,\mathbf{svec}(dZ)) = \mathbf{svec}\left(\widetilde{H}^{-1}\ \left(\bar{D}_{\mathcal{E}}\odot\mathbf{smat}\,(\widetilde{r}_c) - (\bar{D}_{\mathcal{E}}\odot D_{\mathcal{F}})\odot\left(\widetilde{R}\,dZ\,\widetilde{R}^T\right)\right)\ \widetilde{H}^{-T}\right)\ . \quad (4.9)$$

As we have seen throughout §4.1, due to relation (2.11), $\mathcal{E}$, $\mathcal{F}$, and $\mathcal{M}$ can be expressed in several different but mathematically equivalent ways, each of which may lead to a different numerical solution to (2.5). We have chosen to solve (2.5) via the SVD (2.9) in such a way that makes the symmetry of $\mathcal{M}$ explicit and avoids the explicit inversion of $\widetilde{H}$ and $\widetilde{R}$ everywhere except in (4.9). Our approach is somewhat different from that of Todd, Toh and Tütüncü [27]. Algorithm 4.1 below is a more formal description of the method described in this section. We will postpone some details on how to compute expressions in (4.8) and (4.9) to §4.3. We will also discuss a new choice of B in §5.1.

Algorithm 4.1 TTT Methods

1. Choose a matrix B in (2.10-a).
2. Choose $0 \le \sigma < 1$ and determine (dX, dZ, dy) from (2.5-a), (2.5-b), (4.6), (4.8) and (4.9), using
$$\mu = \sigma\cdot\frac{X\bullet Z}{n} = \sigma\cdot\frac{\mathbf{tr}\,(\Sigma^2)}{n}\ .$$
3. Choose steplengths α and β using (3.4) and update the iterates by
$$(X, Z, y) \leftarrow (X + \alpha\,dX, Z + \beta\,dZ, y + \beta\,dy)\ .$$

The main cost of Algorithm 4.1 is in the computation and factorization of $\mathcal{M}$. To compute $\widetilde{\mathcal{A}}$ in (4.6), we need to explicitly compute the matrices $\widetilde{R}\,A_i\,\widetilde{R}^T$ for $i = 1, \cdots, m$, which costs about $3mn^3$ flops (see §4.3). Since $\mathcal{M}$ is symmetric, computing $\mathcal{M}$ from $\widetilde{\mathcal{A}}$ costs about $1/2m^2n^2$ flops. The Cholesky factorization of $\mathcal{M}$ costs about $1/3m^3$ flops. Adding it all up, we see that Algorithm 4.1 costs about $1/3m^3 + 1/2m^2n^2 + 3mn^3$ flops per step, roughly half of the per step cost of Algorithm 3.1. Applying the PC rule to Algorithm 4.1, we get an algorithm similar to Algorithm 3.2.

Algorithm 4.2 TTT Methods with Mehrotra PC Rule

1. Choose a matrix B in (2.10-a).
2. Determine (dX, dZ, dy) from the second step of Algorithm 4.1, using $\mu = 0$.
3. Choose steplengths α and β using (3.4), compute $T = \widetilde{H}\ dX\ dZ\ \widetilde{H}^{-1}$ and define
$$\mu = \frac{X\bullet Z}{n}\cdot\left(\frac{(X+\alpha\,dX)\bullet(Z+\beta\,dZ)}{X\bullet Z}\right)^3 \quad\text{and}\quad \widetilde{r}_c = \mathbf{svec}\left(\mu\,I - \Sigma^2 - \mathbf{H}_B\,(T)\right)\ .$$
4. Redetermine (dX, dZ, dy) from the second step of Algorithm 4.1, using redefined μ and $\widetilde{r}_c$; redetermine steplengths α and β using (3.4); and update the iterates by
$$(X, Z, y) \leftarrow (X + \alpha\,dX, Z + \beta\,dZ, y + \beta\,dy)\ .$$

As with Algorithms 3.1 and 3.2, we can use Algorithm 4.1 with $\sigma = 0.25$ til the residual norms $\|r_p\|_2$ and $\|r_d\|_2$ become relatively small, in which case we use Algorithm 4.2. But we choose the value of τ to be between 0.9 and 0.99, more conservative than Algorithms 3.1 and 3.2 (cf. [3, 27]).

4.2 Variations of the TTT Methods

Todd, Toh and Tütüncü [27] showed that the Schur complement equation (2.5-a) can be expressed as the normal equation of a linear least squares problem. In fact, let $\widetilde{\mathcal{D}}$ be an $n(n+1)/2$-by-$n(n+1)/2$ diagonal matrix that satisfies

$$\widetilde{\mathcal{D}}\ \mathbf{e} = \mathbf{svec}\left(\frac{2\,\beta_i\,\beta_j}{\sqrt{\left(\beta_i^2\,\sigma_i^2+\beta_j^2\,\sigma_j^2\right)\ \left(\beta_i^2+\beta_j^2\right)}}\right)$$

for the vector $\mathbf{e}$ in §1.4. With equation (4.5) and some algebra similar to that used to obtain (4.6), we can write $\mathcal{A}\,\mathcal{E}^{-1} = \widetilde{\mathcal{A}}\,\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\,\widetilde{\mathcal{D}}$. Let $X_r \in \mathbf{S}^n$ be a symmetric matrix such that

$$\widetilde{\mathcal{A}}\ \mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\ \mathbf{svec}(X_r) = r_p\ . \tag{4.10}$$

Then the Schur complement equation (2.5-a) can be rewritten as

$$\left(\widetilde{\mathcal{A}}\,\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\right)\cdot\left(\widetilde{\mathcal{A}}\,\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\right)^T\cdot dy = \left(\widetilde{\mathcal{A}}\,\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\right)\cdot\left(\mathbf{svec}(X_r)+\widetilde{\mathcal{D}}\ (\mathcal{F}\,r_d - r_c)\right)\ ,$$

which is the *normal equation* for the least squares problem

$$\min_{dy}\left\|\left(\widetilde{\mathcal{A}}\,\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\right)^T\cdot dy - \left(\mathbf{svec}(X_r)+\widetilde{\mathcal{D}}\ (\mathcal{F}\,r_d - r_c)\right)\right\|_2\ . \tag{4.11}$$

Hence dy is the solution to the least squares problem (4.11), and can be computed by standard methods for solving least squares problems, which are both efficient and backward stable (see, for example, Golub and van Loan [10, Ch. 5]).

Similar to Algorithms 4.1 and 4.2, the main cost of the least squares approach is in explicitly computing and factorizing the coefficient matrix $\widetilde{\mathcal{A}}\,\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}$. As in Algorithm 4.1, the cost for computing $\widetilde{\mathcal{A}}\,\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}$ is about $3mn^3$ flops. If the least squares problem (4.11) is solved by computing the QR factorization of $\widetilde{\mathcal{A}}\,\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}$, then the cost of this factorization is about $m^2n^2 - 2/3m^3$ flops. Hence the total per step cost of the least squares approach is about $m^2n^2 + 3mn^3 - 2/3m^3$ flops, roughly twice the per step cost of Algorithms 4.1 and 4.2 if $n \ll m \ll n^2$.

Since reliable methods for computing the SVD are only available for dense matrices, a potential drawback with Algorithms 4.1 and 4.2 is that the SVD computation in (2.9) could be very inefficient if matrices R and H were highly sparse or structured. Zhang [32] pointed out that equations (2.5) can be solved without the SVD (2.9) in the special case $P^T\,P = Z$, which corresponds to $B = I$ in (2.10). To see how this is done, choose $S = V$ and $B = I$ in (2.10-a) to get $P = H$. It follows from (4.2-a) that

$$\mathcal{E} = H \otimes_s H \quad \text{and} \quad \mathcal{F} = (H\ X) \otimes_s H^{-T}\ .$$

According to properties of the symmetrized Kronecker products in §1.4, we have

$$\mathcal{E}^{-1} = H^{-1} \otimes_s H^{-1} \quad \text{and} \quad \mathcal{E}^{-1}\,\mathcal{F} = \left(H^{-1}\ H\ X\right) \otimes_s \left(H^{-1}\ H^{-T}\right) = X \otimes_s Z^{-1}\ . \tag{4.12}$$

In view of (2.8), the Schur complement now becomes

$$\mathcal{M} = \left(\mathbf{svec}(A_i)^T\ \left(X \otimes_s Z^{-1}\right)\ \mathbf{svec}(A_j)\right) = \frac{1}{2}\ \left(A_i \bullet \left(X\ A_j\ Z^{-1} + Z^{-1}\ A_j\ X\right)\right)\ .$$

On the other hand, equations (2.8), (2.9) and (4.3) imply that

$$r_c = \mathbf{svec}\left(\mu\, I - V\, \Sigma^2\, V^T\right) = \mathbf{svec}\left(\mu\, I - H\, X\, H^T\right) .$$

Plugging this equation and (4.12) into (2.5), we get

$$\begin{aligned} \mathcal{M}\, dy &= r_p + \mathcal{A}\, \left(\left(X \otimes_s Z^{-1}\right)\, r_d - \mathbf{svec}\left(\mu\, Z^{-1} - X\right)\right) \\ dZ &= \mathbf{smat}\left(r_d - \mathcal{A}^T\, dy\right) \\ dX &= \mu\, Z^{-1} - X - \left(X \otimes_s Z^{-1}\right)\, \mathbf{svec}(dZ) . \end{aligned}$$

Since the matrices $X\, A_i\, Z^{-1}$ are in general non-symmetric, $\mathcal{M}$ in the above forms is slightly more expensive to compute than in (4.6). Since the cost for Cholesky factorizing $\mathcal{M}$ is the same in both cases, computing the search direction without the SVD is slightly more expensive than with it, if we neglect sparsity and structure considerations. Predictor-corrector modifications of both Mehrotra type and Mizuno-Todd-Ye type can be developed for this approach to accelerate convergence (see Mizuno, Todd and Ye [16] and Zhang [32]).

4.3 Preliminary Analysis

To motivate our error analysis of Algorithms 4.1 and 4.2, in §4.3 we examine equation (2.1-c) in *exact arithmetic*. Since

$$P\, dX\, \left(Z\, P^{-1}\right) = S\, B\, \widetilde{H}\, dX\, \widetilde{H}^T\, B^{-1}\, S^T \quad \text{and} \quad (P\, X)\, dZ\, P^{-1} = S\, \widetilde{B}\, \widetilde{R}\, dZ\, \widetilde{R}^T\, \widetilde{B}^{-1}\, S^T ,$$

and since $r_c = \mathbf{svec}\left(\mu\, I - S\, \Sigma^2\, S^T\right)$, equation (2.1-c) simplifies to

$$\mathbf{H}_B\left(\widetilde{H}\, dX\, \widetilde{H}^T\right) + \mathbf{H}_{\widetilde{B}}\left(\widetilde{R}\, dZ\, \widetilde{R}^T\right) = \mu I - \Sigma^2 .$$

With (4.1), this can be further written as

$$\frac{1}{2}\cdot\left(\left(\frac{\beta_i}{\beta_j} + \frac{\beta_j}{\beta_i}\right)\cdot\left(\widetilde{H}\, dX\, \widetilde{H}^T\right)_{i,j}\right) + \frac{1}{2}\cdot\left(\left(\frac{\beta_i\, \sigma_i}{\beta_j\, \sigma_j} + \frac{\beta_j\, \sigma_j}{\beta_i\, \sigma_i}\right)\cdot\left(\widetilde{R}\, dZ\, \widetilde{R}^T\right)_{i,j}\right) = \mu I - \Sigma^2 . \qquad (4.13)$$

Some of the above scaling factors involving β_i's and σ_i's can be *arbitrarily* large for very ill-conditioned B. In addition, the matrices $\widetilde{H}$ and $\widetilde{R}$ themselves could be badly scaled as well. To see this, assume for the moment that X and Z commute so that we can write their eigendecompositions as $X = Q\, \Lambda_X\, Q^T$ and $Z = Q\, \Lambda_Z\, Q^T$, where Q is an orthogonal matrix and both Λ_X and Λ_Z are positive diagonal matrices. Equation (2.8) implies that there exist orthogonal matrices W_X and W_Z such that

$$R = W_X\, \Lambda_X^{\frac{1}{2}}\, Q^T \quad \text{and} \quad H = W_Z\, \Lambda_Z^{\frac{1}{2}}\, Q^T , \quad \text{or} \quad R\, H^T = W_X\, \left(\Lambda_X\, \Lambda_Z\right)^{\frac{1}{2}}\, W_Z^T .$$

By the definitions of the SVD in (2.9) and matrices $\widetilde{R}$ and $\widetilde{H}$ in (2.10), we get

$$\widetilde{R} = W_X^T\, R = \Lambda_X^{\frac{1}{2}}\, Q^T \quad \text{and} \quad \widetilde{H} = W_Z^T\, H = \Lambda_Z^{\frac{1}{2}}\, Q^T .$$

In other words, if X and Z commute, then $\widetilde{R}$ and $\widetilde{H}$ are row-scaled by their singular values. For X and Z that are close to the optimal solution (X^*, Z^*), some of these singular values will be very tiny. In practice, X are Z are usually not commutable but become more and more commutable as they converge to (X^*, Z^*), making $\widetilde{R}$ and $\widetilde{H}$ potentially badly scaled. To fully understand the scaling problem in (4.13), we rewrite $\widetilde{R}$ and $\widetilde{H}$ in scaled forms as

$$\widetilde{R} = \Psi\, \bar{R} \quad \text{and} \quad \widetilde{H} = \Phi\, \bar{H} \,, \tag{4.14}$$

where $\Psi = \mathbf{diag}(\psi_1, \cdots, \psi_n)$ and $\Phi = \mathbf{diag}(\phi_1, \cdots, \phi_n)$ are chosen so that rows of $\bar{R}$ and $\bar{H}$ all have 2-norm 1. Hence equation (4.13) becomes

$$\frac{1}{2} \cdot \left(\left(\frac{\beta_i}{\beta_j} + \frac{\beta_j}{\beta_i} \right) \phi_i\, \phi_j \cdot \left(\bar{H}\, dX\, \bar{H}^T \right)_{i,j} \right) + \frac{1}{2} \cdot \left(\left(\frac{\beta_i\, \sigma_i}{\beta_j\, \sigma_j} + \frac{\beta_j\, \sigma_j}{\beta_i\, \sigma_i} \right) \psi_i\, \psi_j \cdot \left(\bar{R}\, dZ\, \bar{R}^T \right)_{i,j} \right) = \mu I - \Sigma^2. \tag{4.15}$$

The ratios involving β_i's and σ_i's could be huge, but the factors involving ϕ_i's and ψ_i's could be tiny. There are $n(n{+}1)/2$ scalar equations in (4.15). Some of them might have huge coefficients whereas others might *only* have tiny ones. This bad scaling could cause the matrix $\mathcal{J}$ in (2.2) to be very ill-conditioned, even when the optimal solution to (1.1) is well-conditioned. This bad scaling problem can be avoided by dividing the (i, j) entry in the matrix equation by $\left(\frac{\beta_i}{\beta_j} + \frac{\beta_j}{\beta_i} \right) \phi_i\, \phi_j + \left(\frac{\beta_i\, \sigma_i}{\beta_j\, \sigma_j} + \frac{\beta_j\, \sigma_j}{\beta_i\, \sigma_i} \right) \psi_i\, \psi_j$. We will discuss the scaling issue in finite precision in §4.4.

We now discuss the round-off errors in the following operations required in Algorithms 4.1 and 4.2:

$$\mathcal{D}\ (U \otimes_s U)\ \mathbf{svec}(A) = \mathbf{svec}\left(D \odot \left(U\, A\, U^T \right) \right) \quad \text{and} \quad (U \otimes_s U)^{-1}\ \mathbf{svec}(A) = \mathbf{svec}\left(U^{-1}\, A\, U^{-T} \right),$$

where A and $D \in \mathbf{S}^n$; and $\mathcal{D}$ is an $n(n+1)/2$-by-$n(n+1)/2$ diagonal matrix such that $\mathcal{D}\,\mathbf{e} = \mathbf{svec}(D)$. We summarize their computations in Algorithms 4.3 and 4.4 and their error analysis in Lemma 4.1. We leave the proof of Lemma 4.1 to Appendix A.

