Mathematical Programming manuscript No.
(will be inserted by the editor)

D. Goldfarb · K. Scheinberg

Product-Form Cholesky Factorization in Interior Point Methods for Second-Order Cone Programming*

Received: date / Revised version: date

Abstract. Second-order cone programming (SOCP) problems are typically solved by interior point methods. As in linear programming (LP), interior point methods can, in theory, solve SOCPs in polynomial time and can, in practice, exploit sparsity in the problem data. Specifically, when cones of large dimension are present, the density that results in the normal equations that are solved at each iteration can be remedied in a manner similar to the treatment of dense columns in an LP. Here we propose a product-form Cholesky factorization (PFCF) approach, and show that it is more numerically stable than the alternative Sherman-Morrison-Woodbury approach. We derive several PFCF variants and compare their theoretical perfomance. Finally, we prove that the elements of L in the Cholesky factorizations LDL^T that arise in interior point methods for SOCP are uniformly bounded as the duality gap tends to zero as long as the iterates remain is some conic neighborhood of the cental path.

Key words. Product-form Cholesky factorization, Second-order cone programming, interior point methods

1. Introduction

Second-order cone programming (SOCP) problems are optimization problems in which a linear objective is minimized over the intersection of an affine linear manifold with the Cartesian product of second-order (Lorentz) cones (some of which may be cones in one dimension, - i.e., nonnegativity constraints) [20]. This class of problems includes linear programming (LP), convex quadratic programming (QP) and convex quadratically constrained quadratic programming (QCQP) as special cases, and is itself a subclass of semidefinite programming (SDP). SOCP problems are of interest for several reasons: interior point methods with polynomial-time iteration complexity have been developed to solve such problems; the theoretical and algorithmic issues for SOCPs (and SDPs) are more complicated than those for LP problems, yet SOCPs are easier to solve in both

D. Goldfarb: Columbia University, Dept. of IEOR, New York, NY e-mail: goldfarb@columbia.edu

K. Scheinberg: IBM T.J. Watson Research Center, Yorktown Heights, NY e-mail: katyas@watson.ibm.com

Mathematics Subject Classification (1991): 90C25, 90C51, 15A23

* Research supported in part by NSF Grants CDA 97-26385, DMS 01-04282, ONR Grant N000140310514 and DOE Grant GE-FG01-92ER-25126

theory and practice than SDPs; there are many interesting and important applications that can be modeled as SOCP's. For the specifics of applying interior point methods to SOCP problems see [2], [19], [28]. For examples of applications that can be solved as SOCP's and a general survey of SOCP see [1], [16]. For an extension of the simplex method to SOCP see [10].

In this paper we are concerned with the efficient implementation of interior point methods for SOCP ([3]). In particular, we are interested in approaches that exploit sparsity in the problem data. If a problem is large-scale, the matrix of the linear system that defines the affine space is usually sparse. On every iteration of an interior point method a linear system of equations has to be solved. This system is typically first reduced to a smaller system called the *normal equations*. In the LP case the sparsity of the normal equations matrix depends only on the sparsity of the original data. However, in the SOCP case, if the size of at least one of the SOCP cones is relatively large, then the matrix of the normal equations is very dense; namely it has a completely dense block of the same dimension as that of the SOCP cone. On the other hand, if all SOCP cones are relatively small, then the situation is much closer to the LP case.

Although the matrix of the normal equations is very dense if any of the SOCP cones is relatively large, this matrix has a special structure that can be exploited: it can be represented as a sum of a sparse matrix and a *low-rank* dense matrix. A natural approach is to compute a factorization of the sparse part and then update this factorization using the Sherman-Morrison-Woodbury updating formula. Unfortunately, this approach is not numerically stable on most SOCP problems (even randomly generated, nondegenerate problems) [2].

We propose instead using various "product-form" updates of the LDL^T factorization of a symmetric positive semidefinite matrix to handle these low-rank updates. We will take the liberty of referring to LDL^T factorizations as *Cholesky* factorizations, in spite of the fact that historically the latter referred to factorizations of the form $\hat{L}\hat{L}^T$. We will show that the LDL^T factorization is "stable", in the sense that $||L||$ remains bounded independent of μ (even as $\mu \rightarrow 0$), where μ is the barrier parameter or duality gap. Our product-form updates are based on non-product-form methods that were introduced in the 60's and 70's [5], [7], [8], [9]. One of these product-form updates was recently proposed in [11] for use in interior point methods for LP problems that have a small number of dense columns. Moreover, in [11] this update is proved to be stable in the above sense for LP problems.

Using the product-form Cholesky factorization (PFCF) for SOCP problems is both more natural and more complicated than in the LP case. In LP, one needs to use PFCF only when there are dense columns in the original data. Even then the Sherman-Morrison-Woodbury approach or a variant of it often works if the problem is not degenerate; i.e. when the normal equations matrix is well-conditioned. In the SOCP case the low-rank dense part is always present if there is a relatively large SOCP cone. Moreover, the normal equations matrix is very often badly scaled in the limit, so the Sherman-Morrison-Woodbury approach is *almost always* likely to run into numerical difficulties.

Also the analysis for the SOCP case is significantly harder. First, the low rank updates of the factorization involve both additions and subtractions, while in the LP case only additions occur. Second, the eigenvalue/eigenvector structure of the normal equations matrices for SOCPs is more complicated than it is for LPs. For example, assuming that the initial data is well scaled and a primal-dual method is used, the normal equations matrix in LP can have eigenvalues of only two types in the limit: large eigenvalues, approaching ∞ at a rate of approximately $1/\mu$, and small eigenvalues approaching zero at a rate of approximately μ. For SOCP problems the normal equations matrix has an additional type of eigenvalue, which neither converges to ∞ nor to 0, but remains $O(1)$ in value.

The paper is organized as follows. In Section 2 we introduce a primal-dual pair of SOCP problems. We consider the pure primal, pure dual and the Nesterov-Todd (NT) primal-dual search directions and the scaling matrices and normal equations that arise in interior point methods for SOCP that use these search directions. It is shown how the special structure of the scaling matrix, and, hence, the normal equations matrix, can be used to exploit sparsity. In Section 3 we develop several product-form Cholesky factorization approaches and show how and when they should be applied. Section 4 contains our analysis of the stability of the Cholesky and product-form Cholesky factorizations in SOCP. In Section 5 we present an example that shows that the results proved in Section 4 may not hold if the solution path deviates substantially from the central path. Finally in Section 6 we present numerical examples illustrating the stability results and the savings in CPU time achieved by using product-form factorizations in SOCP.

2. Second-order cone programming

We consider the following pair of primal-dual second-order cone programs

$$(P) \qquad \begin{array}{rl} \min & c_1^T x_1 + c_2^T x_2 + \ldots + c_N^T x_N \\ \text{s.t.} & A_1 x_1 + A_2 x_2 + \ldots + A_N x_N = b, \\ & x_i \geq_{K_i} 0, \end{array}$$

and

$$(D) \qquad \begin{array}{rl} \max & b^T y \\ \text{s.t.} & A_i^T y + z_i = c_i, \quad i = 1, \ldots, N \\ & z_i \geq_{K_i} 0, \end{array}$$

where $A_i \in \mathbf{R}^{m \times n_i}$, and $c_i \in \mathbf{R}^{n_i}$, $x_i \in \mathbf{R}^{n_i}$, $z_i \in \mathbf{R}^{n_i}$, $i = 1, \ldots, N$, $b \in \mathbf{R}^m$ and $y \in \mathbf{R}^m$ are all column vectors. The conic "nonnegativity" constraint $x_i \geq_{K_i} 0$ denotes $x_i^0 \geq ||\bar{x}_i||$, where x_i^0 is the first component of the vector x_i and $\bar{x}_i$ is the vector composed of the remaining $n_i - 1$ components of x_i. We will use the notation $\gamma(x_i) = \sqrt{(x_i^0)^2 - ||\bar{x}_i||^2}$, and the notation $A = [A_1, A_2, \ldots, A_N]$, $x = (x_1^T, x_2^T, \ldots, x_N^T)^T$ and $z = (z_1^T, z_2^T, \ldots, z_N^T)^T$. Some of the quadratic cones may be one-dimensional, thus reducing the conic constraint $x_i \geq_{K_i} 0$ to the simple nonnegativity constraint $x_i^0 \geq 0$. Therefore, the above problems include linear variables/constraints as a special case.

It can be shown (e.g., see [2]) that the complementarity conditions for the primal-dual solution pair (x, z) corresponds to the following system of $n_1 + n_2 + \ldots + n_N$ equations

$$\begin{aligned} x_i^0 z_i^0 + \bar{x}_i^T \bar{z}_i &= 0 \quad i = 1, \ldots, N \\ z_i^0 \bar{x}_i + x_i^0 \bar{z}_i &= 0, \quad i = 1, \ldots, N. \end{aligned} \tag{1}$$

If we define an "arrow-shaped" matrix $\mathrm{Arr}(x_i)$ and the block diagonal matrix $\mathrm{Arr}(x)$ respectively as

$$\mathrm{Arr}(x_i) = \begin{bmatrix} x_i^0 & x_i^1 & \ldots & x_i^{n_i-1} \\ x_i^1 & x_i^0 & & \\ \vdots & & \ddots & \\ x_i^{n_i-1} & & & x_i^0 \end{bmatrix} \quad \text{and} \quad \mathrm{Arr}(x) = \begin{bmatrix} \mathrm{Arr}(x_1) & & & \\ & \mathrm{Arr}(x_2) & & \\ & & \ddots & \\ & & & \mathrm{Arr}(x_N) \end{bmatrix},$$

then the complementarity conditions (1) can be succinctly expressed as

$$\mathrm{Arr}(x)z = \mathrm{Arr}(z)x = \mathrm{Arr}(x)\mathrm{Arr}(z)e_0 = 0,$$

where

$$e^{0^T} = (e_1^{0^T}, e_2^{0^T}, \ldots, e_N^{0^T}) \equiv (\underbrace{1, 0, \ldots, 0}_{n_1}, \underbrace{1, 0, \ldots, 0}_{n_2}, \ldots, \underbrace{1, 0, \ldots, 0}_{n_N})^T.$$

It follows from the complementarity conditions for the i-th primal-dual pair of cones that

$$(z_i^0)^2 \bar{x}_i^T \bar{x}_i + (x_i^0)^2 \bar{z}_i^T \bar{z}_i + 2 x_i^0 z_i^0 \bar{x}_i^T \bar{z}_i = (z_i^0)^2 \bar{x}_i^T \bar{x}_i + (x_i^0)^2 \bar{z}_i^T \bar{z}_i - 2(x_i^0 z_i^0)^2 = 0.$$

Hence,

$$(z_i^0)^2((x_i^0)^2 - ||\bar{x}_i||^2) + (x_i^0)^2((z_i^0)^2 - ||\bar{z}_i||^2) = 0,$$

which together with the fact that $x_i \geq_{K_i} 0$ and $z_i \geq_{K_i} 0$ implies that for the i-th pair of optimal primal and dual solutions either

$$\text{(i)} \quad z_i^0 = 0 \quad \text{or} \quad \text{(ii)} \quad x_i^0 = 0 \quad \text{or} \quad \text{(iii)} \quad z_i^0 = ||\bar{z}_i|| \quad \text{and} \quad x_i^0 = ||\bar{x}_i||.$$

Strict complementarity is said to hold for a pair of optimal primal-dual solutions $(x_1, x_2, \ldots, x_N)$ and $(z_1, z_2, \ldots, z_N)$, if one and only one of the above three conditions holds for each optimal primal-dual pair $(x_i, z_i), i = 1, \ldots, N,$. The three possible cases are: type 1) $x_i >_{K_i} 0$ and $z_i = 0$, (the primal solution is in the interior of the cone and dual solution is at the origin); type 2) $x_i = 0$ and $z_i >_{K_i} 0$ (the primal solution is at the origin and the dual solution is in the interior of the cone); and type 3) $x_i^0 > 0$, $x_i^0 = ||\bar{x}_i||$ and $z_i^0 > 0$, $z_i^0 = ||\bar{z}_i||$ (both primal and dual solutions are on the boundary of the cone, but not at the origin). We will say that a cone is of type 1 (2 or 3) if the corresponding type of complementarity holds at the optimal solution for this cone. If $n_i = 1$ then only the first two cases are possible, and the complementarity and strict complementarity conditions reduce to those for LP.

Whereas, in LP a strictly complementary optimal solution pair always exists, in SOCP this is not always the case. However, we shall, henceforth, assume that a strictly complementary optimal solution pair exists for the problems that we shall consider. Note that in this case an interior point method converges to such a solution.

In this paper we are interested in the computational behavior of interior point methods that are path-following or central region following. The following system of perturbed optimality conditions for values of the parameter $\mu > 0$ defines the so-called central path for the primal-dual pair of SOCPs (P) and (D) (e.g., see [28]):

$$\begin{aligned} \mathrm{Arr}(z)\mathrm{Arr}(x)e^0 &= \mu e^0, \\ Ax &= b, \\ A^T y + z &= c. \end{aligned} \tag{2}$$

At each iteration of an interior point method a step is taken towards the solution of this system for a fixed value of μ. The direction of this step $(\Delta x, \Delta y, \Delta z)$ is determined by applying Newton's method to a system of equations equivalent to (2) obtained by either algebraically manipulating (2) before applying Newton's method or scaling the variables in (P) and (D). This yields a system of the form

$$\begin{aligned} F\Delta z + \Delta x &= r_c, \\ A\Delta x &= r_p, \\ A^T \Delta y + \Delta z &= r_d, \end{aligned} \tag{3}$$

that must be solved for $(\Delta x, \Delta y, \Delta z)$. For purely primal methods and purely dual methods, F is a positive definite symmetric matrix. This is also the case for primal-dual methods that use the so-called *commutative class* of the Monteiro-Zhang family of directions ([18], [25], [31]). The right-hand side vector $r \equiv (r_c, r_p, r_d)$ also depends on the interior point method that is used and both F and r depend on the current iterate (x, y, z).

Eliminating Δz from (3) one obtains the *augmented system*

$$\begin{bmatrix} -F^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_d - F^{-1} r_c \\ r_p \end{pmatrix}. \tag{4}$$

If F is symmetric, (4) can be solved by computing the Bunch-Parlett-Kaufman factorization (e.g., see [12]) of the symmetric indefinite matrix that appears on the left-hand side of (4). However, when F is also positive definite most interior point methods solve (4) by eliminating Δx to reduce it to the so-called *normal equations*

$$AFA^T \Delta y = r_p - Ar_c + AFr_d,$$

and then solve this system by Cholesky factorization.

2.1. The Scaling Matrix F

Let us now study the properties of the scaling matrix F. This matrix is block diagonal, i.e.,

$$F = \begin{bmatrix} F_1 & & & \\ & F_2 & & \\ & & \ddots & \\ & & & F_N \end{bmatrix}. \tag{5}$$

To simplify the discussion, we will consider henceforth only purely primal, purely dual and the Nesterov-Todd direction [21], [22](a special case of the commutative class of primal-dual directions). In these three cases, the submatrices F_i, for $i = 1, \ldots, N$, have the form

$$F_i = (2u_i u_i^T + \gamma^2(u_i)R)/2 = \frac{1}{2}\begin{bmatrix} (u_i^0)^2 + ||\bar{u}_i||^2 & 2u_i^0 \bar{u}_i^T \\ 2u_i^0 \bar{u}_i & \gamma^2(u_i)I + 2\bar{u}_i \bar{u}_i^T \end{bmatrix}, \tag{6}$$

where I is the $(n_i - 1) \times (n_i - 1)$ identity matrix and $R = \begin{bmatrix} -1 & \\ & I \end{bmatrix}$. In the purely primal case $u_i = x_i$, in the purely dual case $u_i^0 = z_i^0/\gamma^2(z_i)$ and $\bar{u}_i = -\bar{z}_i/\gamma^2(z_i)$, and in the Nesterov-Todd case

$$u_i^0 = (\gamma(z_i)x_i^0 + \gamma(x_i)z_i^0)/\gamma(x_i)\sqrt{x_i^T z_i + \gamma(x_i)\gamma(z_i)},$$

and

$$\bar{u}_i = (-\gamma(z_i)\bar{x}_i + \gamma(x_i)\bar{z}_i)/\gamma(x_i)\sqrt{x_i^T z_i + \gamma(x_i)\gamma(z_i)}.$$

It is easy to show that in the latter case $\gamma(u_i) = \sqrt{2\gamma(x_i)/\gamma(z_i)}$.

Consider the normal equations matrix

$$AFA^T = \sum_{i=1}^{N} A_i F_i A_i^T.$$

In general, each F_i is completely dense since none of the components of u_i are likely to be zero. Thus, if the size of F_i, for some i, is of the same order of magnitude as the number of rows in A_i (i.e., m), then $A_i F_i A_i^T$ will be dense even if A_i is sparse. On the other hand, if F_i has only a few rows and columns, then $A_i F_i A_i^T$ will be relatively sparse if A_i is sparse and has many rows.

We are interested in exploiting the sparsity of the original data when the dimension of at least one of the SOCP cones is relatively large. Since in this case AFA^T is dense, computing its factorization directly may be very costly. We will make use of the special structure of F: for every i, F_i can be represented as a sum of a scaled identity matrix and two rank-one terms, one with a positive coefficient and the other with a negative coefficient:

$$F_i = \frac{\gamma^2(u_i)}{2} I + u_i u_i^T - \gamma^2(u_i) e_i^0 (e_i^0)^T,$$

where I is the $n_i \times n_i$ identity matrix and e_i^0 is its first column. The matrix AFA^T can, therefore, be represented as

$$\sum_{i=1}^{N} \frac{\gamma^2(u_i)}{2} A_i A_i^T + \sum_{i=1}^{N} (v_i v_i^T - w_i w_i^T),$$

where $v_i = A_i u_i$, and $w_i = \gamma(u_i) A_i^0$ and A_i^0 denotes the first column of the submatrix A_i.

Let $\mathcal{I} \subseteq \{1, \ldots, N\}$ be the set of the indices of second-order cones whose dimensions are large enough that it is worthwhile splitting $A_i F_i A_i^T$ for each $i \in \mathcal{I}$ into its sparse and dense parts. Specifically, we can write AFA^T as $AFA^T = \sum_{i=1}^N A_i \bar{F}_i A_i^T + \sum_{i \in \mathcal{I}} (v_i v_i^T - w_i w_i^T)$, where $\bar{F}_i = F_i$ as defined by (6) for $i \in \{1, \ldots, N\} \backslash \mathcal{I}$, and $\bar{F}_i = \frac{\gamma^2(u_i)}{2} I$ for $i \in \mathcal{I}$. To take advantage of the sparsity in the matrix A, one can first factorize the sparse matrix $\sum_{i=1}^N A_i \bar{F}_i A_i^T$ and then update this factorization to account for the rank-$2|\mathcal{I}|$ dense term $\sum_{i \in \mathcal{I}} (v_i v_i^T - w_i w_i^T)$. Henceforth, we shall assume that $|\mathcal{I}| \ll m$ so that the dense part of AFA^T has low rank. As already mentioned, one way to do this is to use the Sherman-Morrison-Woodbury updating formula. This approach is often used in implementations of interior point methods for linear programming (e.g., see [6]), but it can be numerically unstable [11]. There are modifications of this method that can be applied to linear programming problems with dense columns that were proposed in [4] and in [24], but these modifications still suffer from numerical instability in the SOCP case. In the next section we propose several stable and efficient ways to handle low-rank updates of a Cholesky factorization.

3. Product-form factorizations

In [11], a product-form LDL^T factorization was proposed for exploiting sparsity in linear programming problems with small numbers of dense columns when solving such problems by interior point methods. In this section we describe how this factorization and other product-form factorizations can be used to exploit sparsity when applying interior point methods to SOCP problems.

All of these approaches begin by obtaining the Cholesky factorization LDL^T of the sparse symmetric positive semidefinite matrix $A\bar{F}A^T \equiv \sum_{i=1}^N A_i \bar{F}_i A_i^T$ defined in the previous section, where $D \in \mathbf{R}^{m \times m}$ is a nonnegative diagonal matrix with diagonal elements d_j, $j = 1, \ldots, m$ and $L \in \mathbf{R}^{m \times m}$ is a lower triangular matrix with ones on the diagonal. The LDL^T factorization of a matrix is related to the *standard* Cholesky factorization $\hat{L}\hat{L}^T$ of that matrix ($\hat{L} = LD^{1/2}$), but avoids the computation of square roots required by the latter. Computing L and D is often referred to as *symmetric Gaussian elimination.* We allow $d_j = 0$ since $A\bar{F}A^T$ may not be positive definite. In such a case the j-th column of L is e_j, the j-th column of the identity matrix. (See Higham [13] for discussion of the Cholesky factorization of positive semidefinite matrices.)