Algorithm 4.3 Computing $\mathcal{V} = \mathcal{D}\ (U \otimes_s U)\ \mathbf{svec}(A)$

```
1. Compute the matrix-matrix product $U\, A$.
2. Compute the $(i, j)$ and $(j, i)$ entries of $U\, A\, U^T$ as the sum $\sum_{k=1} (\mathbf{fl}(U\, A))_{ik}\, U_{jk}$.
3. Compute $\mathbf{fl}\left( D \odot \mathbf{fl}\left( U\, A\, U^T \right) \right)$.
```

Algorithm 4.4 Computing $\mathcal{W} = (U \otimes_s U)^{-1}\ \mathbf{svec}(A)$

```
1. Factorize $U$ with an efficient and backward stable method.
2. Let $A = (a_1, \cdots, a_n)$. Compute $U^{-1} A$ by solving $n$ linear systems of equations $U\, v_i = a_i$.
3. Let $\mathbf{fl}\,(U^{-1} A) = (\widetilde{v}_1, \cdots, \widetilde{v}_n)^T$. Compute $U^{-1}\, A\, U^{-T}$ by solving $n$ linear systems of equations
   $\widetilde{w}_i^T\, U^T = \widetilde{v}_i^T$ and symmetrizing $(\widetilde{w}_1, \cdots, \widetilde{w}_n)$.
```

Algorithm 4.4 is not very efficient since it does not take advantage of the symmetry in $U^{-1}\, A\, U^{-T}$. This extra cost can be avoided with a more involved algorithm and is tiny comparing to other costs in Algorithms 4.1 and 4.2. To achieve good accuracy in computing (4.9) for Algorithms 4.1 and 4.2, we rewrite it using (4.14) as

$$\begin{aligned} &\mathcal{E}^{-1} \left(r_c - \mathcal{F} \mathbf{svec}(dZ) \right) \\ &\quad = \mathbf{svec} \left(\bar{H}^{-1} \Phi^{-1} \left(\bar{D}_{\mathcal{E}} \odot \mathbf{smat}\,(\widetilde{r}_c) - (\bar{D}_{\mathcal{E}} \odot D_{\mathcal{F}}) \odot \left(\widetilde{R}\, dZ\, \widetilde{R}^T \right) \right) \Phi^{-1}\, \bar{H}^{-T} \right) , \end{aligned} \tag{4.16}$$

and compute (4.16) using Algorithm 4.4 with $U = \bar{H}$.

Lemma 4.1 *Let $\widehat{\mathcal{V}}$ and $\widehat{\mathcal{W}}$ be the computed counterparts of $\mathcal{V}$ and $\mathcal{W}$ in Algorithms 4.3 and 4.4, respectively and assume that $\kappa(U) \ll 1/\sqrt{\epsilon}$ in Algorithm 4.4. Then there exist $n(n+1)/2$-by-$n(n+1)/2$ matrices Θ_1 and Θ_2 such that*

$$\widehat{\mathcal{V}} = \mathcal{D}\ (U \otimes_s U + \Theta_1)\ \mathbf{svec}(A) \quad \textit{and} \quad (U \otimes_s U + \Theta_2) \cdot \widehat{\mathcal{W}} = \mathbf{svec}(A)\,,$$

where $|\Theta_1| \le O(\epsilon) \cdot (|U| \otimes_s |U|)$ and $\|\Theta_2\|_2 \le O\left(\epsilon \cdot \|U\|_2^2\right)$.

4.4 Error Analysis for the TTT Methods

The error analysis for the TTT methods is much more complicated than that for the AHO method, due to the potentially bad scaling of the complementarity equation (2.1-c) for the TTT methods. To shorten the presentation, we will summarize some technical pieces of analysis into lemmas and discuss them in Appendix B. As in §3.3, we will focus on Algorithm 4.1.

We begin by examining the round-off errors in the decompositions (2.8) and the SVD (2.9). Assume that they are computed backward stably as

$$\widehat{R}^T\,\widehat{R} = X + O(\epsilon\cdot\|X\|_2)\,,\quad \widehat{H}^T\,\widehat{H} = Z + O(\epsilon\cdot\|Z\|_2)\quad\text{and}\quad \widehat{R}\,\widehat{H}^T = \widehat{W}\,\widehat{\Sigma}\,\widehat{V}^T + O\left(\epsilon\cdot\|\widehat{R}\|_2\,\|\widehat{H}\|_2\right)\,,$$

where $\widehat{W}$ and $\widehat{V}$ are nearly orthogonal matrices satisfying $\widehat{W}^T\;\widehat{W} = I + O(\epsilon)$ and $\widehat{V} = I + O(\epsilon)$, respectively; and $\widehat{\Sigma} = \mathbf{diag}\,(\widehat{\sigma}_1,\cdots,\widehat{\sigma}_n)$. Let $\widetilde{R}$ and $\widetilde{H}$ be computed as

$$\mathbf{fl}(\widetilde{R}) = \widehat{W}^T\,\widehat{R} + O(\epsilon\cdot\|\widetilde{R}\|_2)\quad\text{and}\quad \mathbf{fl}(\widetilde{H}) = \widehat{V}^T\,\widehat{H} + O(\epsilon\cdot\|\widetilde{H}\|_2)\,.$$

We define

$$X^\dagger \overset{\text{def}}{=} \left(\mathbf{fl}(\widetilde{R})\right)^T\,\mathbf{fl}(\widetilde{R}) = X + O(\epsilon\cdot\|X\|_2)\quad\text{and}\quad Z^\dagger \overset{\text{def}}{=} \left(\mathbf{fl}(\widetilde{H})\right)^T\,\mathbf{fl}(\widetilde{H}) = Z + O(\epsilon\cdot\|Z\|_2)\,.$$

To make the notation less cluttered, in the remainder of this section, we will drop the symbol **fl** in $\mathbf{fl}(\widetilde{R})$ and $\mathbf{fl}(\widetilde{H})$ and replace them by $\widetilde{R}$ and $\widetilde{H}$, respectively. Combine the above equations to get

$$X^\dagger = \widetilde{R}^T\,\widetilde{R}\,,\quad Z^\dagger = \widetilde{H}^T\,\widetilde{H}\quad\text{and}\quad \widetilde{R}\,\widetilde{H}^T = \widehat{\Sigma} + O\left(\epsilon\cdot\|\widetilde{R}\|_2\,\|\widetilde{H}\|_2\right) \overset{\text{def}}{=} \widehat{\Sigma} + E\,. \tag{4.17}$$

In our analysis, we will think of the search direction defined by (2.2) as a direction defined at the point $(X^\dagger, Z^\dagger, y)$, instead of (X, Z, y). These two points are identical in exact arithmetic, and differ slightly in finite precision. However, this minor difference will make our analysis much simpler. Since the round-off error matrix E in (4.17) is in general non-zero, the expressions in (2.10) for P and the expressions in (4.2) for $\mathcal{E}$ and $\mathcal{F}$, while mathematically equivalent in exact arithmetic, are *inconsistent* in finite precision arithmetic. As in §4.1, we base our analysis on the assumption that P is chosen using (2.10-a): $P = S\;B\;\widetilde{H}$. We also set $S = I$ since it is never involved in any computation (see §4.1). Under this choice of P, the search direction defined by (2.2) at the point $(X^\dagger, Z^\dagger, y)$ satisfies (cf. (4.2-a))

$$\mathcal{J}\;d\mathcal{X} = \mathcal{R}\,,\qquad\qquad \text{where}\quad \mathcal{J} = \begin{pmatrix} \mathcal{E} & \mathcal{F} & 0 \\ 0 & \mathcal{I} & \mathcal{A}^T \\ \mathcal{A} & 0 & 0 \end{pmatrix}\quad\text{and}\quad \mathcal{R} = \begin{pmatrix} r_c \\ r_d \\ r_p \end{pmatrix}\,, \tag{4.18}$$

with $\mathcal{E} = \left(B^{-1}\;\widetilde{H}\right)\otimes_s\left(B\;\widetilde{H}\right)$, $\mathcal{F} = \left(B\;\widetilde{H}\;X^\dagger\right)\otimes_s\left(B^{-1}\;\widetilde{H}^{-T}\right)$, $r_c = \mathbf{svec}\left(\mu\,I - \mathbf{H}_B\left(\widetilde{H}\,X^\dagger\,\widetilde{H}^T\right)\right)$, $r_d = \mathbf{svec}\left(C - Z^\dagger - \mathbf{smat}\left(\mathcal{A}^T\,y\right)\right)$ and $r_p = b - \mathcal{A}\,\mathbf{svec}(X^\dagger)$.

We could try to write the round-off errors during the computation of the search direction as perturbations to (4.18), in a form similar to (3.9). However, the coefficient matrix $\mathcal{J}$ in (4.18) is in general badly scaled and hence ill-conditioned. To make our error analysis meaningful, we need to re-scale the rows of $\mathcal{J}$ to make it balanced, and then examine the error bounds in the re-scaled version of (4.18). In this section, in addition to Assumptions 3.1 through 3.4, we further assume that

Assumption 4.1 *The error matrix E in (4.17) satisfies $\|E\|_2 \le \min_{i=1}^n \widehat{\sigma}_i/2$.*

Assumption 4.2 *The matrix $\bar{H}$ defined in (4.14) satisfies $\kappa(\bar{H}) \ll 1/\sqrt{\epsilon}$.*

We start our analysis by revealing the bad scaling in the matrix $\mathcal{J}$ in (4.18). Rewrite $\mathcal{E}$ and $\mathcal{F}$ according to (4.14) and (4.17),

$$\begin{aligned}
\mathcal{E} &= \left(\left(B^{-1}\,\Phi\right)\otimes_s\left(B\,\Phi\right)\right)\cdot\left(\bar{H}\otimes_s\bar{H}\right)\\
\mathcal{F} &= \left(B\,\left(\widehat{\Sigma}+E\right)^T\,\widetilde{R}\right)\otimes_s\left(B^{-1}\,\left(\widehat{\Sigma}+E\right)^{-1}\widetilde{R}\right)\\
&= \left(\left(B\,\widehat{\Sigma}\right)\cdot\left(I+E\,\widehat{\Sigma}^{-1}\right)^T\,\widetilde{R}\right)\otimes_s\left(\left(B\,\widehat{\Sigma}\right)^{-1}\left(I+E\,\widehat{\Sigma}^{-1}\right)^{-1}\widetilde{R}\right).
\end{aligned}$$

Since the matrix $I+E\,\widehat{\Sigma}^{-1}$ is in general dense and has 2-norm $\Omega(1)$ (see (4.17) and Assumption 4.1), we can choose the diagonal scaling matrices for $\mathcal{E}$ and $\mathcal{F}$ to be

$$\mathcal{S}_{\mathcal{E}}\overset{\text{def}}{=}\left(B^{-1}\,\Phi\right)\otimes_s\left(B\,\Phi\right)\quad\text{and}\quad\mathcal{S}_{\mathcal{F}}\overset{\text{def}}{=}(\phi+\psi)^2\cdot\left(B\,\widehat{\Sigma}\right)\otimes_s\left(B\,\widehat{\Sigma}\right)^{-1},$$

respectively, where $\phi=\max_{i=1}^n\phi_i=\Omega(\|\widetilde{H}\|_2)$ and $\psi=\max_{i=1}^n\psi_i=\Omega(\|\widetilde{R}\|_2)$. The scaled $\mathcal{E}$ matrix $\mathcal{S}_{\mathcal{E}}^{-1}\,\mathcal{E}=\bar{H}\otimes_s\bar{H}$ is well row-scaled due to (4.14) and has 2-norm $\Omega(1)$. The scaled $\mathcal{F}$ matrix $\mathcal{S}_{\mathcal{F}}^{-1}\,\mathcal{F}$ has 2-norm $O(1)$, but could still be badly row-scaled for some E. We have chosen the factor $(\phi+\psi)^2$ instead of ψ^2 in front of $\mathcal{S}_{\mathcal{F}}$ to make our analysis simpler. To see the diagonal entries of $\mathcal{S}_{\mathcal{E}}$ and $\mathcal{S}_{\mathcal{F}}$ more clearly, we apply $\mathcal{S}_{\mathcal{E}}$ and $\mathcal{S}_{\mathcal{F}}$ to the vector $\mathbf{e}$ in §1.4:

$$\mathcal{S}_{\mathcal{E}}\,\mathbf{e}=\frac{1}{2}\cdot\mathbf{svec}\left(\left(\frac{\beta_i}{\beta_j}+\frac{\beta_j}{\beta_i}\right)\,\phi_i\,\phi_j\right)\quad\text{and}\quad\mathcal{S}_{\mathcal{F}}\,\mathbf{e}=\frac{(\phi+\psi)^2}{2}\cdot\mathbf{svec}\left(\frac{\beta_i\,\widehat{\sigma}_i}{\beta_j\,\widehat{\sigma}_j}+\frac{\beta_j\,\widehat{\sigma}_j}{\beta_i\,\widehat{\sigma}_i}\right).$$

Comparing with (4.15), which is the complementarity equation in exact arithmetic, we see that the scaling factors for $\mathcal{E}$ in finite precision arithmetic is similar to those for $\mathcal{E}$ in exact arithmetic. On the other hand, some of the ψ_i's can be much smaller than $\phi+\psi$, so the scaling factors for $\mathcal{F}$ in finite precision arithmetic can be drastically larger than those for $\mathcal{F}$ in exact arithmetic.

We now re-scale $\mathcal{J}$ in (4.18) with $\mathcal{S}_{\mathcal{E}}$ and $\mathcal{S}_{\mathcal{F}}$ to get

$$\mathcal{J}_{\mathcal{S}}\,d\mathcal{X}=\mathcal{R}_{\mathcal{S}}\,,\qquad\text{where}\quad\mathcal{J}_{\mathcal{S}}=\begin{pmatrix}\mathcal{S}^{-1}\,\mathcal{E} & \mathcal{S}^{-1}\,\mathcal{F} & 0\\ 0 & \mathcal{I} & \mathcal{A}^T\\ \mathcal{A} & 0 & 0\end{pmatrix}\quad\text{and}\quad\mathcal{R}_{\mathcal{S}}=\begin{pmatrix}\mathcal{S}^{-1}\,r_c\\ r_d\\ r_p\end{pmatrix},$$

with $\mathcal{S}=\mathcal{S}_{\mathcal{E}}+\mathcal{S}_{\mathcal{F}}$. Let $\widehat{\mathcal{R}}_{\mathcal{S}}$ be the vector $\widehat{\mathcal{R}}$ with $\widehat{r}_c$ replaced by $\mathcal{S}^{-1}\,\widehat{r}_c$. Scale equation (3.9) to get

$$\left(\mathcal{J}_{\mathcal{S}}+\delta\mathcal{J}_{\mathcal{S}}\right)\,\widehat{d\mathcal{X}}=\widehat{\mathcal{R}}_{\mathcal{S}}\,. \tag{4.19}$$

We point out that $\mathcal{S}$ is introduced as part of our error analysis, and is *not* part of Algorithm 4.1. As the scaling factors for $\mathcal{F}$ in finite precision arithmetic and in exact arithmetic can be drastically different, $\mathcal{J}_{\mathcal{S}}$ could still be ill-conditioned (see §5).

We now consider round-off errors in $\mathcal{R}$. Although numerically $\mathcal{R}$ is evaluated at the point (X,Z,y), instead of $(X^{\dagger},Z^{\dagger},y)$, the difference between them is minor. Equation (3.11) still holds for r_d and r_p:

$$\widehat{r}_d=r_d+O\left(\epsilon\cdot\|Z\|_2+\epsilon\cdot\|\mathcal{A}\|_2\,\|y\|_2\right)\quad\text{and}\quad\widehat{r}_p=r_p+O\left(\epsilon\cdot\|\mathcal{A}\|_2\,\|X\|_2\right). \tag{4.20}$$

Since the round-off errors in $\widehat{r}_c$ are more complicated, we summarize the results here and leave the analysis to Appendix B.