Let us consider how to update the factorization LDL^T when a symmetric matrix of the form VSV^T is added to it, where $V \in \mathbf{R}^{m\times k}$ and has full column rank and $S \equiv diag\{s_1, \ldots, s_k\} \in \mathbf{R}^{k\times k}$ is a nonsingular diagonal matrix. In general, some of the diagonal elements s_j of S may be positive and some may be negative; however, we assume that $LDL^T + VSV^T$ is positive semidefinite. In particular, we are interested in the case where VSV^T equals $\sum_{i\in\mathcal{I}}(v_i v_i^T - w_i w_i^T)$ or a partial sum of the rank-one outer products that make up this sum. Since L is nonsingular, the updated LDL^T factorization is

$$\bar{L}\bar{D}\bar{L}^T = LDL^T + VSV^T = L(D + PSP^T)L^T = L\tilde{L}\tilde{D}\tilde{L}^T L^T, \tag{7}$$

where $P \in \mathbf{R}^{m\times k}$ is the solution of

$$LP = V,$$

and $\tilde{L}\tilde{D}\tilde{L}^T$ is the Cholesky factorization of $D + PSP^T$, i.e.,

$$D + PSP^T = \tilde{L}\tilde{D}\tilde{L}^T. \tag{8}$$

Rather than computing $\bar{L} = L\tilde{L}$, we store both L and $\tilde{L}$ (or more accurately, a representation of $\tilde{L}$ as we shall show below).

Let the transpose of $p^{(j)} \in \mathbf{R}^k$ be the $j-th$ row of P and $P^{(j)} \in \mathbf{R}^{(m-j)\times k}$ be the matrix consisting of the last $m-j$ rows of P. Hence $P^{(0)} = P$ and $P^{(j-1)^T} = [p^{(j)}, P^{(j)^T}]$. Also, let $D^{(j)} \in \mathbf{R}^{(m-j)\times(m-j)}$ be the diagonal matrix with diagonal elements $d_{j+1}, \ldots, d_m$ (hence $D^{(0)} = D$) and let $\Sigma_0 = S$. It is easy to verify that after one step of symmetric Gaussian elimination we have the factorization

$$\begin{aligned} D^{(0)} + P^{(0)}\Sigma_0 P^{(0)^T} &= \begin{bmatrix} d_1 + p^{(1)^T}\Sigma_0 p^{(1)} & p^{(1)^T}\Sigma_0 P^{(1)^T} \\ P^{(1)}\Sigma_0 p^{(1)} & D^{(1)} + P^{(1)}\Sigma_0 P^{(1)^T} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ P^{(1)}q^{(1)} & I \end{bmatrix} \begin{bmatrix} \tilde{d}_1 & 0 \\ 0 & D^{(1)} + P^{(1)}\Sigma_1 P^{(1)^T} \end{bmatrix} \begin{bmatrix} 1 & q^{(1)^T}P^{(1)^T} \\ 0 & I \end{bmatrix}, \end{aligned}$$

where

$$\begin{aligned} \tilde{d}_1 &= d_1 + p^{(1)^T}\Sigma_0 p^{(1)}, \\ q^{(1)} &= \begin{cases} (1/\tilde{d}_1)\Sigma_0 p^{(1)}, & \text{if } \tilde{d}_1 \neq 0 \\ 0, & \text{if } \tilde{d}_1 = 0, \end{cases} \\ \Sigma_1 &= \Sigma_0 - \tilde{d}_1 q^{(1)} q^{(1)^T}. \end{aligned}$$

Since the $(m-1)\times(m-1)$ matrix $D^{(1)} + P^{(1)}\Sigma_1 P^{(1)^T}$ is identical in form to the $m\times m$ matrix $D^{(0)} + P^{(0)}\Sigma_0 P^{(0)^T}$, it follows that by repeating the above procedure on the matrix $D^{(j)} + P^{(j)}\Sigma_j P^{(j)^T}$, for $j = 1, 2, \ldots, m-2$, we obtain the factorization (8), where

$$\tilde{L} \equiv \tilde{L}(P^{(1)}, Q) = \begin{bmatrix} 1 & & & & & \\ p^{(2)^T} q^{(1)} & 1 & & & & \\ p^{(3)^T} q^{(1)} & p^{(3)^T} q^{(2)} & 1 & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ p^{(m-1)^T} q^{(1)} & p^{(m-1)^T} q^{(2)} & p^{(m-1)^T} q^{(3)} & \cdots & 1 & \\ p^{(m)^T} q^{(1)} & p^{(m)^T} q^{(2)} & p^{(m)^T} q^{(3)} & \cdots & p^{(m)^T} q^{(m-1)} & 1 \end{bmatrix}, \tag{9}$$

and $Q^T \equiv [q^{(1)}, \ldots, q^{(m-1)}]$, $q^{(j)} \in \mathbf{R}^k$, for $j = 1, \ldots, m-1$ and $\tilde{D} = \mathrm{diag}\{\tilde{d}_1, \ldots, \tilde{d}_m\}$ can be computed from the following recurrence relations:

$$\begin{aligned} &\Sigma_0 := S, \\ &r^{(1)} := \Sigma_0 p^{(1)}, \\ &\tilde{d}_1 := d_1 + r^{(1)^T} p^{(1)}, \\ &\text{for } j = 1, 2, \ldots, m-1 \\ &\quad q^{(j)} := \begin{cases} (1/\tilde{d}_j) r^{(j)}, & \text{if } \tilde{d}_j \neq 0 \\ 0, & \text{if } \tilde{d}_j = 0, \end{cases} \\ &\quad \Sigma_j := \Sigma_{j-1} - q^{(j)} r^{(j)^T} \\ &\quad r^{(j+1)} := \Sigma_j p^{(j+1)}, \\ &\quad \tilde{d}_{j+1} := d_{j+1} + r^{(j+1)^T} p^{(j+1)}. \end{aligned} \tag{10}$$

The above updating algorithm, which we shall refer to as the *rank-k product-form* method, is a specialization of a method proposed by Bennett [5] for computing the LDU factorization of $A + XCY^T$ (U is a unit upper triangular matrix), given the LDU factorization of A, to the symmetric rank-k update (8). Because of its special form, the matrix (9) requires storage of only the two $(m-1) \times k$ matrices $P^{(1)}$ and Q. Moreover, solving $\tilde{L}y = z$ or $\tilde{L}^T y = z$ for y requires $2k(m-1)$ multiplications.

Instead of performing the rank-k update in (7) in one step, one can accomplish the same result as a sequence of k rank-one updates. Specifically, if $V \equiv [V_1, V_2, \ldots, V_k]$ and we let $\tilde{D}_0 \equiv D$ and $\tilde{L}_0 \equiv L$, then

$$\bar{L}\bar{D}\bar{L}^T = LDL^T + \sum_{j=1}^{k} s_j V_j V_j^T = \tilde{L}_0 \tilde{L}_1 \tilde{L}_2 \ldots \tilde{L}_k \tilde{D}_k \tilde{L}_k^T \ldots \tilde{L}_2^T \tilde{L}_1^T \tilde{L}_0^T, \tag{11}$$

where for $j = 1, \ldots, k$, $P_j \in \mathbf{R}^m$ is the solution of $\tilde{L}_0 \tilde{L}_1 \tilde{L}_2 \ldots \tilde{L}_{j-1} P_j = V_j$, and $\tilde{L}_j \tilde{D}_j \tilde{L}_j^T$ is the Cholesky factorization of $\tilde{D}_{j-1} + s_j P_j P_j^T$, i.e.,

$$\tilde{D}_{j-1} + s_j P_j P_j^T = \tilde{L}_j \tilde{D}_j \tilde{L}_j^T. \tag{12}$$

For $j = 1, \ldots, k$, we compute $\tilde{L}_j \equiv \tilde{L}(P_j, Q_j)$ and $\tilde{D}_j$ using the above recurrence formulas (10). Let us consider each rank-one update (12) in more detail. We will use p to denote the vector P_j, p_j to denote its j-th component and s to denote s_j. Hence the vectors $p^{(j)}$, $q^{(j)}$ and $r^{(j)}$ and matrices Σ_j and S become scalars, which we will denote as p_j, q_j, r_j, σ_j and s, respectively, and the matrices $P^{(1)}$ and Q that are needed to represent $\tilde{L} \equiv \tilde{L}_j$ become vectors in $\mathbf{R}^{(m-1)}$. Thus,

$$\tilde{L} = \tilde{L}(p,q) = \begin{bmatrix} 1 & & & & & \\ p_2q_1 & 1 & & & & \\ p_3q_1 & p_3q_2 & 1 & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ p_{m-1}q_1 & p_{m-1}q_2 & p_{m-2}q_3 & \cdots & 1 & \\ p_mq_1 & p_mq_2 & p_mq_3 & \cdots & p_mq_{m-1} & 1 \end{bmatrix}. \tag{13}$$

We will also denote $\tilde{D}_j$ and $\tilde{D}_{j-1}$ by $\tilde{D}$ and D, respectively. Recurrence relations (10) in this case become:

$$\begin{aligned} &\sigma_0 := s, \\ &r_1 := \sigma_0 p_1, \\ &\tilde{d}_1 := d_1 + r_1 p_1, \\ &\text{for } j = 1, 2, \ldots, m-1 \\ &\quad q_j := \begin{cases} (1/\tilde{d}_j) r_j, & \text{if } \tilde{d}_j \neq 0 \\ 0, & \text{if } \tilde{d}_j = 0, \end{cases} \\ &\quad \sigma_j := \sigma_{j-1} - q_j r_j \\ &\quad r_{j+1} := \sigma_j p_{j+1}, \\ &\quad \tilde{d}_{j+1} := d_{j+1} + r_{j+1} p_{j+1}. \end{aligned} \tag{14}$$

As in the rank-k product-form case, we do not form $\bar{L} = \tilde{L}_0 \tilde{L}_1 \tilde{L}_2 \ldots, \tilde{L}_k$; rather we store $\tilde{L}_0 = L$ as a sparse matrix and the pairs of vectors $p^{(j)}$ and $q^{(j)}$, $j = 1, 2, \ldots, k$. Since we need only the last $m-1$ components of each P_j, a total of $2k(m-1)$ amount of storage is required to represent $\bar{L}$ in addition to that needed for L. Solving $\tilde{L}(p,q)y = z$ and $\tilde{L}^T(p,q)y = z$ for y requires $2m-1$ multiplications. See ([11]) for details.

It is easy to derive from the recurrence formulas (14) that

$$\sigma_j := \begin{cases} \sigma_{j-1} d_j / \tilde{d}_j, & \text{if } \tilde{d}_j \neq 0 \\ 0, & \text{if } \tilde{d}_j = 0, \end{cases} \tag{15}$$

which can be used in place of the expression for σ_j in (14) as suggested by Gentleman [8]. In either case we shall refer to these recurrence formulas as the *rank-1 product-form* method. If we define $t_j \equiv 1/\sigma_j$, for all $j = 0, \ldots, m$, then we can with some additional algebra obtain from the above the following recurrence formulas first proposed by Fletcher and Powell [7] and Gill, Murray and Saunders [9].

$$
\begin{aligned}
& t_0 := 1/s, \\
& \text{for } j = 1, 2, \ldots, m \\
& \quad \text{if } t_{j-1} = \infty \text{ or } p_j = 0 \\
& \quad\quad t_j := t_{j-1} \\
& \quad\quad \tilde{d}_j := d_j \\
& \quad\quad \beta_j := 0 \\
& \quad \text{else} \\
& \quad\quad t_j := \begin{cases} t_{j-1} + p_j^2/d_j, & \text{if } d_j \neq 0 \\ \infty, & \text{if } d_j = 0, \end{cases} \\
& \quad\quad \tilde{d}_j := \begin{cases} d_j t_j / t_{j-1}, & \text{if } d_j \neq 0 \\ p_j^2/t_{j-1}, & \text{if } d_j = 0, \end{cases} \\
& \quad\quad \beta_j := \begin{cases} p_j/(d_j t_j), & \text{if } d_j \neq 0 \\ 1/p_j, & \text{if } d_j = 0. \end{cases}
\end{aligned}
\tag{16}
$$

As pointed out by Fletcher and Powell ([7]), even when $s > 0$, positive semidefiniteness of $\tilde{D}$ and hence of $\tilde{L}\tilde{D}\tilde{L}^T$ is not guaranteed in the presence of round-off errors if Bennett's version of the rank-1 product-form method is used. A similar statement applies to the rank-k method. The other two rank-1 methods do ensure that positive semidefiniteness is preserved when $s > 0$. However, when $s < 0$, all three rank-1 methods (and the rank-k method) can lose positive semidefiniteness. In this case, Fletcher and Powell ([7]) (see also Gill, Murray and Saunders ([9])) recommend that the quantities $t_j, j = 1, \ldots, m$ be computed first starting from the negative $t_0 = 1/s$ to make sure that no t_j becomes positive as this would violate the assumed positive semidefiniteness of $D + spp^T$. If this happens then they recommend that the t_j be computed from a backward recurrence starting from $t_m = \epsilon/s$, where ϵ is the relative machine precision so that $t_m < 0$ and has a magnitude comparable to the round-off errors in the computation. Observing that p_j must be zero if d_j is zero, also because of the assumed positive semidefiniteness of $D + spp^T$, this yields the following recurrence formulas for the case of $s < 0$:

$$
\begin{aligned}
& t_0 := 1/s, \\
& \text{for } j = 1, 2, \ldots, m \\
& \quad t_j := \begin{cases} t_{j-1} + p_j^2/d_j, & \text{if } d_j \neq 0 \\ t_{j-1}, & \text{if } d_j = 0, \end{cases} \\
& \text{if } t_j > 0, \text{ for any } j, \\
& \quad t_m := \epsilon/s, \\
& \quad \text{for } j = m, m-1, \ldots, 1 \\
& \quad\quad t_{j-1} := \begin{cases} t_j - p_j^2/d_j, & \text{if } d_j \neq 0 \\ t_j, & \text{if } d_j = 0, \end{cases}
\end{aligned}
\tag{17}
$$

$$
\begin{aligned}
&\text{for } j = 1, 2, \ldots, m \\
&\quad \text{if } d_j = 0 \\
&\quad\quad \tilde{d}_j := 0 \\
&\quad\quad \beta_j := 0 \\
&\quad \text{else} \\
&\quad\quad \tilde{d}_j := d_j t_j / t_{j-1}, \\
&\quad\quad \beta_j := p_j / (d_j t_j).
\end{aligned}
$$

It is possible to derive a rank-k version of the Fletcher and Powell and Gill, Murray and Saunders rank-1 recurrence formulas (17) by noting that the rank-k version of Gentleman's formula (15) is

$$\tilde{d}_j p^{(j)^T} \Sigma_j p^{(j)} = d_j p^{(j)^T} \Sigma_{j-1} p^{(j)},$$

and then defining $T_j \equiv \Sigma_j^{-1}$. If all $d_j \neq 0$, this results in the following recurrence:

$$
\begin{aligned}
&T_0 := S^{-1}, \\
&\text{for } j = 1, 2, \ldots, m \\
&\quad T_j := T_{j-1} + (1/d_j) p^{(j)} p^{(j)^T}, \\
&\quad \tilde{d}_j := d_j (p^{(j)^T} T_{j-1}^{-1} p^{(j)} / p^{(j)^T} T_j^{-1} p^{(j)}), \\
&\quad q_j := (1/d_j) T_j^{-1} p^{(j)}.
\end{aligned}
\tag{18}
$$

Because systems of the form $T_j r = p$ have to be repeatedly solved, this method will not be computationally efficient unless k is equal to 2 or 3 (see Table 1 below).

Block versions of the above methods can also be derived. For example, let us assume for simplicity that m is an integral multiple of k and that the final matrix is positive definite. Then if we define P_j to be the $k \times k$ matrix consisting of the j-th set of k rows of P, D_j to be a $k \times k$ symmetric positive definite matrix and R_j and Q_j to be $k \times k$ nonsymmetric matrices, the block version of the rank-k method is:

$$
\begin{aligned}
&\Sigma_0 := S, \\
&R_1 := \Sigma_0 P_1, \\
&\tilde{D}_1 := D_1 + R_1^T P_1, \\
&\text{for } j = 1, 2, \ldots, m-1 \\
&\quad Q_j := R_j \tilde{D}_j^{-1}, \\
&\quad \Sigma_j := \Sigma_{j-1} - Q_j R_j^T, \\
&\quad R_{j+1} := \Sigma_j P_{j+1},
\end{aligned}
\tag{19}
$$

$$\tilde{D}_{j+1} := D_{j+1} + R_{j+1}^T P_{j+1}.$$

This approach results in a block diagonal matrix $\tilde{D}$ containing $k \times k$ diagonal blocks and an $\tilde{L}$ of the form

$$\tilde{L} = \begin{bmatrix} I & & & & & \\ P_2Q_1 & I & & & & \\ P_3Q_1 & P_3Q_2 & I & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ P_{m-1}Q_1 & P_{m-1}Q_2 & P_{m-1}Q_3 & \cdots & I & \\ P_mQ_1 & P_mQ_2 & P_mQ_3 & \cdots & P_mQ_{m-1} & I \end{bmatrix}. \tag{20}$$

This method will only be competitive if k is very small as shown in Table 1 below.

Our product-form approach can now be outlined as follows. We separate the matrix AFA^T into its sparse and dense parts, as proposed in Section 2. After computing the factorization LDL^T of the sparse part $A\bar{F}A^T = \sum_{i=1}^N A_i\bar{F}_iA_i^T$ of AFA^T, we update this factorization using one of the product-form methods described above to take care of the dense part $\sum_{i\in\mathcal{I}}(v_iv_i^T - w_iw_i^T)$. If any A_i has dense columns these can be handled as part of this dense sum of outer products. Notice that a number of possibilities exist. For example, one can treat recursively each rank-2 outer product pair $(v_iv_i^T - w_iw_i^T)$ by the rank-k Bennett update where $k = 2$. Also, notice that for each cone the addition of the rank-one term $v_iv_i^T$ has to preceed or be done at the same time as the subtraction of the corresponding $w_iw_i^T$ term; otherwise the positive semidefiniteness of the matrix may not be preserved. Apart from that, the order of addition and subtraction of the terms may vary.

Let us now compute the workload of the proposed approaches. We assume that $A\bar{F}A^T$ is sparse – i.e., $A\bar{F}A^T$ has $\Theta(m)$ nonzero elements – and that computing its Cholesky factorization requires $\Theta(m^2)$ multiplications. We also assume that $n = \Theta(m)$. Henceforth, we will only count multiplications (and divisions); the number of additions (and subtractions) is closely correlated with this count. We assume that the matrix L is sparse and also has $\Theta(m)$ nonzero elements and that performing back or forward solves with L requires $\Theta(m)$ multiplications. Combining this with the work required to solve systems involving the special matrices (9), (13) and (20) and to execute the recurrence relations (10),(14),(16),(17), (18) and (19) yields the work required by the various product-form update methods in Table 1.

Clearly, the rank-1 methods require the least amount of work. However, some steps in the rank-k methods can easily take advantage of parallel computation. For example, the solution of $LP = V$ for P can done on k parallel processors, each computing one column of P. Therefore, in a parallel computing environment, the rank-k Bennett method might prove to be the most efficient. The rank-1 methods can also take advantage of parallelism to compute the k vectors $P_1, \ldots, P_k$ (but

Table 1. Work to Compute Product-Form Factorization Updates

Method	Work for Rank-k Update	Work for $k=2$	Work for $k=4$
rank-1, FP (16)	$k\Theta(m)+(k^2+4k)m+o(km)$	$2\Theta(m)+12m$	$4\Theta(m)+32m$
rank-1 Bennett (14)	$k\Theta(m)+(k^2+3k)m+o(km)$	$2\Theta(m)+10m$	$4\Theta(m)+28m$
rank-k Bennett (10)	$k\Theta(m)+\frac{3}{2}k^2+\frac{5}{2}k)m+o(km)$	$2\Theta(m)+11m$	$4\Theta(m)+34m$
rank-k FP (18)	$k\Theta(m)+(\frac{1}{6}k^3+4k^2+\frac{23}{6}k)m+o(km)$	$2\Theta(m)+25m$	$4\Theta(m)+90m$
block Bennett (19)	$k\Theta(m)+(\frac{13}{6}k^3+4k^2-\frac{7}{6}4k)m+o(km)$	$2\Theta(m)+31m$	$4\Theta(m)+198m$

not as fully as the rank-k methods) as follows: let $\tilde{P}_0 \equiv V$ and for $j=0,\ldots,k-1$ solve $\tilde{L}\hat{P}_j=\tilde{P}_j$ for $\tilde{P}_j$ and set $[P_{j+1},\tilde{P}_{j+1}]=\hat{P}_j$ (i.e.,P_{j+1} is the first column of $\hat{P}_j$ and $\tilde{P}_{j+1}$ is the matrix of the remaining $k-(j+1)$ columns of $\hat{P}_j$. Note $\hat{P}_j \in \mathbf{R}^{m\times(k-j)}$ and $\tilde{P}_j \in \mathbf{R}^{m\times(k-j)}$.