Lemma 4.2 *In Algorithm 4.1,*

$$\left\|\mathcal{S}^{-1}\,\left(\widehat{r}_c-r_c\right)\right\|_2=\begin{cases}O\left(\epsilon\cdot\|\widetilde{R}\|_2\,\|\widetilde{H}\|_2\right), & \textit{for HKM direction } P^T\,P=Z \textit{ and NT direction;}\\[2ex] O\left(\epsilon\cdot\kappa\left(\widehat{\Sigma}\right)\,\|\widetilde{R}\|_2\,\|\widetilde{H}\|_2\right), & \textit{in general.}\end{cases}$$

Combining Lemma 4.2 with (4.20), we get

$$\|\widehat{\mathcal{R}}_{\mathcal{S}} - \mathcal{R}_{\mathcal{S}}\|_2 = O\left(\epsilon \cdot \kappa\left(\widehat{\Sigma}\right)\ \|\mathcal{X}\|_2\right) . \tag{4.21}$$

The factor $\kappa(\widehat{\Sigma})$ disappears for the HKM direction $P^T P = Z$ and the NT direction. Since Algorithm 4.1 usually generates iterates that are not far away from the central path, the factor $\kappa(\widehat{\Sigma})$ is in general not very large in practice.

We now analyze the round-off errors in computing the right hand sides of (2.5). To this end, define

$$\mathcal{E}^\dagger = \left(B\,\widehat{\Sigma}\,\widetilde{R}^{-T}\right) \otimes_s \left(B^{-1}\,\widehat{\Sigma}\,\widetilde{R}^{-T}\right),\ \mathcal{F}^\dagger = \left(B\,\widehat{\Sigma}\,\widetilde{R}\right) \otimes_s \left(B^{-1}\,\widehat{\Sigma}^{-1}\,\widetilde{R}\right),\ \mathcal{M}^\dagger = \mathcal{A}\,\left(\mathcal{E}^\dagger\right)^{-1}\,\mathcal{F}^\dagger\mathcal{A}^T. \tag{4.22}$$

Although $\mathcal{E}^\dagger = \mathcal{E}$, $\mathcal{F}^\dagger = \mathcal{F}$, and $\mathcal{M} = \mathcal{M}^\dagger$ in exact arithmetic, these relations do not in general hold in finite arithmetic. Let

$$\widehat{D}_{\mathcal{F}} = \left(\frac{\beta_i^2\,\widehat{\sigma}_i^2 + \beta_j^2\,\widehat{\sigma}_j^2}{\beta_i\,\beta_j\,\widehat{\sigma}_i\,\widehat{\sigma}_j}\right)\ ,\quad \widehat{D}_{\mathcal{E}} = \left(\frac{\beta_i\,\beta_j}{\sigma_i\,\sigma_j\left(\beta_i^2 + \beta_j^2\right)}\right) \quad \text{and} \quad \widehat{D}_{\mathcal{M}} = \widehat{D}_{\mathcal{F}} \odot \widehat{D}_{\mathcal{E}}\ ,$$

and define diagonal matrices $\widehat{\mathcal{D}}_{\mathcal{E}}$, $\widehat{\mathcal{D}}_{\mathcal{F}}$ and $\widehat{\mathcal{D}}_{\mathcal{M}}$ such that

$$\widehat{\mathcal{D}}_{\mathcal{F}}\,\mathbf{e} = \mathbf{svec}\left(\widehat{D}_{\mathcal{F}}\right)\ ,\quad \widehat{\mathcal{D}}_{\mathcal{E}}\,\mathbf{e} = \mathbf{svec}\left(\widehat{D}_{\mathcal{E}}\right) \quad \text{and} \quad \widehat{\mathcal{D}}_{\mathcal{M}}\,\mathbf{e} = \mathbf{svec}\left(\widehat{D}_{\mathcal{M}}\right)\ .$$

The matrix-vector product $\mathbf{fl}\left(\mathcal{F}^\dagger\,r_d\right)$ has the form in Algorithm 4.3 with $\mathcal{D} = \widehat{\mathcal{D}}_{\mathcal{F}}$ and $U = \widetilde{R}$. According to Lemma 4.1, the round-off errors satisfy

$$\mathbf{fl}\left(\mathcal{F}^\dagger\,\widehat{r}_d\right) = \left(\mathcal{F}^\dagger + \delta_1\mathcal{F}^\dagger\right)\,\widehat{r}_d\ , \quad \text{where} \quad \left|\delta_1\mathcal{F}^\dagger\right| \le O(\epsilon)\cdot\widehat{\mathcal{D}}_{\mathcal{F}}\cdot\left(\left|\widetilde{R}\right| \otimes_s \left|\widetilde{R}\right|\right)\ .$$

As in §3.3, there exists an $n(n+1)/2$-by-$n(n+1)/2$ diagonal perturbation matrix $\Delta_1 = O(\epsilon)$ such that

$$\mathbf{fl}\left(\mathcal{F}^\dagger\,\widehat{r}_d - \widehat{r}_c\right) = (\mathcal{I} + \Delta_1)\,\left(\left(\mathcal{F}^\dagger + \delta_1\mathcal{F}^\dagger\right)\,\widehat{r}_d - \widehat{r}_c\right)\ .$$

The application of $\left(\mathcal{E}^\dagger\right)^{-1}$ to $\mathcal{F}^\dagger\,\widehat{r}_d - \widehat{r}_c$ in (4.8) can be performed by applying $\left(\widetilde{R}^T \otimes_s \widetilde{R}^T\right)\widehat{\mathcal{D}}_{\mathcal{E}}$ to $\mathbf{fl}\left(\mathcal{F}^\dagger\,\widehat{r}_d - \widehat{r}_c\right)$. Similar to Lemma 4.1, the round-off errors satisfy

$$\begin{aligned}
\mathbf{fl}\left(\left(\mathcal{E}^\dagger\right)^{-1}\,\left(\mathcal{F}^\dagger\,\widehat{r}_d - \widehat{r}_c\right)\right) &= \left(\left(\mathcal{E}^\dagger\right)^{-1} + \Theta_1\right)\,\mathbf{fl}\left(\mathcal{F}^\dagger\,\widehat{r}_d - \widehat{r}_c\right),\ |\Theta_1| \le O(\epsilon)\cdot\left(\left|\widetilde{R}^T\right| \otimes_s \left|\widetilde{R}^T\right|\right)\cdot\widehat{\mathcal{D}}_{\mathcal{E}} \\
&= \left(\left(\mathcal{E}^\dagger\right)^{-1} + \Theta_2\right)\,\left(\left(\mathcal{F}^\dagger + \delta_1\mathcal{F}^\dagger\right)\,\widehat{r}_d - \widehat{r}_c\right)\ ,
\end{aligned}$$

$$\text{where} \quad \Theta_2 \overset{\text{def}}{=} \left(\left(\mathcal{E}^\dagger\right)^{-1} + \Theta_1\right)\,(\mathcal{I} + \Delta_1) - \left(\mathcal{E}^\dagger\right)^{-1} \quad \text{satisfies} \quad |\Theta_2| \le O(\epsilon)\cdot\left(\left|\widetilde{R}^T\right| \otimes_s \left|\widetilde{R}^T\right|\right)\widehat{\mathcal{D}}_{\mathcal{E}}\ .$$

With these relations, we can write

$$\begin{aligned}
&\mathbf{fl}\left(\widehat{r}_p + \mathcal{A}\,\left(\mathcal{E}^\dagger\right)^{-1}\,\left(\mathcal{F}^\dagger\,\widehat{r}_d - \widehat{r}_c\right)\right) = (\mathcal{I} + \Delta_2)\,\left(\widehat{r}_p + (\mathcal{A} + \delta_1\mathcal{A})\,\mathbf{fl}\left(\left(\mathcal{E}^\dagger\right)^{-1}\,\left(\mathcal{F}^\dagger\,\widehat{r}_d - \widehat{r}_c\right)\right)\right) \\
&= (\mathcal{I} + \Delta_2)\,\left(\widehat{r}_p + (\mathcal{A} + \delta_1\mathcal{A})\,\left(\left(\mathcal{E}^\dagger\right)^{-1} + \Theta_2\right)\,\left(\left(\mathcal{F}^\dagger + \delta_1\mathcal{F}^\dagger\right)\,\widehat{r}_d - \widehat{r}_c\right)\right)\ ,
\end{aligned} \tag{4.23}$$

where $\delta_1\mathcal{A} = O(\epsilon\cdot\|\mathcal{A}\|_2)$ and $\Delta_2 = O(\epsilon)$ are perturbation matrices, with Δ_2 being diagonal. The round-off errors in solving (2.5-c) are analyzed by Lemma 4.3 below.

Lemma 4.3 *The round-off errors in solving (2.5-c) satisfy*

$$\left(\mathcal{M}^\dagger + \delta\mathcal{M}^\dagger\right)\ \widehat{dy} = \mathbf{fl}\left(\widehat{r}_p + \mathcal{A}\,\mathcal{E}^{-1}\ (\mathcal{F}\,\widehat{r}_d - \widehat{r}_c)\right), \qquad \delta\mathcal{M}^\dagger = O\left(\epsilon\cdot\|\mathcal{A}\|_2^2\ \left\|X^\dagger\right\|_2\ \left\|\left(Z^\dagger\right)^{-1}\right\|_2\right)\ .$$

A remarkable feature of Lemma 4.3 is that the upper bound on $\delta\mathcal{M}^\dagger$ does not depend on B. Hence the Schur complement equation is solved to the same accuracy no matter how badly the complementarity equation (2.1-c) is scaled (see (4.15)). As in §3.3, the round-off errors in equation (2.5-b) satisfy

$$\widehat{dZ} = \mathbf{smat}\left((\mathcal{I}+\Delta_3)\ \left(\widehat{r}_d - (\mathcal{A}+\delta_2\mathcal{A})^T\ \widehat{dy}\right)\right)\ ,$$

where $\delta_1\mathcal{A} = O(\epsilon\cdot\|\mathcal{A}\|_2)$ and $\Delta_2 = O(\epsilon)$ are perturbation matrices, with Δ_2 being diagonal.

Now we consider the round-off errors in solving equation (2.5-c) using (4.16). Similar to (4.23),

$$\mathbf{fl}\left(\widehat{r}_c - \mathcal{F}^\dagger\,\mathbf{svec}\left(\widehat{dZ}\right)\right) = (\mathcal{I}+\Delta_4)\ \left(\widehat{r}_c - \left(\mathcal{F}^\dagger + \delta_2\mathcal{F}^\dagger\right)\ \mathbf{svec}\left(\widehat{dZ}\right)\right)\ ,$$

where $\Delta_4 = O(\epsilon) \in \mathbf{R}^{m\times m}$ is diagonal and $\left|\delta_2\mathcal{F}^\dagger\right| \leq O(\epsilon)\cdot\widehat{\mathcal{D}}_{\mathcal{F}}\cdot\left(\left|\widetilde{R}\right|\otimes_s\left|\widetilde{R}\right|\right)$. By Assumption 4.2 and Lemma 4.1, we write the round-off errors in the solution of equation (2.5-c) as

$$\left(B^{-1}\ \Phi\otimes_s B\ \Phi\right)\ \left(\bar{H}\otimes_s\bar{H}+\Theta_3\right)\ \widehat{dX} = (\mathcal{I}+\Delta_4)\ \left(\widehat{r}_c - \left(\mathcal{F}^\dagger+\delta_2\mathcal{F}^\dagger\right)\ \mathbf{svec}\left(\widehat{dZ}\right)\right)\ ,$$

where $\|\Theta_3\|_2 = O\left(\epsilon\cdot\|\bar{H}\|_2^2\right) = O(\epsilon)$. Since Δ_4 is a diagonal matrix, this last equation becomes

$$(\mathcal{E}+\delta\mathcal{E})\ \widehat{dX} = \widehat{r}_c - \left(\mathcal{F}^\dagger+\delta_2\mathcal{F}^\dagger\right)\ \mathbf{svec}\left(\widehat{dZ}\right)\ , \tag{4.24}$$

$$\text{where}\qquad \delta\mathcal{E} \overset{\text{def}}{=} (\mathcal{I}+\Delta_4)^{-1}\ \left(B^{-1}\ \Phi\otimes_s B\ \Phi\right)\ \left(\bar{H}\otimes_s\bar{H}+\Theta_3\right) - \mathcal{E} = \left(B^{-1}\ \Phi\otimes_s B\ \Phi\right)\Theta_4\ ,$$

$$\text{with}\qquad \Theta_4 \overset{\text{def}}{=} (\mathcal{I}+\Delta_4)^{-1}\left(\bar{H}\otimes_s\bar{H}+\Theta_3\right) - \bar{H}\otimes_s\bar{H} = O(\epsilon)\ .$$

Putting Lemma 4.3 and all these relations together, we get an equation similar to (2.4),

$$\begin{pmatrix} \mathcal{E}+\delta\mathcal{E} & \mathcal{F}^\dagger+\delta_2\mathcal{F}^\dagger & 0 \\ 0 & (\mathcal{I}+\Delta_3)^{-1} & (\mathcal{A}+\delta_2\mathcal{A})^T \\ 0 & 0 & \mathcal{M}^\dagger+\delta\mathcal{M}^\dagger \end{pmatrix} \widehat{d\mathcal{X}} = \begin{pmatrix} \widehat{r}_c \\ \widehat{r}_d \\ \mathbf{fl}\left(\widehat{r}_p + \mathcal{A}\ \left(\mathcal{E}^\dagger\right)^{-1}\ \left(\mathcal{F}^\dagger\,\widehat{r}_d - \widehat{r}_c\right)\right) \end{pmatrix}\ ,$$

Combining this with (4.23), we obtain an equation similar to (2.3) and (3.16):

$$\begin{pmatrix} \mathcal{I} & 0 & 0 \\ 0 & \mathcal{I} & 0 \\ \mathcal{L}_{3,1} & \mathcal{L}_{3,2} & (\mathcal{I}+\Delta_2)^{-1} \end{pmatrix} \begin{pmatrix} \mathcal{E}+\delta\mathcal{E} & \mathcal{F}^\dagger+\delta_2\mathcal{F}^\dagger & 0 \\ 0 & (\mathcal{I}+\Delta_3)^{-1} & (\mathcal{A}+\delta_2\mathcal{A})^T \\ 0 & 0 & \mathcal{M}^\dagger+\delta\mathcal{M}^\dagger \end{pmatrix} \widehat{d\mathcal{X}} = \begin{pmatrix} \widehat{r}_c \\ \widehat{r}_d \\ \widehat{r}_p \end{pmatrix}\ ,$$

$$\text{where}\quad \mathcal{L}_{3,1} = (\mathcal{A}+\delta_1\mathcal{A})\ \left(\left(\mathcal{E}^\dagger\right)^{-1}+\Theta_2\right)\quad\text{and}\quad \mathcal{L}_{3,2} = -\mathcal{L}_{3,1}\ \left(\mathcal{F}^\dagger+\delta_1\mathcal{F}^\dagger\right)\ .$$

Scale the first row by $\mathcal{S}^{-1}$, we arrive at equation (4.19) with the backward error matrix $\delta\mathcal{J}_\mathcal{S}$ satisfying

$$\begin{aligned} \delta\mathcal{J}_\mathcal{S} &= \begin{pmatrix} \mathcal{S}^{-1} & 0 & 0 \\ 0 & \mathcal{I} & 0 \\ \mathcal{L}_{3,1} & \mathcal{L}_{3,2} & (\mathcal{I}+\Delta_2)^{-1} \end{pmatrix} \begin{pmatrix} \mathcal{E}+\delta\mathcal{E} & \mathcal{F}^\dagger+\delta_2\mathcal{F}^\dagger & 0 \\ 0 & (\mathcal{I}+\Delta_3)^{-1} & (\mathcal{A}+\delta_2\mathcal{A})^T \\ 0 & 0 & \mathcal{M}^\dagger+\delta\mathcal{M}^\dagger \end{pmatrix} - \mathcal{J}_\mathcal{S} \\ &= \begin{pmatrix} \mathcal{S}^{-1}\,\delta\mathcal{E} & \mathcal{S}^{-1}\left(\mathcal{F}^\dagger - \mathcal{F} + \delta_2\mathcal{F}^\dagger\right) & 0 \\ 0 & (\mathcal{I}+\Delta_3)^{-1}-\mathcal{I} & \delta_2\mathcal{A}^T \\ \mathcal{L}_{3,1}\,\mathcal{E}-\mathcal{A} & \mathcal{L}_{3,1}\ \left(\mathcal{F}^\dagger - \mathcal{F}^\dagger\ (\mathcal{I}+\Delta_3)^{-1}\right) & (\mathcal{I}+\Delta_2)^{-1}\ \mathcal{M}^\dagger + \mathcal{L}_{3,2}\ \mathcal{A}^T \end{pmatrix} \\ &\quad + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \mathcal{L}_{3,1}\cdot\delta\mathcal{E} & \mathcal{L}_{3,1}\ \left(\delta_2\mathcal{F}^\dagger - \delta_1\mathcal{F}^\dagger\ (\mathcal{I}+\Delta_3)^{-1}\right) & (\mathcal{I}+\Delta_2)^{-1}\ \delta\mathcal{M}^\dagger + \mathcal{L}_{3,2}\cdot\delta_2\mathcal{A}^T \end{pmatrix}. \end{aligned} \tag{4.25}$$

Lemma 4.4 *The matrix $\delta\mathcal{J}_S$ in (4.25) can be bounded as*

$$\delta\mathcal{J}_S = \begin{cases} O\left(\epsilon\cdot\rho\right), & \textit{for HKM direction } P^T\,P = Z \textit{ and NT direction;} \\ O\left(\epsilon\cdot\sqrt{\rho}\,\left(\kappa\left(\widehat{\Sigma}\right)+\sqrt{\rho}\right)\right), & \textit{in general.} \end{cases}$$

where $\rho = \left(\|X^{\dagger}\|_2+\|Z^{\dagger}\|_2\right)\left(\|\left(X^{\dagger}\right)^{-1}\|_2+\|\left(Z^{\dagger}\right)^{-1}\|_2\right)$.

As we argued after Lemma 4.2, the factor $\kappa(\widehat{\Sigma})$ is usually not very large in practice. For the sake of argument in the following we assume that it is less than $\sqrt{\rho}$. Now the bound in Lemma 4.4 looks like (3.17). With arguments similar to those in §3.3, we conclude that Algorithm 4.1 could stop making further progress as soon as it reaches an iterate (X, Z, y) that satisfies

$$\frac{\min\left(\lambda_{\min}(Z),\lambda_{\min}(X)\right)}{\max\left(\|Z\|_2,\|X\|_2\right)} = \Omega\left(\epsilon\cdot\kappa\left(\mathcal{J}_S\right)\right)\,, \tag{4.26}$$

and Algorithm 4.1 could be numerically unstable if $\mathcal{J}_S$ is ill-conditioned. As with the AHO method, by repeating the arguments in §4.4, it is easy to see that Algorithm 4.2 could also be numerically unstable if $\mathcal{J}_S$ is ill-conditioned.

If $\kappa(\widehat{\Sigma}) \gg 1$, then the error bound in equation (4.21) on the scaled right hand side of (2.2) will be large. We can eliminate the factor $\kappa(\widehat{\Sigma})$ in the error bound by choosing a scaling matrix $\mathcal{S}$ with larger diagonal entries, thereby making $\mathcal{J}_S$ potentially worse scaled and therefore worse conditioned. Similar considerations apply to Algorithm 4.2.

At first sight, equation (4.26) seems to suggest that the TTT methods could be as accurate as the AHO method. However, our numerical results in §5.3 show that the matrix $\mathcal{J}_S$ for the HKM methods and NT method is in general much worse conditioned than the matrix $\mathcal{J}$ for the AHO method, indicating that these methods are in general *less* accurate. In §5.1 we discuss a choice of B that appears to make $\mathcal{J}_S$ better conditioned than other choices.

The above analysis was on Algorithms 4.1 and 4.2 only. Since the NT method [23, 24] as implemented in [27] is not identical to Algorithms 4.1 and 4.2, our results do not directly apply to it. However, the difference between these variations does not appear to be fundamental. It is very likely that the NT method in [23, 24, 27] suffers from the same numerical instability problems Algorithms 4.1 and 4.2 face. The same argument holds for the HKM direction $P^T\,P = X^{-1}$.

While the HKM direction $P^T\,P = Z$ can be computed without the SVD, the matrix Z^{-1} is still needed in the formation of $\mathcal{M}$ (see §4.2). Hence we would expect the upper bound on the round-off errors in solving equation (2.5-c) for this direction to be at least compatible with that in Lemma 4.3. Consequently, we could expect this variation to be numerically unstable if $\mathcal{J}_S$ is ill-conditioned.

Unlike the AHO method, the potential numerical instability of Algorithms 4.1 and 4.2 in general remains even if the search direction is computed by solving equation (2.2) as a dense linear system of equations (see §5.3).

4.5 Error Analysis for the Least Squares Variation of the TTT Methods

Now we discuss the round-off errors for the least squares variation of the TTT methods discussed in §4.2. In addition to Assumptions 3.1 through 4.1, we further assume that

Assumption 4.3 *Problem (4.11) is solved via an efficient and backward stable method.*

As in §4.1, we will think of the search direction defined by (2.2) as a direction defined at the point $(X^{\dagger}, Z^{\dagger}, y)$ in (4.17), instead of the point (X, Z, y). Hence the search direction satisfies equation (4.18).

Let $\mathcal{R}$ be computed as before and let $\widehat{X}_r$ be the computed version of X_r in equation (4.10). Define $\mathcal{E}^\dagger$, $\mathcal{F}^\dagger$, and $\mathcal{M}^\dagger$ as in (4.22) and let the coefficient matrix and the right hand side vector of the least squares problem (4.11) be computed as $\mathbf{fl}\left(\widetilde{\mathcal{A}}\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\right)$ and $\mathbf{fl}\left(\mathbf{svec}(\widehat{X}_r) + \widetilde{\mathcal{D}}\ (\mathcal{F}\, r_d - r_c)\right)$, respectively. With analysis similar to that in equation (4.23), we write

$$\mathbf{fl}\left(\mathbf{svec}(\widehat{X}_r) + \widetilde{\mathcal{D}}\ (\mathcal{F}\, r_d - r_c)\right) = (\mathcal{I} + \Delta_2)\ \left(\mathbf{svec}(\widehat{X}_r) + (\mathcal{I} + \Delta_1)\ \widetilde{\mathcal{D}}\ \left(\left(\mathcal{F}^\dagger + \delta_1\mathcal{F}^\dagger\right)\ \widehat{r}_d - \widehat{r}_c\right)\right)\ , \quad (4.27)$$

where Δ_1 and Δ_2 are diagonal perturbation matrices and $\delta_1\mathcal{F}^\dagger$ is a perturbation to $\mathcal{F}^\dagger$. Furthermore, with an analysis similar to that in the proof of Lemma 4.3, we can write

$$\mathbf{fl}\left(\widetilde{\mathcal{A}}\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\right) = \widetilde{\mathcal{A}}\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}} + \delta\mathcal{M}^{\frac{1}{2}}\ , \quad \text{where} \quad \delta\mathcal{M}^{\frac{1}{2}} = O\left(\|\mathcal{A}\|_2\ \|\widetilde{R}\|_2\ \|\widetilde{H}^{-1}\|_2\right)\ . \qquad (4.28)$$

By standard error analysis (see Higham [12, Ch. 19]), the computed solution $\widehat{dy}$ is the *exact* solution to a slightly perturbed least squares problem

$$\min_{dy} \left\| \left(\mathbf{fl}\left(\widetilde{\mathcal{A}}\,\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\right) + \Theta\right)^T \cdot dy - \mathbf{fl}\left(\mathbf{svec}(\widehat{X}_r) + \widetilde{\mathcal{D}}\ (\mathcal{F}\ r_d - r_c)\right) \right\|_2\ ,$$

where the m-by-$n(n+1)/2$ matrix $\Theta = O\left(\epsilon \cdot \left\|\mathbf{fl}\left(\widetilde{\mathcal{A}}\,\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\right)\right\|_2\right)$ is a perturbation to $\mathbf{fl}\left(\widetilde{\mathcal{A}}\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\right)$. Stating this result in an equivalent way, $\widehat{dy}$ is the *exact* solution to the normal equation of this perturbed least squares problem:

$$\left(\mathbf{fl}\left(\widetilde{\mathcal{A}}\,\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\right) + \Theta\right)\ \left(\mathbf{fl}\left(\widetilde{\mathcal{A}}\,\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\right) + \Theta\right)^T \widehat{dy} = \left(\mathbf{fl}\left(\widetilde{\mathcal{A}}\,\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\right) + \Theta\right) \cdot \mathbf{fl}\left(\mathbf{svec}(\widehat{X}_r) + \widetilde{\mathcal{D}}\ (\mathcal{F}\ r_d - r_c)\right)\ .$$

In light of (4.28), this equation can be rewritten in the form of Lemma 4.3 as

$$\begin{aligned}
\left(\mathcal{M}^\dagger + \delta\mathcal{M}^\dagger\right)\ \widehat{dy} \quad &= \quad \left(\mathbf{fl}\left(\widetilde{\mathcal{A}}\,\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\right) + \Theta\right) \cdot \mathbf{fl}\left(\mathbf{svec}(\widehat{X}_r) + \widetilde{\mathcal{D}}\ (\mathcal{F}\ r_d - r_c)\right)\ , \\
\text{where} \quad \delta\mathcal{M}^\dagger \quad &\overset{\text{def}}{=} \quad \left(\widetilde{\mathcal{A}}\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}} + \delta\mathcal{M}^{\frac{1}{2}} + \Theta\right)\ \left(\widetilde{\mathcal{A}}\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}} + \delta\mathcal{M}^{\frac{1}{2}} + \Theta\right)^T - \mathcal{M}^\dagger \\
&= \quad \widetilde{\mathcal{A}}\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\ \left(\delta\mathcal{M}^{\frac{1}{2}} + \Theta\right)^T + \left(\delta\mathcal{M}^{\frac{1}{2}} + \Theta\right)\ \left(\widetilde{\mathcal{A}}\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}} + \delta\mathcal{M}^{\frac{1}{2}} + \Theta\right)^T \\
&= \quad O\left(\epsilon \cdot \|\mathcal{A}\|_2^2\ \left\|X^\dagger\right\|_2\ \left\|\left(Z^\dagger\right)^{-1}\right\|_2\right).
\end{aligned}$$

Comparing with Lemma 4.3, the error bounds for $\delta\mathcal{M}^\dagger$ in both cases are identical. Although in the least squares approach $\delta\mathcal{M}^\dagger$ has a special form, it does not seem to make $\|\delta\mathcal{M}^\dagger\|_2$ smaller.

Assume that equations (2.5-b) and (2.5-c) in the least squares approach are solved as in Algorithm 4.1. We get an equation similar to (2.4),

$$\begin{pmatrix} \mathcal{E} + \delta\mathcal{E} & \mathcal{F}^\dagger + \delta_2\mathcal{F}^\dagger & 0 \\ 0 & (\mathcal{I} + \Delta_3)^{-1} & (\mathcal{A} + \delta_2\mathcal{A})^T \\ 0 & 0 & \mathcal{M}^\dagger + \delta\mathcal{M}^\dagger \end{pmatrix} \widehat{d\mathcal{X}} = \begin{pmatrix} \widehat{r}_c \\ \widehat{r}_d \\ \mathbf{fl}\left(\mathbf{svec}(\widehat{X}_r) + \widetilde{\mathcal{D}}\ (\mathcal{F}\ r_d - r_c)\right) \end{pmatrix}\ ,$$

which can be combined with (4.27) to give an equation similar to (2.3):

$$\begin{pmatrix} \mathcal{I} & 0 & 0 \\ 0 & \mathcal{I} & 0 \\ \mathcal{L}_{3,1} & \mathcal{L}_{3,2} & \mathcal{I} \end{pmatrix} \begin{pmatrix} \mathcal{E} + \delta\mathcal{E} & \mathcal{F}^\dagger + \delta_2\mathcal{F}^\dagger & 0 \\ 0 & (\mathcal{I} + \Delta_3)^{-1} & (\mathcal{A} + \delta_2\mathcal{A})^T \\ 0 & 0 & \mathcal{M}^\dagger + \delta\mathcal{M}^\dagger \end{pmatrix} \widehat{d\mathcal{X}} = \begin{pmatrix} \widehat{r}_c \\ \widehat{r}_d \\ \widetilde{r}_p \end{pmatrix}\ ,$$

where $\mathcal{L}_{3,1} = \left(\mathbf{fl}\left(\widetilde{\mathcal{A}}\,\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\right) + \Theta\right) (\mathcal{I} + \Delta_2)\ (\mathcal{I} + \Delta_1)\ \widetilde{\mathcal{D}}\ ,\quad \mathcal{L}_{3,2} = -\mathcal{L}_{3,1}\ \left(\mathcal{F}^\dagger + \delta_1\mathcal{F}^\dagger\right)$

and $\widetilde{r}_p = \left(\mathbf{fl}\left(\widetilde{\mathcal{A}}\,\mathcal{D}_{\mathcal{M}}^{\frac{1}{2}}\right) + \Theta\right)\cdot(\mathcal{I}+\Delta_2)\ \mathbf{svec}(\widehat{X}_r)$. As in §4.1, we can now scale the above equation by $\mathcal{S}$ and bound the round-off errors in both $\mathcal{J}_\mathcal{S}$ and $\mathcal{R}_\mathcal{S}$. Since the upper bounds for $\delta\mathcal{M}^\dagger$ are the same in both cases, the upper bound on the backward errors in the $(3,3)$ block of $\mathcal{J}_\mathcal{S}$ for the least squares problem will be about the same as that of $\mathcal{J}$ for Algorithms 4.1 and 4.2, regardless of how small the backward errors in other blocks of $\mathcal{J}_\mathcal{S}$ might be. Since the upper bound for $\delta\mathcal{M}^\dagger$ is roughly the upper bound in Lemma 4.4, under the assumption that $\kappa\left(\widehat{\Sigma}\right) = O(\sqrt{p})$ and that $\|\left(X^\dagger\right)^{-1}\|_2 = \Omega\left(\|\left(Z^\dagger\right)^{-1}\|_2\right)$, it appears that the least squares approach in general does not seem to be more accurate than Algorithms 4.1 and 4.2.

5 Numerical Experiments

In §5 we first discuss a new choice of search direction in the TTT family. We then discuss how to measure the amount of accuracy in a numerical solution to problem (1.1). And finally we present results from our numerical experiments that support our analysis for the AHO method and the TTT methods.

5.1 A New Search Direction in the TTT Family

Our error analysis of the TTT methods indicates that one factor that potentially limits the amount of accuracy in the numerical solution is the scaled condition number $\kappa(\mathcal{J}_\mathcal{S})$ (see (4.26)). To achieve maximum accuracy in the numerical solution, we would like to find a direction in the TTT family that minimizes $\kappa(\mathcal{J}_\mathcal{S})$.

However, such a direction appears to be very hard to find. Instead, we note that the source of potential bad scaling in equation (4.15) is the ill-conditioning of the matrix P in (2.10-a). This motivates us to choose a direction in the TTT family that minimizes $\kappa(P)$. As in §2.2, write $B = \mathbf{diag}(B_1, \cdots, B_k)$, where the dimension of B_j is the multiplicity of the singular value σ_j of $R\,H^T$. Partition the matrix $\widetilde{H}$ in (2.10-a) accordingly as $\widetilde{H} = \left(\widetilde{H}_1, \cdots, \widetilde{H}_k\right)^T$. The following result of Demmel suggests a particular choice of B that is at most a factor of $\sqrt{k}$ away from optimal.

Lemma 5.1 (Demmel [6]) *Define $\bar{B} = \mathbf{diag}\left(\bar{B}_1, \cdots, \bar{B}_k\right)$, where $\bar{B}_j$ is chosen so that $\bar{B}_j\,\widetilde{H}_j^T$ is row orthonormal, i.e., $\bar{B}_j\,\widetilde{H}_j^T\,\widetilde{H}_j\,\bar{B}_j^T = I_j$, for $j = 1, \cdots, k$. Then*

$$\kappa\left(\bar{B}\,\widetilde{H}\right) \le \sqrt{k}\ \min\left\{\kappa\left(B\,\widetilde{H}\right) \mid \quad where \quad B = \mathbf{diag}(B_1, \cdots, B_k).\right\}.$$

We compared the TTT method with $B = \bar{B}$ with the AHO method and other TTT methods in our numerical experiments. In our implementation, we ignored the possibility of multiple singular values in $R\,H^T$ and instead scaled $\widetilde{H}$ as in (4.14) and chose $\bar{B} = \Phi^{-1}$. This choice of $\bar{B}$ corresponds to the matrix $\bar{B}$ in Lemma 5.1 with $k = n$. See §5.3 for numerical results.