Finally, let us compare the work required to solve the system $AFA^T\Delta y=r_y$ by computing the product-form Cholesky factorization (PFCF) using the rank-1 updates (16) and (17) to that required by computing the Cholesky factorization of the whole matrix. We assume that A is sparse and that the problem involves $|\mathcal{I}|=k/2$ large cones. It is easy to see that from Table 1 and our earlier remarks that

$$\textbf{Work(PFCF)} \qquad \Theta(m^2)+2(k+1)\Theta(m)+(k^2+4k)m+o(km).$$

The full matrix AFA^T is expected to have $\Theta(\sum_{i\in\mathcal{I}}n_i^2)$ nonzeros and computing its Cholesky factorization requires $\Theta(m(\sum_{i\in\mathcal{I}}n_i^2))$ multiplications. The total workload is

$$\textbf{Work(CF)} \qquad \Theta((\sum_{i\in\mathcal{I}}n_i^2)m)+2\Theta(\sum_{i\in\mathcal{I}}n_i^2)+o(nm).$$

If $n_i\approx\sqrt{m}$ and $k\approx\sqrt{m}$, then the work required by the PFCF approach $(\Theta(m^2))$, is a factor of the order of $\sqrt{m}$ less than that required by the standard Cholesky approach $(\Theta(m^{5/2})$. If $n_i\sim m$ (i.e., n_i is of the same order of magnitude as m) and $k\sim 1$ then the savings factor is of the order of m, and if $n_1\sim 1$ and $k\sim 1$ then there is no benefit in using PFCF.

4. Analysis

In this section we consider numerical properties of both the Cholesky factorization LDL^T and the product-form Cholesky factorization $\tilde{L}_0\tilde{L}_1\tilde{L}_2\ldots,\tilde{L}_k\tilde{D}^k\tilde{L}_k^T\ldots\tilde{L}_2^T\tilde{L}_1^T\tilde{L}_0^T$ of $M=AFA^T$. Specifically, we show that the Cholesky factors L and $\tilde{L}_i$, for $i=0,1,\ldots,k$ are uniformly bounded as $\mu\to 0$ as long as the interior point iterates stay close to the central path.

The results in this section are analogous to similar results proved in [11] for the LP case. The main difference is that in the LP case the matrix F is diagonal, and the uniform boundedness of L continues to hold for any diagonal matrix

$F(\mu)$. This means that the iterates of an interior point method can converge to the optimal solution in an arbitrary manner. In the SOCP case we will assume that the iterates converge to a strictly complementary optimal solution and all lie within some central neighborhood. Our proofs here are also more complicated (although we tried to use similar techniques wherever possible) due to the more complicated structure of the normal equations matrix and the nature of SOCP. Our analysis makes use of the eigenvalue/eigenvector decomposition of F: $F = Q\Lambda Q^T$.

4.1. Eigen-structure of F

It is straightforward to verify that the eigenvalue/eigenvector decomposition of F_i is:

$$F_i = Q_i \Lambda_i Q_i^T ,$$

where

$$\Lambda_i = \frac{1}{2} \begin{bmatrix} (u_i^0 + \|\bar{u}_i\|)^2 & & & & \\ & \gamma^2(u_i) & & & \\ & & \ddots & & \\ & & & \gamma^2(u_i) & \\ & & & & (u_i^0 - \|\bar{u}_i\|)^2 \end{bmatrix}, \tag{21}$$

$$Q_i = \begin{bmatrix} 1/\sqrt{2} & 0 \cdots 0 & 1/\sqrt{2} \\ \bar{u}_i/\sqrt{2}\|\bar{u}_i\| & q_i^2 \cdots q_i^{n_i-1} & -\bar{u}_i/\sqrt{2}\|\bar{u}_i\| \end{bmatrix}, \tag{22}$$

and $q_i^2, \ldots, q_i^{n_i-1}$ are such that Q_i is an orthogonal matrix.

For our analysis we need to estimate the rate of convergence of the elements in Q_i and Λ_i to their respective limit values as $\mu \to 0$. To do so we use the following notation. Let $\phi(\mu)$ and $\psi(\mu)$ be arbitrary functions of a parameter $\mu \geq 0$, in whose behavior we are interested as $\mu \to 0$. We write $\psi = O(\phi)$ ($\psi = \Theta(\phi)$) as $\mu \to 0$ if there are positive constants γ_1 and γ_2, independent of μ, such that $|\psi(\mu)| \leq \gamma_2|\phi(\mu)|$ ($\gamma_1|\phi(\mu)| \leq |\psi(\mu)| \leq \gamma_2|\phi(\mu)|$) for all $\mu > 0$ sufficiently small. If ψ is a vector or a matrix valued function, then $\psi = O(\phi)$ means that every component of ψ is $O(\phi)$ - i.e., $O(\phi)$ may represent a vector or matrix of values. This should be evident from the context. We say that ϕ is uniformly bounded as $\mu \to 0$ if there exists a constant γ_3, independent of μ, such that $|\phi(\mu)| \leq \gamma_3$ for all $\mu > 0$ sufficiently small. The notation $\psi = O(\mu)$ and $\psi = O(1)$ are special cases with $\phi(\mu) \equiv \mu$ and $\phi(\mu) \equiv 1$. It is easy to see that $\psi = O(1)$ is equivalent to ψ being uniformly bounded.

Assume that we have a primal-dual interior solution (x, y, z) near the central path with duality gap μ. We know that, under the assumption that there exists a strictly complementary optimal solution, the analytic center of the optimal face (x^*, y^*, z^*) is such a solution and moreover $\|(x, y, z) - (x^*, y^*, z^*)\| = \Theta(\mu)$ [17].

To simplify matters, let us now focus on the purely primal case. For solutions (x, y, z) closely following the central path we can write $x = x^* + \tilde{x}(\mu)\mu$ and

Table 2. Size of Eigenvalues for Three Types of Second-Order Cones

Type of Cone	Type 1 $x_i^* = \Theta(1)$ $\gamma(x_i^*) = \Theta(1)$	Type 2 $x_i^* = 0$ $\gamma(x_i^*) = 0$	Type 3 $x_i^* = \Theta(1)$ $\gamma(x_i^*) = 0$
$\gamma^2(x_i)$	$\Theta(1)$	$\Theta(\mu^2)$	$\Theta(\mu)$
x_i	$\Theta(1)$	$\Theta(\mu)$	$\Theta(1)$
$x_i^0 + \lvert\lvert\bar{x}_i\rvert\rvert$	$\Theta(1)$	$\Theta(\mu)$	$\Theta(1)$
$x_i^0 - \lvert\lvert\bar{x}_i\rvert\rvert$	$\Theta(1)$	$\Theta(\mu)$	$\Theta(\mu)$
Eigenvalues			
$(x_i^0 + \lvert\lvert\bar{x}_i\rvert\rvert)^2/2$	$\Theta(1)$	$\Theta(\mu^2)$	$\Theta(1)$
$\gamma^2(x_i)/2$	$\Theta(1)$	$\Theta(\mu^2)$	$\Theta(\mu)$
$(x_i^0 - \lvert\lvert\bar{x}_i\rvert\rvert)^2/2$	$\Theta(1)$	$\Theta(\mu^2)$	$\Theta(\mu^2)$

$z = z^* + \tilde{z}(\mu)\mu$, where $\tilde{x}(\mu) = (x - x^*)/\mu$, $\tilde{z}(\mu) = (z - z^*)/\mu$ and the elements of x^*, $\tilde{x}(\mu)$, z^* and $\tilde{z}(\mu)$ are either $\Theta(1)$ or zero. Using the fact that

$$\gamma^2(x_i) = \gamma^2(x_i^*) + 2((x_i^*)^0\tilde{x}_i^0(\mu) - \sum_{j=1}^{n_i}(x_i^*)^j\tilde{x}_i^j(\mu))\mu + \gamma^2(\tilde{x}_i(\mu))\mu^2.$$

and the assumptions above, we can easily derive the entries that appear in Table 2. The entry for $x_i^0 - ||\bar{x}_i||$ for type 3 cones follows from the fact that $\gamma^2(x_i) = (x_i^0 + ||\bar{x}_i||)(x_i^0 - ||\bar{x}_i||)$.

For purely dual interior point methods the possible eigenvalues have sizes $\Theta(1)$, $\Theta(1/\mu)$ and $\Theta(1/\mu^2)$, while for primal-dual methods the eigenvalue sizes are $\Theta(\mu)$, $\Theta(1)$ and $\Theta(1/\mu)$.

Recall that $AFA^T = \sum_{i=1}^N A_i\bar{F}_iA_i^T + \sum_{i\in\mathcal{I}}(v_iv_i^T - w_iw_i^T)$, where $\mathcal{I} \subseteq \{1,\ldots,N\}$ is the subset of indices of "large" cones, $\bar{F}_i = F_i$ as defined by (6) for $i \in \{1,\ldots,N\}\backslash\mathcal{I}$, and $\bar{F}_i = \frac{\gamma^2(u_i)}{2}I$ for $i \in \mathcal{I}$. For our stability analysis of the Cholesky factorization and PFCF methods we need to consider the eigenstructure of $\bar{F}$ and $\hat{F} = \bar{F} + \sum_{i\in\mathcal{I}_1} x_ix_i^T$, where $\mathcal{I}_1$ is a subset of $\mathcal{I}$. The eigenstructure of $\bar{F}$ follows trivially from that of F, since the diagonal blocks of $\bar{F}$ either coincide with those of F or are equal to $\gamma^2(x_i)/2$. The eigenstructure for $\hat{F}$ can also be easily derived from that of F and the eigenvalue/eigenvector decomposition $\hat{Q}\hat{\Lambda}\hat{Q}^T$ of $\hat{F}_i = \frac{\gamma^2(x_i)}{2}I + x_ix_i^T$:

$$\hat{\Lambda}_i = \frac{1}{2}\begin{bmatrix} \gamma^2(x_i) + 2||x_i||^2 & & & \\ & \gamma^2(x_i) & & \\ & & \ddots & \\ & & & \gamma^2(x_i) \end{bmatrix}, \tag{23}$$

$$\hat{Q}_i = \left[\frac{x_i}{||x_i||}, \hat{q}_i^2, \cdots, \hat{q}_i^{n_i}\right], \tag{24}$$

where $\hat{q}_i^2, \ldots, \hat{q}_i^{n_i}$ are such that $\hat{Q}_i$ is an orthogonal matrix. It is easy to see from Table 2 that the eigenvalues of $\hat{F}_i$ are all $\Theta(1)$ for a type 1 cone, all $\Theta(\mu^2)$ for a type 2 cone and either $\Theta(1)$ or $\Theta(\mu)$ for a type 3 cone.