5.2 Measuring Accuracy in a Numerical Solution

Some of the recent numerical studies of interior-point methods on SDPs measured the amount of accuracy in a numerical solution by computing $\|r_p\|_2$, $\|r_d\|_2$ and $\mathbf{tr}(X\,Z)$, the *duality gap*. In the case of linear programming, the current iterate (X, Z, y) is in general close to the optimal solution if these three quantities are sufficiently small. However, this may no longer be true in the case of SDP. Since the matrix $X\,Z$ need not be symmetric, let along positive definite, a small duality gap does not necessarily imply a small $\|X\,Z\|_2$. In this paper, we measure the accuracy in (X, Z, y) by computing the *residual*

$$\widetilde{\mathcal{R}} = \begin{pmatrix} \mathbf{svec}(X\,Z + Z\,X)/2 \\ \mathbf{svec}\left(Z + \mathbf{smat}\left(\mathcal{A}^T\,y\right) - C\right) \\ b - \mathcal{A}\,\mathbf{svec}(X) \end{pmatrix}.$$

To relate $\widetilde{\mathcal{R}}$ to the amount of accuracy in (X, Z, y), we note that $X^* Z^* = 0$ and hence

$$\begin{aligned} X\,Z + Z\,X &= X\,Z + Z\,X - X^*\,Z^* - Z^*\,X^* \\ &= (X - X^*)\;Z + X^*\;(Z - Z^*) + (Z - Z^*)\;X + Z^*\;(X - X^*)\;. \end{aligned}$$

Since the left hand side is symmetric, we symmetrize the right hand side to get

$$X\,Z + Z\,X = (X - X^*)\;\frac{Z + Z^*}{2} + \frac{Z + Z^*}{2}\;(X - X^*) + \frac{X + X^*}{2}\;(Z - Z^*) + (Z - Z^*)\;\frac{X + X^*}{2}\;.$$

This, and the fact that (X^*, Z^*, y^*) is the exact solution to equations (1.2), imply

$$(\mathcal{J} + \mathcal{J}^*) \cdot (\mathcal{X} - \mathcal{X}^*) = 2\,\widetilde{\mathcal{R}}\,,$$

$$\text{where}\quad \mathcal{J} = \begin{pmatrix} Z \otimes_s I & X \otimes_s I & 0 \\ 0 & \mathcal{I} & \mathcal{A}^T \\ \mathcal{A} & 0 & 0 \end{pmatrix} \quad\text{and}\quad \mathcal{J}^* = \begin{pmatrix} Z^* \otimes_s I & X^* \otimes_s I & 0 \\ 0 & \mathcal{I} & \mathcal{A}^T \\ \mathcal{A} & 0 & 0 \end{pmatrix}.$$

Note that $\mathcal{J}$ is the coefficient matrix in equation (3.1). Writing this equation in the form of (3.10), and assuming that (X, Z) is sufficiently close to (X^*, Z^*), we get

$$\frac{\|\mathcal{X} - \mathcal{X}^*\|_2}{\|\mathcal{X}\|_2} \le \kappa\,(\mathcal{J} + \mathcal{J}^*) \cdot \frac{2\,\left\|\widetilde{\mathcal{R}}\right\|_2}{\|\mathcal{J} + \mathcal{J}^*\|_2\;\|\mathcal{X}\|_2} \approx \kappa\,(\mathcal{J} + \mathcal{J}^*) \cdot \frac{\left\|\widetilde{\mathcal{R}}\right\|_2}{\|\mathcal{J}\|_2\;\|\mathcal{X}\|_2}\,.$$

We call the ratio in the last expression the *normalized residual.* This equation suggests that the smaller the normalized residual, the more accurate the numerical solution. Similarly, the smaller $\kappa\,(\mathcal{J} + \mathcal{J}^*)$ is, the more accurate the numerical solution will be. The quantity $\kappa\,(\mathcal{J} + \mathcal{J}^*)$ appears to play the role of the condition number for the SDP. In general, we would expect a stable numerical method to reduce the normalized residual to the order of machine precision, independent of how big $\kappa\,(\mathcal{J} + \mathcal{J}^*)$ might be.

5.3 Numerical Results

We have implemented the AHO method and the TTT methods in `matlab` and have performed a number of numerical experiments. We summarize some of the numerical results below. The computations were done on an Ultra Sparc Station in double precision ($\epsilon \approx 2{\times}10^{-16}$). We tested the following methods:

- The AHO method.
- The NT method by choosing $B = \Sigma^{-\frac{1}{2}}$ in Algorithms 4.1 and 4.2.
- The HKM method with $P^T\,P = Z$, without the SVD, as discussed in §4.2.
- The method discussed in §5.1. We will call it the New method.

In our numerical experiments, we also tested Algorithms 4.1 and 4.2 with $B = I$, which is a variation of the HKM method with $P^T\,P = Z$, without the SVD. Our numerical results indicated that these two variations are compatible in terms of the number of iterations and the amount of accuracy in the numerical solution. The NT method in our experiments is not identical to the NT method in [23, 24, 27]. However, as we argued at the end of §4.4, we expect both variations to suffer from similar numerical instability problems.

For comparison, we also implemented the above four methods by solving the corresponding equation (2.2) with a backward stable dense linear equation solver, with proper re-scaling whenever necessary.

	Nos. of Iterations			
(r, n, m)	AHO	NT	HKM	New
$(3, 10, 9)$	13	19	28	15
$(6, 20, 24)$	12	18	26	16

	Normalized Residuals			
(r, n, m)	AHO	NT	HKM	New
$(3, 10, 9)$	7.2×10^{-16}	1.5×10^{-10}	3.3×10^{-13}	4.8×10^{-14}
$(6, 20, 24)$	5.1×10^{-15}	1.9×10^{-10}	9.9×10^{-12}	2.3×10^{-14}

	(Scaled) Condition Numbers			
(r, n, m)	AHO	NT	HKM	New
$(3, 10, 9)$	1.5×10^{2}	3.5×10^{8}	4.5×10^{5}	1.1×10^{5}
$(6, 20, 24)$	5.7×10^{3}	1.7×10^{9}	1.9×10^{8}	3.6×10^{4}

Table 5.1: Type-I SDPs, With Block LU Factorization

In all cases, we set the initial guess to be $X = Z = I$ and $y = 0$. We chose $\sigma = 0.25$ and $\tau = 0.98$, and switched to the Mehrotra predictor-corrector versions as soon as (cf. [3])

$$\frac{\|r_p\|_2}{\|\mathcal{A}\|_2 \ \|X\|_F} + \frac{\|r_d\|_2}{\|Z\|_F + \|\mathcal{A}\|_2 \ \|y\|_2} \le 10^{-4} .$$

We chose the following two types of test problems:

- Type-I SDPs. For any given m and n, we generate the following quantities randomly:
  - an n-by-n orthogonal matrix Q^*; the m-by-$n(n+1)/2$ matrix $\mathcal{A} = (\mathbf{svec}(A_1), \cdots, \mathbf{svec}(A_m))^T$ and the m-vector y^*.
  - an integer r in $(0, n)$ and positive diagonal matrices Λ_1^* and Λ_2^* with dimensions r-by-r and $(n-r)$-by-$(n-r)$, respectively.

  We then define the SDP by setting

  $$X^* = Q^* \,\mathbf{diag}(\Lambda_1^*, 0)\, (Q^*)^T \ , \ Z^* = Q^* \,\mathbf{diag}(0, \Lambda_2^*)\, (Q^*)^T \ , \ b = \mathcal{A}\,\mathbf{svec}(X^*) \text{ and } C = Z^* + \mathcal{A}^T y^*.$$

  It is straightforward to verify that (X^*, Z^*, y^*) is a solution to (1.2). Type-I SDPs tend to have a relatively well-conditioned unique solution if $r(r+1)/2 \le m \le rn - r(r-1)/2$.

- Type-II SDPs. We generate the symmetric matrices $A_1, \cdots, A_m$ as

  $$A_k = Q^* \begin{pmatrix} U_k & L_k^T \\ L_k & V_k \end{pmatrix} (Q^*)^T ,$$

  where $U_k \in \mathbf{S}^r$ and $V_k \in \mathbf{S}^{n-r}$ are random symmetric matrices, and L_k is an $(n-r)$-by-r matrix such that $\|L_k\|_2 \ll \|A_k\|_2 = \Omega(1)$. The rest of the SDP is generated as in Type-I. With the analysis given in Alizadeh, Haeberly and Overton [3], it can be shown that Type-II SDPs generally have a relatively ill-conditioned unique solution if $r(r+1)/2 \le m \le rn - r(r-1)/2$.

	Nos. of Iterations			
(r,n,m)	AHO	NT	HKM	New
$(3,10,9)$	14	19	23	18
$(6,20,24)$	13	18	20	19

	Normalized Residuals			
(r,n,m)	AHO	NT	HKM	New
$(3,10,9)$	9.3×10^{-17}	1.2×10^{-10}	3.2×10^{-11}	5.0×10^{-16}
$(6,20,24)$	1.7×10^{-16}	1.9×10^{-10}	4.4×10^{-12}	1.2×10^{-16}

	(Scaled) Condition Numbers			
(r,n,m)	AHO	NT	HKM	New
$(3,10,9)$	1.5×10^{2}	1.8×10^{8}	4.8×10^{6}	5.1×10^{2}
$(6,20,24)$	5.7×10^{3}	4.2×10^{9}	5.3×10^{6}	5.7×10^{3}

Table 5.2: Type-I SDPs, Without Block LU Factorization

Our analysis in §4.4 indicates that the amount of accuracy in the numerical solution computed by the TTT methods is related to $\kappa(\mathcal{J}_{\mathcal{S}})$. But our choice of $\mathcal{S}$ suggested in §4.4 may not be optimal. In the numerical experiments we computed the condition number as $\kappa_s(\mathcal{J}) = \kappa(\mathcal{D}\,\mathcal{J})$, where $\mathcal{D}$ is a diagonal matrix chosen so that the rows of $\mathcal{D}\,\mathcal{J}$ all have 2-norm 1. Since $\mathcal{J}$ is an $(m+n(n+1))$-by-$(m+n(n+1))$ matrix, by Lemma 5.1, $\kappa_s(\mathcal{J})$ is at most a factor of $\sqrt{m+n(n+1)}$ away from the optimal. In our numerical experiments, we also computed the normalized residual returned from each of the methods, and the number of iterations it took to achieve it.

Tables 5.1 through 5.4 summarize our results. Table 5.1 shows that for the Type-I SDPs tested, the AHO method was able to reduce the normalized residual to $O(\epsilon)$, and its corresponding $\kappa(\mathcal{J})$ was modest. On the other hand, the NT method could only reduce the normalized residual to about 10^{-10}, and its corresponding $\kappa_s(\mathcal{J})$ was quite large. The HKM and New methods were more accurate than the NT method, but less accurate than the AHO method. Among the three TTT methods, the New method had the smallest $\kappa_s(\mathcal{J})$ and took the least number of iterations.

We also solved the problems in Table 5.1 using these four methods by solving (2.2) as dense linear systems of equations. The results are summarized in Table 5.2. It is interesting to note that the NT method and the HKM method still failed to reduce the normalized residual to full machine precision. Since the normalized residuals were still quite small, it is unlikely that this failure is due to non-convergence of these methods. This suggests that the numerical instability problem with these two methods is *inherent* and there might be no way to overcome this problem for these two methods.

Table 5.3 shows that for the Type-II SDPs tested, *every* one of these methods failed to reduce the normalized residual to (ϵ), and the corresponding $\kappa(\mathcal{J})$ or $\kappa_s(\mathcal{J})$ was very large for all methods. Table 5.3 supports our conclusion that the AHO method and the TTT methods could be numerically unstable if the $\mathcal{J}$ matrices have large (scaled) condition numbers.

As in Table 5.2, we also solved the problems in Table 5.3 using these four methods by solving (2.2) as dense linear systems of equations. We summarize the results in Table 5.4. As in Table 5.2, both the AHO and the New methods were able to reduce the normalized residual to full machine precision, but the NT and the HKM methods still failed to do so.

(r, n, m)	Nos. of Iterations			
	AHO	NT	HKM	New
$(3, 10, 9)$	11	16	16	15
$(6, 20, 24)$	11	14	18	26

(r, n, m)	Normalized Residuals			
	AHO	NT	HKM	New
$(3, 10, 9)$	2.3×10^{-9}	1.5×10^{-8}	3.6×10^{-8}	4.8×10^{-8}
$(6, 20, 24)$	1.6×10^{-10}	1.3×10^{-8}	1.1×10^{-11}	1.8×10^{-8}

(r, n, m)	(Scaled) Condition Numbers			
	AHO	NT	HKM	New
$(3, 10, 9)$	3.6×10^{10}	2.2×10^{12}	7.0×10^{11}	7.8×10^{11}
$(6, 20, 24)$	2.5×10^{11}	8.9×10^{11}	3.2×10^{12}	5.3×10^{11}

Table 5.3: Type-II SDPs, With Block LU Factorization

6 Conclusions and Future Work

In this paper, we analyzed the AHO method in finite precision. We also developed the TTT family of methods and analyzed them in finite precision. Our results indicate that the AHO method and the TTT methods could be numerically stable if a (scaled) condition number associated with the coefficient matrix in (2.2) is well-conditioned, but unstable otherwise. Our analysis also resolves a number of other computational issues related to these methods.

Our analysis raises a number of questions as well. For example, in his finite precision analysis of a number of interior-point methods for linear programming and linear complementarity problems, Wright [29, 30] concluded that these methods are numerically stable under the assumption that the optimal solution is well-conditioned. In light of our results, it would be interesting to investigate how these methods behave if the optimal solution is relatively ill-conditioned.

Our numerical experiments indicate the reason the AHO method appears to be more accurate than the TTT methods is that the condition number for the AHO method is smaller. It is not clear whether this is true in general. A related question is how to choose a direction in the TTT family to achieve best convergence and maximum numerical accuracy.

Finally, the most important question to be answered is whether it is possible to develop interior-point methods that are as efficient and as robust as existing ones but are always numerically stable.

Acknowledgements. The author thanks Professor Michael Overton of the Courant Institute for attracting his attention to the subject of this paper, for many helpful discussions, and for making available to him a `matlab` code of the AHO method and other methods.

	Nos. of Iterations			
(r, n, m)	AHO	NT	HKM	New
$(3, 10, 9)$	19	23	35	24
$(6, 20, 24)$	18	22	21	27

	Normalized Residuals			
(r, n, m)	AHO	NT	HKM	New
$(3, 10, 9)$	6.5×10^{-17}	1.4×10^{-10}	1.6×10^{-12}	3.7×10^{-16}
$(6, 20, 24)$	1.1×10^{-16}	3.3×10^{-11}	9.4×10^{-12}	1.3×10^{-16}

	(Scaled) Condition Numbers			
(r, n, m)	AHO	NT	HKM	New
$(3, 10, 9)$	4.3×10^{11}	4.7×10^{13}	1.9×10^{11}	4.4×10^{11}
$(6, 20, 24)$	3.7×10^{11}	3.5×10^{13}	4.2×10^{11}	3.4×10^{11}

Table 5.4: Type-II SDPs, Without Block LU Factorization

Appendix

In this Appendix we provide proofs to the lemmas in §3 and §4.

A Proofs of Lemmas 3.1 and 4.1

Proof of Lemma 3.1. In equation (3.12), if the matrix A is close to an orthogonal matrix, $A = A^\dagger + O(\epsilon)$, where $A^\dagger$ is exactly orthogonal, then (3.12) can be rewritten as

$$\mathbf{fl}\,(A\,x) = A^\dagger\ ((I + \Delta)\ x) = (I + \bar{\Delta})\ (A^\dagger\, x)\,, \tag{A.1}$$

where $\Delta = \left(A^\dagger\right)^{-1}\ (A + \delta A) - I = O(\epsilon)$ and $\bar{\Delta} = (A + \delta A)\ \left(A^\dagger\right)^{-1} - I = O(\epsilon)\,.$

Since $\|\mathbf{smat}(v)\|_F = \left\|Q^\dagger\ \mathbf{smat}(v)\ \left(Q^\dagger\right)^T\right\|_F = \left\|\left(Q^\dagger\right)^T\ \mathbf{smat}(v)\ Q^\dagger\right\|_F$ for all $v \in \mathbf{S}^n$, it follows that

$$v \rightarrow \mathbf{svec}\left(Q^\dagger\ \mathbf{smat}(v)\ \left(Q^\dagger\right)^T\right) \quad \text{and} \quad v \rightarrow \mathbf{svec}\left(\left(Q^\dagger\right)^T\ \mathbf{smat}(v)\, Q^\dagger\right)$$

are orthogonal linear transformations on $\mathbf{R}^{n(n+1)/2}$. With (3.5), the matrix $\bar{V}$ in (3.3) is computed as $\mathbf{fl}\,(\bar{V}) = \mathbf{fl}\left(\widehat{Q}^T\, V\, \widehat{Q}\right)$. This can be viewed as a matrix-vector product with a nearly orthogonal matrix. Similar to (A.1), there exists an $n(n+1)/2$-by-$n(n+1)/2$ perturbation matrix $\Delta_1 = O(\epsilon)$ such that

$$\mathbf{fl}\,(\bar{V}) = \left(Q^\dagger\right)^T\ \mathbf{smat}((\mathcal{I} + \Delta_1)\ v)\ Q^\dagger\,. \tag{A.2}$$

The matrix $\bar{U}$ in (3.3) is computed from $\mathbf{fl}\,(\bar{V})$ and $\widehat{\Lambda}$ as

$$\mathbf{fl}\,(\bar{U}) = \left(\mathbf{fl}\left(\frac{2\ (\mathbf{fl}\,(\bar{V}))_{i,j}}{\widehat{\lambda}_i + \widehat{\lambda}_j}\right)\right)\,.$$

By our model of arithmetic (1.7), every entry in $\bar{U}$ is computed to full relative accuracy from $\mathbf{fl}\,(\bar{V})$ and $\widehat{\Lambda}$. In other words, $\mathbf{fl}\,(\bar{U})$ satisfies the equation

$$\mathbf{fl}\,(\bar{U})\ \widehat{\Lambda} + \widehat{\Lambda}\,\mathbf{fl}\,(\bar{U}) = 2\,\mathbf{fl}\,(\bar{V}) + \delta\bar{V}\ , \tag{A.3}$$

where the perturbation matrix $\delta\bar{V} \in \mathbf{S}^n$ satisfies

$$\left|(\delta\bar{V})_{i,j}\right| \le O(\epsilon)\cdot\left|(\mathbf{fl}\,(\bar{V}))_{i,j}\right| = O(\epsilon\cdot\|v\|_2)\ .$$

Furthermore, the solution to equation (3.2) is computed from $\mathbf{fl}\,(\bar{U})$ as $\mathbf{fl}\,(U) = \mathbf{fl}\left(\widehat{Q}\,\mathbf{fl}\,(\bar{U})\ \widehat{Q}^T\right)$. Similar to (A.1) and (A.2), $\mathbf{fl}\,(U)$ satisfies

$$\mathbf{fl}\,(U) = \mathbf{smat}\left((\mathcal{I}+\Delta_2)\cdot\mathbf{svec}\left(Q^\dagger\,\mathbf{fl}\,(\bar{U})\ \left(Q^\dagger\right)^T\right)\right)\ , \tag{A.4}$$

where $\Delta_2 = O(\epsilon)$ is an $n(n+1)/2$-by-$n(n+1)/2$ perturbation matrix.

To put all this together, we note that relations (3.6), (A.2) and (A.3) imply

$$\begin{aligned}
&\mathbf{smat}\left(\mathcal{E}^\dagger\,\mathbf{svec}\left(Q^\dagger\,\mathbf{fl}\,(\bar{U})\ \left(Q^\dagger\right)^T\right)\right)\\
&= \frac{\left(Q^\dagger\,\mathbf{fl}\,(\bar{U})\ \left(Q^\dagger\right)^T\right)\ Z^\dagger + Z^\dagger\ \left(Q^\dagger\,\mathbf{fl}\,(\bar{U})\ \left(Q^\dagger\right)^T\right)}{2} = \frac{\left(Q^\dagger\cdot\left(\mathbf{fl}\,(\bar{U})\ \widehat{\Lambda}+\widehat{\Lambda}\,\mathbf{fl}\,(\bar{U})\right)\cdot\left(Q^\dagger\right)^T\right)}{2}\\
&= Q^\dagger\,\mathbf{fl}\,(\bar{V})\ \left(Q^\dagger\right)^T + \frac{Q^\dagger\,\mathbf{fl}\,(\delta\bar{V})\ \left(Q^\dagger\right)^T}{2} = \mathbf{smat}\,((\mathcal{I}+\Delta_1)\ v) + \frac{Q^\dagger\,\mathbf{fl}\,(\delta\bar{V})\ \left(Q^\dagger\right)^T}{2}\\
&= \mathbf{smat}\,((\mathcal{I}+\Delta_3)\ v)\ ,\ \text{ where }\ \Delta_3 \overset{\text{def}}{=} \Delta_1 + \frac{\mathbf{svec}\left(Q^\dagger\,\delta\bar{V}\ \left(Q^\dagger\right)^T\right)\cdot v^T}{2\,\|v\|_2^2}.
\end{aligned}$$

According to (A.2) and (A.3), we have $\Delta_3 = O(\epsilon)$ for all non-zero vector v. Hence

$$\mathbf{svec}\left(Q^\dagger\,\mathbf{fl}\,(\bar{U})\ \left(Q^\dagger\right)^T\right) = \left(\mathcal{E}^\dagger\right)^{-1}\ ((\mathcal{I}+\Delta_3)\ v)\ .$$

Combining this with (A.4) yields the equation in Lemma 3.1. ▮

For the proof of Lemma 4.1, we need to introduce some notation. The standard Kronecker product of any two $n\times n$ matrices G and K is $G\otimes K = (g_{ij}\ K)$. As is shown in the appendix of [27], there exists an $n(n+1)/2$-by-n^2 row orthogonal matrix $\mathcal{Q}$ such that

$$G\otimes_s K = \frac{1}{2}\mathcal{Q}\ (G\otimes K + K\otimes G)\,\mathcal{Q}^T \quad \text{for all } G \text{ and } K \in \mathbf{R}^{n\times n}. \tag{A.5}$$

Let $\mathbf{vec}(G)$ be the n^2-dimensional vector obtained by stacking all the columns of G. Then

$$\mathcal{Q}\ \mathcal{Q}^T = \mathcal{I} \quad \text{and} \quad \mathbf{svec}(H) = \mathcal{Q}\,\mathbf{vec}(H) \quad \text{and} \quad \mathcal{Q}^T\ \mathcal{Q}\,\mathbf{vec}(H) = \mathbf{vec}(H) \quad \text{for all } H\in\mathbf{S}^n. \tag{A.6}$$

Let $\mathcal{P}\in\mathbf{R}^{n^2\times n^2}$ be the permutation matrix such that $\mathcal{P}\,\mathbf{vec}(G) = \mathbf{vec}(G^T)$ for all $G\in\mathbf{R}^{n\times n}$. It is easy to verify that

$$\mathcal{P} = \mathcal{P}^T\ ,\quad \mathcal{P}\ (A\otimes A)\ \mathcal{P} = A\otimes A \quad \text{and} \quad \mathcal{P}\,\mathbf{vec}(H) = \mathbf{vec}(H) \quad \text{for} \quad H\in\mathbf{S}^n\ . \tag{A.7}$$

We also need the following result concerning round-off errors in a dot product (see Higham [12, Ch. 3]),

$$\mathbf{fl}\left(x^T \cdot y\right) = x^T \cdot (y + \delta y)\,, \quad \text{where} \quad |\delta y| \leq O(\epsilon) \cdot |y|\,. \tag{A.8}$$

Proof of Lemma 4.1. Let $A = (a_1, \cdots, a_n)$. According to equation (3.12), the i-th column of $U\,A$ is computed as $(U + \delta_i U)\,a_i$, where $|\delta_i U| \leq O(\epsilon) \cdot |U|$. Hence

$$\mathbf{fl}\,(U\,A) = ((U + \delta_1 U)\,a_1, \cdots, (U + \delta_n U)\,a_n)\;.$$

By Algorithm 4.3 and equation (A.8), both the (i,j) and (j,i) entries of $U\,A\,U^T$ are computed as

$$\left(\mathbf{fl}\left(U\,A\,U^T\right)\right)_{ij} = \sum_{k=1} (\mathbf{fl}(U\,A))_{ik}\;(U_{jk} + \delta_{ij} U_{jk})\;, \quad \text{where} \quad |\delta_{ij} U_{jk}| \leq O(\epsilon) \cdot |U_{jk}|\;.$$

Now we use the above round-off error quantities to define a linear transformation

$$\Theta\;\mathbf{svec}\,(\bar{A}) \overset{\text{def}}{=} \mathbf{svec}\left(\sum_{k=1} ((U + \delta_1 U)\,\bar{a}_1, \cdots, (U + \delta_n U)\,\bar{a}_n)_{ik}\;(U_{jk} + \delta_{ij} U_{jk})\right) - \mathbf{svec}\left(U\;\bar{A}\;U^T\right) \tag{A.9}$$

for any $\bar{A} = (\bar{a}_1, \cdots, \bar{a}_n) \in \mathbf{S}^n$. The $n(n+1)/2$-by-$n(n+1)/2$ matrix Θ is defined by (A.9) and satisfies

$$\mathbf{fl}\,((U \otimes_s U)\;\mathbf{svec}\,(A)) = (U \otimes_s U + \Theta)\;\mathbf{svec}\,(A)\;.$$

To bound Θ, we choose $\bar{A} \geq 0$ and rewrite (A.9) as

$$\Theta\,\mathbf{svec}\,(\bar{A}) = \mathbf{svec}\left(\sum_{k=1} ((U + \delta_1 U)\,\bar{a}_1, \cdots, (U + \delta_n U)\,\bar{a}_n)_{ik}\;\delta_{ij} U_{jk} + \sum_{k=1} (\delta_1 U\,\bar{a}_1, \cdots, \delta_n U\,\bar{a}_n)_{ik}\;U_{jk}\right).$$

Taking absolute value entry-wise, and using the upper bounds on the round-off error quantities, we immediately get

$$|\Theta\;\mathbf{svec}\,(\bar{A})| \leq O(\epsilon) \cdot \mathbf{svec}\left(\sum_{k=1} (|U|\;\bar{a}_1, \cdots, |U|\;\bar{a}_n)_{ik}\;|U_{jk}|\right) = O(\epsilon) \cdot (|U| \otimes_s |U|)\;\mathbf{svec}\,(\bar{A})\;.$$

Since the last relation holds for all $\bar{A} \geq 0$, we conclude that $|\Theta| \leq O(\epsilon) \cdot (|U| \otimes_s |U|)$.

The last step of Algorithm 4.3 is applying a diagonal matrix $\mathcal{D}$ to $\mathbf{fl}\,((U \otimes_s U)\;\mathbf{svec}\,(A))$. By our model of arithmetic (1.7), there exists a diagonal perturbation matrix Δ_1 such that

$$\begin{aligned} \widehat{\mathcal{V}} &= \mathbf{fl}\,(\mathcal{D}\;\mathbf{fl}\,((U \otimes_s U)\;\mathbf{svec}\,(A))) = \mathcal{D}\;(\mathcal{I} + \Delta_1)\;\mathbf{fl}\,((U \otimes_s U)\;\mathbf{svec}\,(A)) \\ &= \mathcal{D}\;((U \otimes_s U + \Theta_1)\;\mathbf{svec}\,(A))\;, \end{aligned}$$

where

$$\Theta_1 \overset{\text{def}}{=} (\mathcal{I} + \Delta_1)\;(U \otimes_s U + \Theta) - (U \otimes_s U) \quad \text{satisfies} \quad |\Theta_1| \leq O(\epsilon) \cdot (|U| \otimes_s |U|)\;.$$

To prove the remaining part of Lemma 4.1, define and partition

$$\widehat{V} = \mathbf{fl}\left(U^{-1}\;A\right) = (\widehat{v}_1, \cdots, \widehat{v}_n) = \begin{pmatrix} \widetilde{v}_1^T \\ \vdots \\ \widetilde{v}_n^T \end{pmatrix} \quad \text{and} \quad \widehat{W} = \mathbf{fl}\left(\mathbf{fl}\left(U^{-1}\;A\right)\;U^{-T}\right) = \begin{pmatrix} \widetilde{w}_1^T \\ \vdots \\ \widetilde{w}_n^T \end{pmatrix}.$$

It follows from Algorithm 4.4 that there exist round-off error matrices $\delta_i U$ with $\|\delta_i U\|_2 = O\,(\epsilon \cdot \|U\|_2)$ such that $(U + \delta_i U)\,\widehat{v}_i = a_i$, and $\bar{\delta}_i U$ with $\|\bar{\delta}_i U\|_2 = O\,(\epsilon \cdot \|U\|_2)$ such that $\widetilde{w}_i^T\,(U + \bar{\delta}_i U) = \widetilde{v}_i^T$ for all i. Putting all these relations together and simplifying, we get

$$(U \otimes U + \Delta_2)\;\mathbf{vec}\left(\widehat{W}\right) = \mathbf{vec}(A)\,, \quad \text{where} \quad \|\Delta_2\|_2 = O\left(\epsilon \cdot \|U\|_2^2\right) \in \mathbf{R}^{n^2 \times n^2}\;. \tag{A.10}$$

To convert this equation into the form in Lemma 4.1, we apply $(\mathcal{I}+\mathcal{P})/2$ to it and simplify to get

$$(U \otimes U)\ \frac{(\mathcal{I}+\mathcal{P})\ \mathbf{vec}\left(\widehat{W}\right)}{2} = \mathbf{vec}(A) - \frac{\mathcal{I}+\mathcal{P}}{2}\ \Delta_2\ \mathbf{vec}\left(\widehat{W}\right)\ , \tag{A.11}$$

where we have used equation (A.7) and the fact that $A \in \mathbf{S}^n$. By definition and relation (A.6),

$$\frac{(\mathcal{I}+\mathcal{P})\ \mathbf{vec}\left(\widehat{W}\right)}{2} = \mathbf{vec}\left(\frac{\widehat{W}+\widehat{W}^T}{2}\right) = \mathcal{Q}^T\ \mathbf{svec}\left(\frac{\widehat{W}+\widehat{W}^T}{2}\right)\ .$$

Apply $\mathcal{Q}$ to (A.11), and simply the resulting equation with this relation and (A.5) and (A.6) to get,

$$(U \otimes_s U)\ \mathbf{svec}\left(\frac{\widehat{W}+\widehat{W}^T}{2}\right) = \mathbf{svec}(A) - \mathcal{Q}\ \frac{\mathcal{I}+\mathcal{P}}{2}\ \Delta_2\ \mathbf{vec}\left(\widehat{W}\right)\ ,$$

which can be further rewritten as

$$((U \otimes_s U) + \Delta_3)\ \mathbf{svec}\left(\frac{\widehat{W}+\widehat{W}^T}{2}\right) = \mathbf{svec}(A), \quad \Delta_3 \stackrel{\text{def}}{=} \frac{\mathcal{Q}\,\frac{\mathcal{I}+\mathcal{P}}{2}\,\Delta_2\,\mathbf{vec}\left(\widehat{W}\right)\cdot\mathbf{svec}\left(\frac{\widehat{W}+\widehat{W}^T}{2}\right)^T}{\left\|\mathbf{svec}\left(\frac{\widehat{W}+\widehat{W}^T}{2}\right)\right\|_2^2}.$$