4.2. Numerical Analysis of the LDL^T factorization of AFA^T

In this section we analyze the numerical behavior of the Cholesky factors L and $\tilde{L}_i$, for $i = 1, \ldots, k$ of F as $\mu \to 0$. As in the previous section, we focus on the purely primal case for simplicity. ¿From the discussion in Section 4.1, expressions (22), (21) for Q and Λ and the fact that $x(\mu) = x^* + O(\mu)$ we have that some of the diagonal elements of Λ are bounded away from 0 as $\mu \to 0$, some elements of Λ converge to zero with the rate $\Theta(\mu)$ and some with the rate of $\Theta(\mu^2)$ as $\mu \to 0$. Let us partition the matrices Q and Λ according to the rate of convergence of the elements in Λ: $Q = [Q_1, Q_\mu, Q_{\mu^2}]$, and $\Lambda = \mathrm{diag}[\Lambda_1, \Lambda_\mu, \Lambda_{\mu^2}]$. Also, let $Q_{1,\mu} = [Q_1, Q_\mu]$, $Q_{\mu,\mu^2} = [Q_\mu, Q_{\mu^2}]$, $\Lambda_{1,\mu} = \mathrm{diag}[\Lambda_1, \Lambda_\mu]$, and $\Lambda_{\mu,\mu^2} = \mathrm{diag}[\Lambda_\mu, \Lambda_{\mu^2}]$.

Lemma 1. *If L is a lower triangular matrix with ones on the diagonal and subdiagonal elements that depend on a parameter μ and are uniformly bounded as $\mu \to 0$, then the elements of L^{-1} are also uniformly bounded as $\mu \to 0$.*

Proof. If some element of L^{-1} diverges to ∞ as $\mu \to 0$, then $L^{-T}L^{-1}$ has a diagonal element that diverges to ∞ as $\mu \to 0$, implying that LL^T has an infinitesimally small eigenvalue, which is not possible because $\det(LL^T) = 1$ and the largest eigenvalue of LL^T is bounded. □

Lemma 2. *Let $A \in \mathbf{R}^{m\times n}$ be a fixed matrix and, for all μ, let $Q'(\mu)$ be a matrix whose columns are orthonormal and $r(\mu) \in \mathbf{R}^m$ be a vector, all of whose elements are uniformly bounded and whose k-th element $r_k(\mu) \equiv 1$ and $r_j(\mu) \equiv 0$ $\forall j > k$. If $r(\mu)^T AQ'(\mu) \to 0$ as $\mu \to 0$, then for any accumulation point $\bar{Q}'$ of the set $Q'(\mu)$ as $\mu \to 0$, the k-th row of $A\bar{Q}'$ is linearly dependent on the first $k-1$ rows of $A\bar{Q}'$.*

Proof. Consider a sequence $\{\mu_s\} \to 0$ for which $Q'(\mu_s) \to \bar{Q}'$, and a subsequence $\{\mu_{s_t}\} \to 0$ of this sequence such that $r(\mu_{s_t}) \to \bar{r}$, where $\bar{r}$ is some accumulation point of the set of $r_s(\mu_s)$ as $\mu_s \to 0$. Clearly, $\bar{r}_k = 1$ and $\bar{r}_j = 0$ $\forall j > k$ and

$$r(\mu_{s_t})^T AQ(\mu_{s_t})' \to \bar{r}^T A\bar{Q}' = 0.$$

This trivially implies that the k-th row of $A\bar{Q}'$ is linearly dependent on the first $k-1$ rows of $A\bar{Q}'$. □

Lemma 3. *Let $Q = [Q', Q'']$ be an orthogonal $n \times n$ matrix, partitioned columnwise into two submatrices $Q' \in \mathbf{R}^{n\times n_1}$ and $Q'' \in \mathbf{R}^{n\times n_2}$, with $n_1 + n_2 = n$. Let $A \in \mathbf{R}^{m\times n}$ and $a_j^T = e_j^T A$ be the j-th row of A. If for some k*

$$a_k^T Q' = \sum_{j=1}^{k-1} \alpha_j a_j^T Q', \tag{25}$$

where α_1, …, α_{k-1} are arbitrary scalars, then

$$a_k = \sum_{j=1}^{k-1} \alpha_j a_j + Q''\beta, \tag{26}$$

where β is an n_2-dimensional vector.

Proof. Since $QQ^T = Q'(Q')^T + Q''(Q'')^T = I$

$$a_k - \sum_{j=1}^{k-1} \alpha_j a_j = Q'(Q')^T (a_k - \sum_{j=1}^{k-1} \alpha_j a_j) + Q''(Q'')^T (a_k - \sum_{j=1}^{k-1} \alpha_j a_j). \tag{27}$$

¿From equation (25), the first term of the right hand side of equation (27) is zero. Hence, by setting $\beta = (Q'')^T (a_k - \sum_{j=1}^{k-1} \alpha_j a_j)$ we obtain the desired result. □

Lemma 4. *Let the i-th cone in (P) be a cone of type 3. Let $x_i(\mu) = x_i^* + O(\mu)$, as $\mu \to 0$, where x_i^* is an optimal solution corresponding to the i-th cone in (P). Let $Q_i(\mu)$ be the eigenvector matrix defined by (22), and let $\bar{Q}_i$ be any accumulation point of $Q_i(\mu)$, as $\mu \to 0$. Let $q_1(\mu)$ be the eigenvector that corresponds to the $\Theta(1)$ eigenvalue of F_i, $q_{\mu^2}(\mu)$ be the eigenvector that corresponds to the $\Theta(\mu^2)$ eigenvalue, and $q_\mu(\mu)$ be any eigenvector corresponding to one of the $\Theta(\mu)$ eigenvalues. Let $\bar{q}_1$, $\bar{q}_{\mu^2}$ and $\bar{q}_\mu$ be the appropriate columns of $\bar{Q}_i$. Then,*

$$q_1(\mu)^T \bar{q}_\mu = O(\mu)$$

$$q_\mu(\mu)^T \bar{q}_{\mu^2} = O(\mu)$$

$$q_1(\mu)^T \bar{q}_{\mu^2} = O(\mu^2).$$

Proof.

First we will show that $q_1(\mu) = \bar{q}_1 + O(\mu)$ and $q_{\mu^2}(\mu) = \bar{q}_{\mu^2} + O(\mu)$. Now,

$$Q_i = \begin{bmatrix} 1/\sqrt{2} & 0 \cdots 0 & 1/\sqrt{2} \\ \bar{x}_i/\sqrt{2}||\bar{x}_i|| & q_i^2 \cdots q_i^{n_i-1} & -\bar{x}_i/\sqrt{2}||\bar{x}_i|| \end{bmatrix},$$

Since the i-th cone is of type 3, $\bar{x}_i^* \neq 0$. We know that $\bar{x}_i(\mu) = \bar{x}_i^* + O(\mu)$; hence, $\|\bar{x}_i(\mu)\| = \|\bar{x}_i^*\| + O(\mu)$ from the properties of the norm. We will omit the index i for the remainder of the proof, since it stays the same. It is easy to see now that

$$\bar{x}(\mu)/\sqrt{2}||\bar{x}(\mu)|| \to \bar{x}^*(\mu)/\sqrt{2}||\bar{x}^*(\mu)|| + O(\mu),$$

and hence that $q_1(\mu) = \bar{q}_1 + O(\mu)$ and $q_{\mu^2}(\mu) = \bar{q}_{\mu^2} + O(\mu)$. Then the first two relations of the lemma follow automatically from the fact that $q_1(\mu)^T \bar{q}_\mu = \bar{q}_1^T \bar{q}_\mu + O(\mu)^T \bar{q}_\mu = O(\mu)^T \bar{q}_\mu$ and $q_\mu(\mu)^T \bar{q}_{\mu^2} = q_\mu(\mu)^T q_{\mu^2}(\mu) + q_\mu(\mu)^T O(\mu) = q_\mu(\mu)^T O(\mu)$.

To show that $q_1(\mu)^T \bar{q}_{\mu^2} = (1/\sqrt{2}, \bar{x}(\mu)^T/\sqrt{2}||\bar{x}(\mu)||)(1/\sqrt{2}, -(\bar{x}_i^*)^T/\sqrt{2}||\bar{x}^*||)^T = O(\mu^2)$, for any given value of μ let us write $\bar{x}(\mu) = \bar{x}^* + \mu\hat{x}$, where $\hat{x}$ depends on μ. Then

$$q_1(\mu)^T \bar{q}_{\mu^2} = \frac{1}{2} - \frac{(\bar{x}^* + \mu\hat{x})^T \bar{x}^*}{2||\bar{x}^* + \mu\hat{x}|| \cdot ||\bar{x}^*||} = \frac{1}{2||\bar{x}^* + \mu\hat{x}|| \cdot ||\bar{x}^*||}(||\bar{x}^* + \mu\hat{x}|| \cdot ||\bar{x}^*|| - (\bar{x}^*)^T \bar{x}^* - \mu\hat{x}^T \bar{x}^*).$$

Since $\bar{x}^* \neq 0$, we only need to show that the expression in the parenthesis is $O(\mu^2)$. It can be rewritten as

$$\sqrt{((\bar{x}^*)^T\bar{x}^*)((\bar{x}^*)^T\bar{x}^* + 2\mu\hat{x}^T\bar{x}^* + \mu^2\hat{x}^T\hat{x})} - (\bar{x}^*)^T\bar{x}^* - \mu\hat{x}^T\bar{x}^* =$$
$$\sqrt{((\bar{x}^*)^T\bar{x}^* + \mu\hat{x}^T\bar{x}^*)^2 + (\hat{x}^T\hat{x} - (\hat{x}^T\bar{x}^*)^2)\mu^2} - (\bar{x}^*)^T\bar{x}^* - \mu\hat{x}^T\bar{x}^* = O(\mu^2).$$

The last relation is due to the Taylor series expansion of $\sqrt{a+\mu^2}$.