To derive an upper bound on Δ_3, we define $W^* = U^{-1}\,A\,U^{-T} \in \mathbf{S}^n$. By assumption in Lemma 4.1, $\kappa(U) \ll 1/\sqrt{\epsilon}$. It follows that $\kappa(U \otimes_s U) \ll 1/\epsilon$. Since the backward error in (A.10) is of the order $O\left(\epsilon \cdot \|U\|_2^2\right)$, it follows from standard perturbation theory (cf. (3.10) and see Demmel [7, Ch. 2]) that

$$\frac{\left\|\mathbf{vec}\left(\widehat{W}\right) - \mathbf{vec}\left(W^*\right)\right\|_2}{\left\|\mathbf{vec}\left(\widehat{W}\right)\right\|_2} \ll O(1)\ .$$

Consequently,

$$\begin{aligned}\left\|\mathbf{svec}\left(\frac{\widehat{W}+\widehat{W}^T}{2}\right)\right\|_2 &= \left\|\mathbf{svec}\left(\widehat{W} + \frac{\left(\widehat{W}-W^*\right)^T - \left(\widehat{W}-W^*\right)}{2}\right)\right\|_2 \\ &\geq \left\|\mathbf{vec}\left(\widehat{W}\right)\right\|_2 - \left\|\mathbf{vec}\left(\frac{\left(\widehat{W}-W^*\right)^T - \left(\widehat{W}-W^*\right)}{2}\right)\right\|_2 = \Omega\left(\left\|\mathbf{vec}\left(\widehat{W}\right)\right\|_2\right)\ .\end{aligned}$$

Plugging this into the definition of Δ_3, we have

$$\|\Delta_3\|_2 = O\left(\|\Delta_2\|_2\right) = O\left(\epsilon \cdot \|U\|_2^2\right)\ .$$

To complete the proof, we note that $\widehat{\mathcal{W}}$ in Lemma 4.1 is obtained by symmetrizing $\widehat{W}$ in finite precision. By our model of arithmetic (1.7), there exists a diagonal perturbation matrix $\Delta_4 = O(\epsilon)$ such that

$$\widehat{\mathcal{W}} = (I + \Delta_4)\ \mathbf{svec}\left(\frac{\widehat{W}+\widehat{W}^T}{2}\right)\ .$$

Hence $\widehat{\mathcal{W}}$ satisfies the equation in Lemma 4.1 with

$$\Theta_2 \stackrel{\text{def}}{=} ((U \otimes_s U) + \Delta_3)\,(I + \Delta_4)^{-1} - U \otimes_s U = O\left(\epsilon \cdot \|U\|_2^2\right)\ . \ \blacksquare$$

B Proofs of Lemmas in §4.4

Proof of Lemma 4.2. According to Algorithm 4.1 and (4.17),

$$\begin{aligned}
\mu &= \sigma\,\frac{X^{\dagger}\bullet Z^{\dagger}}{n} = \sigma\,\frac{\left(\widetilde{R}\,\widetilde{H}^T\right)\bullet\left(\widetilde{R}\,\widetilde{H}^T\right)}{n} = \sigma\,\frac{\left(\widehat{\Sigma}+E\right)^T\,\left(\widehat{\Sigma}+E\right)}{n}\,. \\
\mathbf{fl}(\mu) &= \mathbf{fl}\left(\sigma\,\frac{\mathbf{tr}\left(\widehat{\Sigma}^2\right)}{n}\right) = \mu + \sigma\,\frac{\mathbf{tr}\left(\widehat{\Sigma}^2-\left(\widehat{\Sigma}+E\right)^T\,\left(\widehat{\Sigma}+E\right)\right)}{n} + O\left(\epsilon\cdot\|\widehat{\Sigma}\|_2^2\right) \\
&= \mu + O\left(\epsilon\cdot\|\widehat{\Sigma}\|_2\,\|\widetilde{R}\|_2\,\|\widetilde{H}\|_2\right)\ ,
\end{aligned}$$

where we have used Assumption 4.1. With $\mathbf{fl}(\mu)$, r_c in (4.3) can be computed as

$$\mathbf{smat}\,(\widehat{r}_c) = \mathbf{fl}\left(\mathbf{fl}(\mu)\,I-\widehat{\Sigma}^2\right) = \mu\,I-\widehat{\Sigma}^2+E_c\,,$$

where both $\widehat{r}_c$ and E_c are diagonal matrices, with $|E_c| \le O\left(\epsilon\cdot\|\widehat{\Sigma}\|_2\,\|\widetilde{R}\|_2\,\|\widetilde{H}\|_2\right)\,I$. It now follows from (4.17) and (4.18) that

$$\begin{aligned}
\mathbf{smat}\,(r_c) &= \mu\,I-\mathbf{H}_B\left(\widetilde{H}\,\widetilde{R}^T\,\widetilde{R}\,\widetilde{H}^T\right) = \mu\,I-\mathbf{H}_B\left(\left(\widehat{\Sigma}+E\right)^T\left(\widehat{\Sigma}+E\right)\right) \\
&= B\left(\mu\,I-\widehat{\Sigma}^2\right)\,B^{-1}-\mathbf{H}_B\left(E^T\,\widehat{\Sigma}+\widehat{\Sigma}\,E+E^T\,E\right) \\
&= \mathbf{smat}\,(\widehat{r}_c)-E_c-\mathbf{H}_B\left(E^T\,\widehat{\Sigma}+\widehat{\Sigma}\,E+E^T\,E\right) \\
&= \mathbf{smat}\,(\widehat{r}_c)-\mathbf{H}_B\left(\widetilde{E}_c\right)\ ,\quad\text{where}\quad \widetilde{E}_c \stackrel{\text{def}}{=} E_c+E^T\,\widehat{\Sigma}+\widehat{\Sigma}\,E+E^T\,E\,.
\end{aligned}$$

It follows from Assumption 4.1 and the upper bound on E_c that $\widetilde{E}_c = O\left(\epsilon\cdot\|\widehat{\Sigma}\|_2\,\|\widetilde{R}\|_2\,\|\widetilde{H}\|_2\right)$. Thus,

$$\begin{aligned}
\left\|\mathcal{S}^{-1}\ (\widehat{r}_c-r_c)\right\|_2 &= \left\|\mathbf{smat}\left(\mathcal{S}^{-1}\ \mathbf{svec}\left(\mathbf{H}_B\left(\widetilde{E}_c\right)\right)\right)\right\|_F \\
&= \left\|\left(\frac{2\ \left(\mathbf{H}_B\left(\widetilde{E}_c\right)\right)_{ij}}{\left(\dfrac{\beta_i}{\beta_j}+\dfrac{\beta_j}{\beta_i}\right)\ \phi_i\,\phi_j+(\phi+\psi)^2\ \left(\dfrac{\beta_i\,\widehat{\sigma}_i}{\beta_j\,\widehat{\sigma}_j}+\dfrac{\beta_j\,\widehat{\sigma}_j}{\beta_i\,\widehat{\sigma}_i}\right)}\right)\right\|_F \\
&\le \left\|\left(\frac{\dfrac{\beta_i}{\beta_j}+\dfrac{\beta_j}{\beta_i}}{\dfrac{\beta_i\,\widehat{\sigma}_i}{\beta_j\,\widehat{\sigma}_j}+\dfrac{\beta_j\,\widehat{\sigma}_j}{\beta_i\,\widehat{\sigma}_i}}\cdot\frac{O\left(\epsilon\cdot\|\widehat{\Sigma}\|_2\,\|\widetilde{R}\|_2\,\|\widetilde{H}\|_2\right)}{(\phi+\psi)^2}\right)\right\|_F \\
&= \left\|\left(\frac{\widehat{\sigma}_i\,\widehat{\sigma}_j\left(\beta_i^2+\beta_j^2\right)}{\beta_i^2\,\widehat{\sigma}_i^2+\beta_j^2\,\widehat{\sigma}_j^2}\cdot\frac{O\left(\epsilon\cdot\|\widehat{\Sigma}\|_2\,\|\widetilde{R}\|_2\,\|\widetilde{H}\|_2\right)}{(\phi+\psi)^2}\right)\right\|_F\ . \qquad \text{(B.1)}
\end{aligned}$$

The HKM search direction [11, 13, 18] $P^T\,P = Z$ and the NT direction [23, 24] correspond to $B = I$ and $B = \widehat{\Sigma}^{-\frac{1}{2}}$, respectively. Since $(\phi+\psi)^2 = O(\|\widehat{\Sigma}\|_2)$, the expression on the right hand side of (B.1) is bounded by $O\left(\epsilon\cdot\|\widetilde{R}\|_2\,\|\widetilde{H}\|_2\right)$ for these two choices of B. In general, we use (4.7) to bound the right hand side of (B.1) by

$$\left\|\left(\frac{\left(\widehat{\sigma}_i^2+\widehat{\sigma}_j^2\right)}{\widehat{\sigma}_i\,\widehat{\sigma}_j}\cdot\frac{O\left(\epsilon\cdot\|\widehat{\Sigma}\|_2\,\|\widetilde{R}\|_2\,\|\widetilde{H}\|_2\right)}{(\phi+\psi)^2}\right)\right\|_F = O\left(\epsilon\cdot\kappa\left(\widehat{\Sigma}\right)\ \|\widetilde{R}\|_2\,\|\widetilde{H}\|_2\right)\ . \quad \blacksquare$$

Proof of Lemma 4.3. We first consider the round-off errors in computing $\widetilde{\mathcal{A}}$ in (4.6). It follows from Lemma 4.1 that

$$\left|\mathbf{fl}\left(\widetilde{\mathcal{A}}\right) - \widetilde{\mathcal{A}}\right| \le O(\epsilon)\cdot\left(\mathbf{svec}\left(\left|\widetilde{R}\right|\ |A_1|\ \left|\widetilde{R}\right|^T\right), \cdots, \mathbf{svec}\left(\left|\widetilde{R}\right|\ |A_m|\ \left|\widetilde{R}\right|^T\right)\right)^T = O(\epsilon)\cdot|\mathcal{A}|\cdot\left(\left|\widetilde{R}\right| \otimes_s \left|\widetilde{R}\right|\right)^T .$$

Hence the round-off errors in the computed Schur complement $\mathcal{M}$ in (4.6) are bounded by

$$\begin{aligned}
& O(\epsilon)\cdot|\mathcal{A}|\cdot\left(\left|\widetilde{R}\right| \otimes_s \left|\widetilde{R}\right|\right)^T \cdot \widehat{\mathcal{D}}_{\mathcal{M}} \cdot \left(\left|\widetilde{R}\right| \otimes_s \left|\widetilde{R}\right|\right)\cdot|\mathcal{A}|^T \\
= \ & O(\epsilon)\cdot\left(\left(\left|\widetilde{R}\right|\ |A_i|\ \left|\widetilde{R}\right|^T\right)\cdot\left(\left(\frac{\beta_i^2\ \widehat{\sigma}_i^2 + \beta_j^2\ \widehat{\sigma}_j^2}{\widehat{\sigma}_i^2\ \widehat{\sigma}_j^2\ \left(\beta_i^2 + \beta_j^2\right)}\right) \odot \left(\left|\widetilde{R}\right|\ |A_j|\ \left|\widetilde{R}\right|^T\right)\right)\right) \\
\le \ & O(\epsilon)\cdot\left(\left(\left|\widetilde{R}\right|\ |A_i|\ \left|\widetilde{R}\right|^T\right)\cdot\left(\left(\frac{1}{\widehat{\sigma}_j^2} + \frac{1}{\widehat{\sigma}_i^2}\right) \odot \left(\left|\widetilde{R}\right|\ |A_j|\ \left|\widetilde{R}\right|^T\right)\right)\right) \\
= \ & O(\epsilon)\cdot\left(\left(\left|\widetilde{R}\right|\ |A_i|\ \left|\widetilde{R}\right|^T\ \widehat{\Sigma}^{-1}\right) \bullet \left(\left|\widetilde{R}\right|\ |A_j|\ \left|\widetilde{R}\right|^T\ \widehat{\Sigma}^{-1}\right)\right) \\
& + O(\epsilon)\cdot\left(\left(\widehat{\Sigma}^{-1}\ \left|\widetilde{R}\right|\ |A_i|\ \left|\widetilde{R}\right|^T\right) \bullet \left(\widehat{\Sigma}^{-1}\ \left|\widetilde{R}\right|\ |A_j|\ \left|\widetilde{R}\right|^T\right)\right) ,
\end{aligned}$$

where we have used (4.7). By Assumption 4.1, we have

$$\widehat{\Sigma}^{-1}\ \left|\widetilde{R}\right| = \left|\left(I + \widehat{\Sigma}^{-1}\ E\right)\widetilde{H}^{-T}\right| = \Omega\left(\left\|\widetilde{H}^{-1}\right\|_2\right) .$$

With this estimate, the round-off errors in the computed Schur complement can now bounded by

$$O\left(\epsilon\cdot\|\mathcal{A}\|_2^2\ \left\|\widetilde{R}\right\|_2^2\ \left\|\widehat{\Sigma}^{-1}\ \widetilde{R}\right\|_2^2\right) = O\left(\epsilon\cdot\|\mathcal{A}\|_2^2\ \left\|\widetilde{R}\right\|_2^2\ \left\|\widetilde{H}^{-1}\right\|_2^2\right) = O\left(\epsilon\cdot\|\mathcal{A}\|_2^2\ \left\|X^\dagger\right\|_2\ \left\|\left(Z^\dagger\right)^{-1}\right\|_2\right) .$$

In other words, the computed $\mathcal{M}$ can be written as $\mathcal{M}^\dagger + O\left(\epsilon\cdot\|\mathcal{A}\|_2^2\ \left\|X^\dagger\right\|_2\ \left\|\left(Z^\dagger\right)^{-1}\right\|_2\right)$. In addition, it is clear from this analysis that

$$\left\|\mathcal{M}^\dagger\right\|_2 \le \|\mathcal{A}\|_2^2\ \left\|X^\dagger\right\|_2\ \left\|\left(Z^\dagger\right)^{-1}\right\|_2 .$$

By Assumption 3.4, the backward errors committed by the backward solver to solve (2.5-c) after $\mathcal{M}$ is computed are bounded by $O\left(\epsilon\cdot\|\mathcal{M}^\dagger\|_2\right)$. Putting all these errors together, we arrive at Lemma 4.3. ▌

Proof of Lemma 4.4. It is obvious that terms in the second row of $\delta\mathcal{J}_S$ in equation (4.25) are bounded by $(\epsilon\cdot(1+\|\mathcal{A}\|_2))$. In the following we will derive upper bounds on the terms in the second and third rows that do not depend on B.