□

Lemma 5. *Let the conditions of Lemma 4 hold and let $Q_i(\mu)$ be the eigenvector matrix defined by (24) (instead of (22)). Let $q_1(\mu)$, $q_\mu(\mu)$, $\bar{q}_1$, and $\bar{q}_\mu$ be defined as in Lemma 4 (in this case there are no $q_{\mu^2}(\mu)$ and $\bar{q}_{\mu^2}$). Then,*

$$q_1(\mu)^T\bar{q}_\mu = O(\mu).$$

Proof. The proof is a trivial repetition of the first part of the proof of Lemma 4, adapted to the $Q_i(\mu)$ defined by (24). □

Corollary 1. *Let $x(\mu) = x^* + O(\mu)$, as $\mu \to 0$, where x^* is an optimal solution to (P). Let $Q(\mu)$ be the block diagonal eigenvector matrix some of whose blocks are defined by (22) and some by (24), and let $\bar{Q}$ be any accumulation point of $Q(\mu)$, as $\mu \to 0$. Consider the partition $Q = [Q_1, Q_\mu, Q_{\mu^2}]$ described at the beginning of this subsection. Then*

$$Q_1(\mu)^T\bar{Q}_\mu = O(\mu)$$

$$Q_\mu(\mu)^T\bar{Q}_{\mu^2} = O(\mu)$$

$$Q_1(\mu)^T\bar{Q}_{\mu^2} = O(\mu^2).$$

Proof. Since matrices $Q(\mu)$ and $\bar{Q}$ are block diagonal, the inner product of a column from $Q(\mu)$ with one from $\bar{Q}$ is always zero unless these columns come from the same cone. We know that for cones of type 1 or 2 the eigenvalues of the matrices F_i or $\hat{F}_i$ are either all $\Theta(1)$ or all $\Theta(\mu^2)$. Hence the only nontrivial case is when columns in $Q_1(\mu)$ and $\bar{Q}_\mu$ (or $Q_\mu(\mu)$ and $\bar{Q}_{\mu^2}$, or $Q_1(\mu)$ and $\bar{Q}_{\mu^2}$) come from the same cone and that cone is of type 3. Hence, the statement of this corollary follows immediately from the Lemmas 4 and 5.

□

Now we state and prove the main theorem.

Theorem 1. *Let A be a fixed $m \times n$ matrix, $\mathcal{F}(\mu)$ be a block diagonal matrix whose blocks are either the primal scaling matrices defined by (6) corresponding to the primal solution $x(\mu) = x^* + O(\mu)$, or the matrices $\bar{F}_i$ or $\hat{F}_i$ defined in Section 4.1, and LDL^T be the Cholesky factorization of $M = A\mathcal{F}(\mu)A^T$. Then the elements of L are uniformly bounded as $\mu \to 0$.*

Proof. Assume the statement of the theorem is not true. Then there exist a sequence $\mu_s \to 0$ such that $|l_{ik}(\mu_s)| \to \infty$ for some indices i and k. Among all such pairs of indices (i,k) let us consider a pair with the smallest k (and an arbitrary i). Clearly $i > k$ since the matrix L is lower triangular with a unit diagonal. Let us define (using standard Matlab notation to denote submatrices of a matrix)

$$L^{(k-1)} = \begin{bmatrix} L_{1:k-1,1:k-1} & 0 \\ L_{k:m,1:k-1} & I \end{bmatrix}$$

and

$$\bar{M} = [L^{(k-1)}]^{-1} M [L^{(k-1)}]^{-T} = \begin{bmatrix} D_{1:k-1,1:k-1} & 0 \\ 0 & M_{k:m,k:m} - L_{k:m,1:k-1} D_{1:k-1,1:k-1} [L_{k:m,1:k-1}]^{T} \end{bmatrix}.$$

Here we have used the fact that $L^{(k-1)}$ is the partial Cholesky factor (with unit diagonal) of M and, hence, for example,

$$[L^{(k-1)}_{1:k-1,1:k-1}]^{-1} M_{1:k-1,1:k-1} [L^{(k-1)}_{1:k-1,1:k-1}]^{-T} = D_{1:k-1,1:k-1}.$$

The elements of $L^{(k-1)}$ are uniformly bounded since k is the smallest column index for which this property is violated. Hence, by Lemma 1, the elements of $(L^{(k-1)})^{-1}$ are uniformly bounded.

Let us define for $j = 1, \ldots, m$,

$$r_j^T = e_j^T (L^{(k-1)})^{-1}.$$

Then

$$\bar{m}_{kk} = r_k^T M r_k,$$
$$\bar{m}_{ik} = r_i^T M r_k$$

and from the properties of Cholesky factorization and our assumptions on l_{ik},

$$|l_{ik}| = |\bar{m}_{ik}|/\bar{m}_{kk} \to \infty. \tag{28}$$

Let us now consider a partition $(\mathcal{I}', \mathcal{I}'')$ of the indices of the diagonal elements of Λ according to their rates of convergence to 0 relative to the rate of convergence of $\bar{m}_{kk}$; i.e., $\bar{m}_{kk}/\lambda_j \to 0$ for all $j \in \mathcal{I}'$ as $\mu_s \to 0$ and $\lambda_j/\bar{m}_{kk}$ is uniformly bounded for all $j \in \mathcal{I}''$ as $\mu_s \to 0$.

Let $[Q', Q'']$ denote a partition of Q such that $Q' \in \mathbf{R}^{m \times |\mathcal{I}'|}$ is the matrix composed of those columns of Q whose indices are in $\mathcal{I}'$ and $Q'' \in \mathbf{R}^{m \times |\mathcal{I}''|}$ is the matrix composed of those columns of Q whose indices are in $\mathcal{I}''$. Similarly we define $\Lambda' = \mathrm{diag}(\{\lambda_j \,|\, j \in \mathcal{I}'\}) \in \mathbf{R}^{|\mathcal{I}'| \times |\mathcal{I}'|}$ and $\Lambda'' = \mathrm{diag}(\{\lambda_j \,|\, j \in \mathcal{I}''\}) \in \mathbf{R}^{|\mathcal{I}''| \times |\mathcal{I}''|}$.

We have

$$\bar{m}_{kk} = r_k^T A Q' \Lambda' (Q')^T A^T r_k + r_k^T A Q'' \Lambda'' (Q'')^T A^T r_k.$$

Since both terms in the above sum are nonnegative, both are $O(\bar{m}_{kk})$. ¿From the definition of Λ', every element of Λ' converges to zero more slowly than

$\bar{m}_{kk}$. Since the first term in the sum is $O(\bar{m}_{kk})$ it has to hold that $r_k^T AQ' \to 0$. Consider any subsequence $\{\mu_{s_t}\}$ of the sequence μ_s for which $Q \to \bar{Q}$, where $\bar{Q}$ is some limit point of Q as $\mu \to 0$. We now can apply Lemmas 2 and 3 to conclude that

$$a_k = \sum_{j=1}^{k-1} \alpha_j a_j + \bar{Q}''\beta,$$

where $a_j^T = e_j^T A$, $j = 1, \ldots, k$ and the α_j are fixed coefficients and β is a fixed column vector.

Now, from the definition of $\bar{M}$ it is easy to show that $\bar{m}_{ik} = r_i^T M r_k = r_i^T M e_k$; hence

$$\begin{aligned}\bar{m}_{ik} &= r_i^T AQ\Lambda Q^T a_k = r_i^T AQ'\Lambda'(Q')^T a_k + r_i^T AQ''\Lambda''(Q'')^T a_k \\ &= r_i^T AQ'\Lambda'(Q')^T (\sum_{j=1}^{k-1} \alpha_j a_j + \bar{Q}''\beta) + r_i^T AQ''\Lambda''(Q'')^T a_k.\end{aligned}$$

But, $r_i^T AQ\Lambda Q^T a_j = r_i^T (AQ'\Lambda'(Q')^T a_j + AQ''\Lambda''(Q'')^T a_j) = 0$, $1 \leq j < k < i$. Therefore, for all $i > k$

$$\sum_{j=1}^{k-1} \alpha_j r_i^T AQ'\Lambda'(Q')^T a_j = -\sum_{j=1}^{k-1} \alpha_j r_i^T AQ''\Lambda''(Q'')^T a_j = O(\bar{m}_{kk}). \tag{29}$$

Clearly,

$$r_i^T AQ''\Lambda''(Q'')^T a_k = O(\bar{m}_{kk}). \tag{30}$$

To complete our estimate of the rate of convergence of $\bar{m}_{ik}$ to zero, we need to estimate the expression

$$\sigma_i \equiv r_i^T AQ'\Lambda'(Q')^T \bar{Q}''\beta.$$

To do this, consider the following four possible cases:

i $\Lambda' = [\ ]$, $Q' = [\ ]$, $\Lambda'' = Q$ and $Q'' = Q$, where '[ ]' denotes an empty matrix;
ii $\Lambda' = \Lambda_1$, $Q' = Q_1$, $\Lambda'' = \Lambda_{\mu,\mu^2}$ and $Q'' = Q_{\mu,\mu^2}$;
iii $\Lambda' = \Lambda_{1,\mu}$, $Q' = Q_{1,\mu}$, $\Lambda'' = \Lambda_{\mu^2}$ and $Q'' = Q_{\mu^2}$; and
iv $\Lambda' = \Lambda$, $Q' = Q$, $\Lambda'' = [\ ]$ and $Q'' = [\ ]$.