We first consider the third row of $\delta\mathcal{J}_S$. With arguments similar to those in the proof of Lemma 4.3, we can bound all the terms in the $(3,3)$ and $(3,2)$ blocks of $\delta\mathcal{J}_S$ by $O\left(\epsilon\cdot\|\mathcal{A}\|_2^2\ \left\|X^\dagger\right\|_2\ \left\|\left(Z^\dagger\right)^{-1}\right\|_2\right)$ and $O\left(\epsilon\cdot\|\mathcal{A}\|_2\ \left\|X^\dagger\right\|_2\ \left\|\left(Z^\dagger\right)^{-1}\right\|_2\right)$, respectively. To bound the round-off errors in the $(3,1)$ block, rewrite

$$\begin{aligned}
\mathcal{L}_{3,1}\,\mathcal{E} - \mathcal{A} + \mathcal{L}_{3,1}\,\delta\mathcal{E} \ &= \ (\mathcal{A} + \delta_1\mathcal{A})\left(\left(\mathcal{E}^\dagger\right)^{-1} + \Theta_2\right)\mathcal{E} - \mathcal{A} + \mathcal{L}_{3,1}\,\delta\mathcal{E} \\
&= \ \mathcal{A}\left(\left(\mathcal{E}^\dagger\right)^{-1}\mathcal{E} - \mathcal{I}\right) + \mathcal{A}\,\Theta_2\,\mathcal{E} + \delta_1\mathcal{A}\left(\left(\mathcal{E}^\dagger\right)^{-1} + \Theta_2\right)\mathcal{E} + \mathcal{L}_{3,1}\,\delta\mathcal{E}. \qquad \text{(B.2)}
\end{aligned}$$

By definitions (4.17), (4.18) and (4.22), we have

$$\begin{aligned}
\left(\mathcal{E}^\dagger\right)^{-1} \mathcal{E} &= \left(\left(B\,\widehat{\Sigma}\,\widetilde{R}^{-T}\right) \otimes_s \left(B^{-1}\,\widehat{\Sigma}\,\widetilde{R}^{-T}\right)\right)^{-1} \left(\left(B^{-1}\,\widetilde{H}\right) \otimes_s \left(B\,\widetilde{H}\right)\right) \\
&= \left(\left(B \otimes_s B^{-1}\right) \cdot \left(\widehat{\Sigma} \otimes_s \widehat{\Sigma}\right) \cdot \left(\widetilde{R}^{-T} \otimes_s \widetilde{R}^{-T}\right)\right)^{-1} \left(\left(B^{-1} \otimes_s B\right) \cdot \left(\widetilde{H} \otimes_s \widetilde{H}\right)\right) \\
&= \left(\widetilde{R}^T \otimes_s \widetilde{R}^T\right) \cdot \left(\widehat{\Sigma}^{-1} \otimes_s \widehat{\Sigma}^{-1}\right) \cdot \left(\widetilde{H} \otimes_s \widetilde{H}\right) = \left(\widehat{\Sigma}^{-1}\,\widetilde{R}^T\,\widetilde{H}\right) \otimes_s \left(\widehat{\Sigma}^{-1}\,\widetilde{R}^T\,\widetilde{H}\right) \\
&= \left(\widehat{\Sigma}^{-1}\,\left(\widehat{\Sigma} + E\right)^T\right) \otimes_s \left(\widehat{\Sigma}^{-1}\,\left(\widehat{\Sigma} + E\right)^T\right) \\
&= \mathcal{I} + \left(\widehat{\Sigma}^{-1}\,E^T\right) \otimes_s \left(\widehat{\Sigma}^{-1}\,\left(\widehat{\Sigma} + E\right)^T\right) + I \otimes_s \left(\widehat{\Sigma}^{-1}\,E^T\right) .
\end{aligned}$$

By Assumption 4.1 and with the last expression, we bound the first term in (B.2) as

$$\begin{aligned}
\left\|\mathcal{A}\,\left(\left(\mathcal{E}^\dagger\right)^{-1}\,\mathcal{E} - \mathcal{I}\right)\right\|_2 &\le O\left(\|\mathcal{A}\|_2\,\left\|\widehat{\Sigma}^{-1}\right\|_2\,\|E\|_2\right) = O\left(\epsilon \cdot \|\mathcal{A}\|_2\,\|\widetilde{R}\|_2\,\|\widetilde{H}\|_2\,\left\|\widehat{\Sigma}^{-1}\right\|_2\right) \\
&\le O\left(\epsilon \cdot \|\mathcal{A}\|_2\,\left(\|X^\dagger\|_2\,\|\left(X^\dagger\right)^{-1}\|_2\,\|Z^\dagger\|_2\,\|\left(Z^\dagger\right)^{-1}\|_2\right)^{\frac{1}{2}}\right) .
\end{aligned}$$

By definition (4.18) and Lemma 4.1, all the other terms in (B.2) are bounded by

$$\begin{aligned}
&O\left(\epsilon\right) \cdot |\mathcal{A}| \cdot \left(\left|\widetilde{R}\right|^T \otimes_s \left|\widetilde{R}\right|^T\right)\,\left(B\,\widehat{\Sigma} \otimes_s B^{-1}\,\widehat{\Sigma}\right)^{-1}\,\left(\left(B^{-1} \otimes_s B\right)|\Theta|\right) \\
&= O\left(\epsilon\right) \cdot |\mathcal{A}| \cdot \left(\left|\widehat{\Sigma}^{-1}\,\widetilde{R}\right|^T \otimes_s \left|\widehat{\Sigma}^{-1}\,\widetilde{R}\right|^T\right)|\Theta| \quad \text{for some} \quad \Theta = \Omega\left(\|\widetilde{H}\|_2^2\right) .
\end{aligned}$$

Similar to the proof of Lemma 4.3, this bound can be simplified to $O\left(\epsilon \cdot \|\mathcal{A}\|_2\,\|Z^\dagger\|_2\,\|\left(Z^\dagger\right)^{-1}\|_2\right)$. Adding it all up, the terms in the third row of $\delta\mathcal{J}_S$ are bounded by

$$\begin{aligned}
&O\left(\epsilon \cdot (1 + \|\mathcal{A}\|_2)^2\,\left(\|X^\dagger\|_2 + \|Z^\dagger\|_2\right)\,\left(\|\left(X^\dagger\right)^{-1}\|_2 + \|\left(Z^\dagger\right)^{-1}\|_2\right)\right) \\
&= O\left(\epsilon \cdot \left(\|X^\dagger\|_2 + \|Z^\dagger\|_2\right)\,\left(\|\left(X^\dagger\right)^{-1}\|_2 + \|\left(Z^\dagger\right)^{-1}\|_2\right)\right) . \qquad \text{(B.3)}
\end{aligned}$$

Now we consider the terms in the first row of $\delta\mathcal{J}_S$. A bound on $\delta\mathcal{E}$ is given in (4.24). Since

$$\mathcal{S} = \mathcal{S}_\mathcal{E} + \mathcal{S}_\mathcal{F} \ge \mathcal{S}_\mathcal{E} = \left(B^{-1}\,\Phi \otimes_s B\,\Phi\right) ,$$

it follows from (4.24) that the term in the $(1,1)$ block of $\delta\mathcal{J}_S$ satisfies

$$\left\|\mathcal{S}^{-1}\,\delta\mathcal{E}\right\|_2 = \left\|\mathcal{S}^{-1}\,\mathcal{S}_S\,\Theta_4\right\|_2 \le \|\Theta_4\|_2 = O(\epsilon) .$$

For the $(1,2)$ block, we use definitions (4.17), (4.18) and (4.22) to get

$$\begin{aligned}
\mathcal{F}^\dagger - \mathcal{F} &= \left(B\,\widehat{\Sigma}\,\widetilde{R}\right) \otimes_s \left(B^{-1}\,\widehat{\Sigma}^{-1}\,\widetilde{R}\right) - \left(B\,\widetilde{H}\,X^\dagger\right) \otimes_s \left(B^{-1}\,\widetilde{H}^{-T}\right) \\
&= \left(B\,\widehat{\Sigma}\,\widetilde{R}\right) \otimes_s \left(B^{-1}\,\widehat{\Sigma}^{-1}\,\left(\widehat{\Sigma} + E\right)\,\widetilde{H}^{-T}\right) - \left(B\,\left(\widehat{\Sigma} + E\right)^T\,\widetilde{R}\right) \otimes_s \left(B^{-1}\,\widetilde{H}^{-T}\right) \\
&= \left(B\,\widehat{\Sigma}\,\widetilde{R}\right) \otimes_s \left(B^{-1}\,\widehat{\Sigma}^{-1}\,E\,\widetilde{H}^{-T}\right) - \left(B\,E^T\,\widetilde{R}\right) \otimes_s \left(B^{-1}\,\widetilde{H}^{-T}\right) . \qquad \text{(B.4)}
\end{aligned}$$

Since $\mathcal{S} \geq \mathcal{S}_{\mathcal{F}} = (\phi+\psi)^2 \cdot \left(B\,\widehat{\Sigma}\right) \otimes_s \left(B\,\widehat{\Sigma}\right)^{-1}$, we can scale and bound the first term in (B.4) as

$$\begin{aligned} \left\|\mathcal{S}^{-1}\left(B\,\widehat{\Sigma}\,\widetilde{R}\right) \otimes_s \left(B^{-1}\,\widehat{\Sigma}^{-1}\,E\;\,\widetilde{H}^{-T}\right)\right\|_2 &\leq \frac{\|\widetilde{R}\|_F\,\|E\|_F\,\|\widetilde{H}^{-T}\|_F}{(\phi+\psi)^2} \\ &\leq\; O\left(\epsilon\cdot\left(\|X^{\dagger}\|_2+\|Z^{\dagger}\|_2\right)^{\frac{1}{2}}\,\left(\|\left(X^{\dagger}\right)^{-1}\|_2+\|\left(Z^{\dagger}\right)^{-i}\|_2\right)^{\frac{1}{2}}\right)\;, \end{aligned}$$

where we have used (4.17) and the fact that $\phi = \Omega(\|\widetilde{H}\|_2)$ and $\psi = \Omega(\|\widetilde{R}\|_2)$. We also chose to write the bound in a form similar to (B.3). For the second term in (B.4), we have

$$\left\|\mathcal{S}^{-1}\left(B\,E^T\,\widetilde{R}\right) \otimes_s \left(B^{-1}\,\widetilde{H}^{-T}\right)\right\|_2 \leq \left\|\mathcal{S}_{\mathcal{F}}^{-1}\left(B \otimes_s B^{-1}\right)\right\|_2 \cdot \|E\|_F\,\|\widetilde{R}\|_F\,\|\widetilde{H}^{-T}\|_F\,. \qquad \text{(B.5)}$$

To see the diagonal entries of $\mathcal{S}_{\mathcal{F}}^{-1}\left(B \otimes_s B^{-1}\right)$ more clearly, we apply it to the vector $\mathbf{e}$ in §1.4:

$$\mathcal{S}_{\mathcal{F}}^{-1}\left(B \otimes_s B^{-1}\right)\;\mathbf{e} = \frac{1}{(\phi+\psi)^2}\mathbf{svec}\left(\frac{\dfrac{\beta_i}{\beta_j}+\dfrac{\beta_j}{\beta_i}}{\dfrac{\beta_i\,\widehat{\sigma}_i}{\beta_j\,\widehat{\sigma}_j}+\dfrac{\beta_j\,\widehat{\sigma}_j}{\beta_i\,\widehat{\sigma}_i}}\right)\,.$$

Similar to (B.1), the entries in the last matrix is bounded by $1/(\phi+\psi)^2$ for the HKM search direction $P^T\,P = Z$ and the NT direction, and bounded by $\kappa\left(\widehat{\Sigma}\right)/(\phi+\psi)^2$ in general. Combine this with (4.17) and (B.5), we obtain a bound on the second term in (B.4) similar to that on the first term.

$$\left\|\mathcal{S}^{-1}\left(B\,E^T\,\widetilde{R}\right) \otimes_s \left(B^{-1}\,\widetilde{H}^{-T}\right)\right\|_2 \leq O\left(\epsilon\cdot\kappa\left(\widehat{\Sigma}\right)\,\left(\|X^{\dagger}\|_2+\|Z^{\dagger}\|_2\right)^{\frac{1}{2}}\,\left(\|\left(X^{\dagger}\right)^{-1}\|_2+\|\left(Z^{\dagger}\right)^{-1}\|_2\right)^{\frac{1}{2}}\right).$$

The last term of $\delta\mathcal{J}_{\mathcal{S}}$ to be bounded is $\mathcal{S}^{-1}\delta_2\mathcal{F}^{\dagger}$. It follows from Lemma 4.1 that

$$\left|\delta_2\mathcal{F}^{\dagger}\right| \leq O(\epsilon)\cdot\left(B\,\widehat{\Sigma}\,\left|\widetilde{R}\right|\right) \otimes_s \left(B^{-1}\,\widehat{\Sigma}^{-1}\,\left|\widetilde{R}\right|\right) = O(\epsilon)\cdot\mathcal{S}_{\mathcal{F}}\,\left(\left|\widetilde{R}\right| \otimes_s \left|\widetilde{R}\right|\right)\,.$$

Hence an analysis similar to above yields $\left\|\mathcal{S}^{-1}\delta_2\mathcal{F}^{\dagger}\right\|_2 = O(\epsilon)$. Adding up bounds for all three rows of $\delta\mathcal{J}_{\mathcal{S}}$, we arrive at the equation in Lemma 4.4. ▮

References

[1] F. Alizadeh. *Combinatorial Optimization with Interior Point Methods and Semidefinite Matrices.* PhD thesis, University of Minnesota, 1991.

[2] F. Alizadeh. Interior-point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Opt.*, 5:13–51, 1995.

[3] F. Alizadeh, J.-P. Haeberly, and M. Overton. Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results. Computer Science Dept. Technical Report 721, New York University, 1996.

[4] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* SIAM, Philadelphia, 1994.

[5] S. Chandrasekaran and I. Ipsen. Backward errors for eigenvalue and singular value decompositions. *Numer. Math.*, 68:215–223, 1994.

[6] J. Demmel. The condition number of equivalence transformations that block diagonalize matrix pencils. *SIAM J. Num. Anal.*, 20(3):599–610, June 1983.

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

[8] A. Forsgren, P. Gill, and J. Shinnerl. Stability of symmetric ill-conditioned systems arising in interior methods for constrained optimization. Report TRITA-MAT-1994-24, Royal Institute of Technology, 1994.

[9] K. Fujisawa, M. Kojima, and K. Nakata. SDPA User's Manual. ftp.is.titech.ac.jp/pub/OpRes/articles/b308.ps.Z, 1997.

[10] G. Golub and C. van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, MD, 3nd edition, 1996.

[11] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz. An interior point method for semidefinite programming. *SIAM J. Opt.*, 6:342–361, 1996.

[12] N. J. Higham. *Accuracy and Stability of Numerical Algorithms.* SIAM, 1996.

[13] M. Kojima, S. Shindoh, and S. Hara. Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices. Research Reports on Information Sciences No. B-282, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, April 1994. Revised April 1995.

[14] A. S. Lewis and M. L. Overton. Eigenvalue optimization. In A. Iserles, editor, *Acta Numerica, Volume 5*, pages 149–190. Cambridge University Press, 1996.

[15] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM J. Opt.*, 2:575–601, 1992.

[16] S. Mizuno, M. J. Todd, and Y. Ye. On adaptive step primal-dual interior-point algorithms for linear programming. *Math. of Operations Research*, 18:964–981, 1993.

[17] R. D. C. Monteiro. Polynomial convergence of primal-dual algorithms for semidefinite programming based on Monteiro and Zhang family of directions. Unpublished manuscript, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 1996.

[18] R. D. C. Monteiro. Primal-dual path following algorithms for semidefinite programming. Unpublished manuscript, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 1995.

[19] R. D. C. Monteiro and T. Tsuchiya. Polynomial convergence of a new family of primal-dual algorithms for semidefinite programming. Unpublished manuscript, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 1996.

[20] R. D. C. Monteiro and T. Tsuchiya. Polynomiality of primal-dual algorithms for semidefinite linear complementarity problems based on the Kojima-Shindoh-Hara family of directions. Unpublished manuscript, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 1996.

[21] R. D. C. Monteiro and Y. Zhang. A unified analysis for a class of path-following primal-dual interior-point algorithms for semidefinite programming. Unpublished manuscript, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 1996.

[22] Y. Nesterov and A. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming.* SIAM, Philadelphia, 1994.

[23] Y. E. Nesterov and M. J. Todd. Self-scaled barriers and interior-point methods in convex programming. Technical Report 1091, School of Operations Research and Industrial Engineering, Cornell University, 1994. To appear in Mathematics of Operations Research.

[24] Y. E. Nesterov and M. J. Todd. Primal-dual interior-point methods for self-scaled cones. Technical Report 1125, School of Operations Research nad Industrial Engineering, Cornell University, 1995.

[25] D. B. Ponceleón. *Barrier Methods for Large-Scale Quadratic Programming.* PhD thesis, Stanford University, 1990.

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

[27] M. J. Todd, K. C. Toh, and R. H. Tütüncü. On the Nesterov-Todd direction in semidefinite programming. Technical Report 1154, School of Operations Research nad Industrial Engineering, Cornell University, 1996.

[28] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[29] S. J. Wright. Stability of augmented system factorizations in interior-point methods. Preprint MCS-P446-0694, Argonne National Lab., June 1994. Revised July 1995.

[30] S. J. Wright. Stability of linear equations solvers in interior-point methods. *SIAM J. Matrix Anal. Appl.*, 16:1287–1307, 1994.

[31] S. J. Wright. *Primal-Dual Interior-Point Methods.* SIAM, Philadelphia, 1997.

[32] Y. Zhang. On extending primal-dual interior-point algorithms from linear programming to semidefinite programming. Technical Report TR95-20, Department of Mathematics and Statistics, University of Maryland Baltimore County, 1995.