In cases (i) and (iv), $\sigma_i = 0$ since $(Q')^T \bar{Q}'' = 0$. Recall that r_i and β are bounded. Therefore, in case (ii), $\sigma_i = O(\mu)$, since $\Lambda'(Q')^T \bar{Q}'' = [\Lambda_1(Q_1)^T \bar{Q}_\mu, \Lambda_1(Q_1)^T \bar{Q}_{\mu^2}] = [O(\mu), O(\mu^2)]$ by Corollary 1. Moreover, from the boundedness of $\Lambda''/\bar{m}_{kk} = \Lambda_{\mu,\mu^2}/\bar{m}_{kk}$, $\mu = O(\bar{m}_{kk})$; hence $\sigma_i = O(\bar{m}_{kk})$. In case (iii), $\sigma_i = O(\mu^2)$, since $\Lambda'(Q')^T \bar{Q}'' = \begin{bmatrix} \Lambda_1(Q_1)^T \bar{Q}_{\mu^2} \\ \Lambda_\mu(Q_\mu)^T \bar{Q}_{\mu^2} \end{bmatrix} = O(\mu^2)$ by Corollary 1. Moreover, from the boundedness of $\Lambda''/\bar{m}_{kk} = \Lambda_{\mu^2}/\bar{m}_{kk}$, $\mu^2 = O(\bar{m}_{kk})$; hence $\sigma_i = O(\bar{m}_{kk})$.

Thus $\sigma_i = O(\bar{m}_{kk})$ in all cases, which when combined with (29) and (30) yields

$$|\bar{m}_{ik}| = O(\bar{m}_{kk})$$

for some subsequence $\{\mu_{s_t}\}$ of the sequence $\mu_s \to 0$. Hence $|\bar{m}_{ik}|/\bar{m}_{kk}$ is uniformly bounded as the sequence μ_{s_t} converges to 0. This contradicts the assumption that $|l_{ik}(\mu_s)| \to \infty$ since this implies that $|l_{ik}(\mu_{s_t})| \to \infty$. □

4.3. Numerical Analysis of the Product-Form Factorization

We will now show that the uniform boundedness of the elements of the Cholesky factors of the normal equations matrix extends to the elements of all unit lower triangular matrices that make up the product-form factorizations described in Section 3.

Recall the definition of matrix $\mathcal{F}$ in the statement of Theorem 1. Here we consider the matrix

$$AFA^T = A\mathcal{F}A^T + VSV^T = \bar{L}\tilde{L}D\tilde{L}^T\bar{L}^T, \tag{31}$$

where $\bar{L}\bar{D}\bar{L}^T$ is the factorization of $A\mathcal{F}A^T$. Let LDL^T be the factorization of the complete matrix AFA^T. Hence, we have $L = \bar{L}\tilde{L}$.

¿From Theorem 1 we know that the elements of L and $\bar{L}$ are uniformly bounded as $\mu \to 0$. Hence the elements of $\bar{L}^{-1}$ are uniformly bounded by Lemma 1. Then from $L = \bar{L}\tilde{L}$ it immediately follows that $\tilde{L} = \bar{L}^{-1}L$ is uniformly bounded.

Notice that the rank-one addition and rank-one subtraction updates are special cases of (31). The case of multiple updates follows by induction. Thus, we have the following theorem.

Theorem 2. *Let $\tilde{L}_0\tilde{L}_1\tilde{L}_2\ldots\tilde{L}_k\tilde{D}^k\tilde{L}_k^T\ldots\tilde{L}_2^T\tilde{L}_1^T\tilde{L}_0^T$ be a product-form Cholesky factorization of $M = AFA^T$ obtained by any method described in Section 3. Then the elements of $\tilde{L}_i$ for any $i = 0,\ldots,k$ are uniformly bounded as $\mu \to 0$.*

We refer the reader to [11] for the error analysis of the Fletcher-Powell rank-one product-form Cholesky factorization method when there are only rank-one additions. It is shown there that the product-form factorization is numerically stable when applied to LP problems, as long as the initial data is well scaled. This analysis can be easily extended to rank-one subtractions and hence can be applied to the PFCF of the normal equations for SOCP problems.

In [11] a pivot skipping strategy within the product-form Cholesky factorization procedure, similar in spirit to that of Wright [30], was proposed and analyzed. A similar strategy can be used for SOCP. However, in our numerical experiments we did not encounter a situation where pivot skipping was needed or useful. The main reason for this is that interior point methods for SOCP rarely achieve as much accuracy as they do for LP problems; thus, they typically stop before any pivot skipping is applicable.

5. An example of unbounded growth L

In this section we exhibit an example of an interior trajectory of primal solutions $x(\mu)$, parameterized by μ, along which Theorem 1 does not hold. Unlike on the central path some components of the solutions $x(\mu)$ along this trajectory converge to their optimal values at a slower rate than the rate of convergence to the boundary, which is μ, as will be shown below. Hence the property $x(\mu) = x^* + O(\mu)$ is no longer valid and the theory of the previous section fails.

Let us consider the following 3×3 constraint matrix

$$A = \begin{bmatrix} 1 & -1 & 1 \\ 1 & 1 & 1 \\ 1 & -1 & -1 \end{bmatrix}$$

and an (infeasible) interior trajectory

$$x(\mu) = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} + \begin{pmatrix} 3\mu \\ \sqrt{\mu} \\ 0 \end{pmatrix}$$

to an optimal solution $x^* = (1, 0, -1)^T$. Clearly, as $\mu \to 0$, $x(\mu) \to x^*$, but $x(\mu)$ does not closely follow the central path. Rather, the trajectory $x(\mu)$ becomes tangent to the boundary of the cone at x^*. This follows from the fact that

$$x_0 - \|\bar{x}\| = \Theta(\mu),$$

which means that the overall convergence to the boundary is occurring at a rate μ, while one of the variables is converging to its limit at a slower rate of $\sqrt{\mu}$.

Now, let us estimate the size of the elements in the normal equations.

$$Ax = \begin{pmatrix} \Theta(\sqrt{\mu}) \\ \Theta(\sqrt{\mu}) \\ \Theta(1) \end{pmatrix}, \quad \gamma^2(x) = \Theta(\mu)$$

and

$$AFA^T = \frac{\gamma^2(x)}{2} A \begin{bmatrix} -1 & & \\ & 1 & \\ & & 1 \end{bmatrix} A^T + Axx^T A^T = \begin{bmatrix} \Theta(\mu) & \Theta(\mu) & \Theta(\sqrt{\mu}) \\ \Theta(\mu) & \Theta(\mu) & \Theta(\sqrt{\mu}) \\ \Theta(\sqrt{\mu}) & \Theta(\sqrt{\mu}) & \Theta(1) \end{bmatrix}.$$

The sizes of the elements of the Cholesky factorization $AFA^T = LDL^T$ are:

$$L = \begin{bmatrix} 1 & & \\ \Theta(1) & 1 & \\ \Theta(1/\sqrt{\mu}) & \Theta(1/\sqrt{\mu}) & 1 \end{bmatrix}, \quad D = \begin{bmatrix} \Theta(\mu) & & \\ & \Theta(\mu^2) & \\ & & \Theta(1) \end{bmatrix}.$$

Thus some elements of L approach infinity as $\mu \to 0$.

To observe which part of our proofs fail in this case notice that

$$Q = \begin{bmatrix} \Theta(1) & 0 & \Theta(1) \\ \Theta(\sqrt{\mu}) & \Theta(1) & \Theta(\sqrt{\mu}) \\ \Theta(1) & \Theta(\sqrt{\mu}) & \Theta(1) \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \Theta(1) & & \\ & \Theta(\mu) & \\ & & \Theta(\mu^2) \end{bmatrix},$$

and for this matrix Q Lemma 4 does not hold. Hence when using a partition as in the proof of Theorem 1

$$AFA^T = AQ\Lambda Q^T A^T = AQ_1\Lambda_1 Q_1^T A^T + AQ_\mu \Lambda_\mu Q_\mu^T A^T,$$

we obtain

$$m_{11} = a_1^T F a_1 = a_1^T Q_1 \Lambda_1 Q_1 a_1 + a_1^T Q_\mu \Lambda_\mu Q_\mu^T a_1,$$

$$\Lambda_1 = \Theta(1),\ a_1^T Q_1 = \Theta(\sqrt{\mu}), m_{11} = \Theta(\mu),$$

$$m_{31} = a_3^T Q_1 \Lambda_1 Q_1^T a_1 + \ldots = \Theta(\sqrt{\mu}).$$

We conclude from this that for good numerical performance it may be important that an interior point method for SOCP closely follows the central trajectory. Sturm [27] also observed this phenomena for SDP problems and gave an example of a similar nature of the infinite growth of the elements of the Cholesky factors in the SDP framework.

It is interesting to note that a bad example can be created only by violating the centrality condition within a type 3 cone; that is, variables corresponding to the same cone must have different rates of convergence to optimality. If, however, within each cone the convergence rate to the optimal solution is the same for all variables, but these rates differ *between* different cones, then our theory still applies. Theorem 1 can be extended to this situation, by using techniques described in [11] for the LP case, where no assumptions on the convergence rates of individual LP variables are made.

6. Computational Results

In this section we present some computational results that support the claims of the previous sections. Our test set consists of three randomly generated problems and problems from the DIMACS challenge set [23] that contain SOCP cones. We used the state-of-the-art Matlab code SeDuMi 1.04 (*beta* version) [26] to solve these problems. SeDuMi handles large second-order cones using the product-form Cholesky factorization based on the rank-1 Fletcher-Powell method presented in Section 3. We performed our experiments on an IBM RISC 6000 workstation with 120MB memory.

First we would like to demonstrate that the elements of the Cholesky factorization matrix do not grow without bound when $\mu \to 0$. In Table 3, for each

Table 3. Size of elements of L

problem name	conic structure lin | quad | sdp	nonzeros in A	scaling of A	# of $l_{ij} > 10^2$	# of $l_{ij} > 10^3$	relative dual. gap	D_{max}/D_{min}
rand1000q	| 1000 |	2996	4.5e-01	0	0	1.0e-15	5.9e+00
rand1000_500	400 | 1000,500,5,3,3 |	10480	1.3e-01	0	0	1.0e-13	2.4e+11
rand500x3q	400 | 500(3) |	10430	1.4e-01	10	9	1.0e-14	5.7e+12
sched50_50	2501 |2475 |	27985	2.7e-07	9	0	1.0-11	3.0e+22
sched100_50	5002 | 4742 |	60288	2.1e-07	288	3	1.0e-4	2.2e+14
sched100_100	10002 | 8236 |	114899	3.0e-09	496	34	1.0e-5	2.2e+17
sched200_100	20002 | 17885 |	280500	3.4e-08	407	2	1.0e-4	1.8e+14
nb	4 | 793(3) |	192439	2.2e-17	0	0	1.0e-11	4.1e+12
nb_L2	4 | 839(3) |	404397	6.9e-18	0	0	1.0e-9	3.5e+11
nql60	14402 | 3600(3) |	107639	8.3e-03	0	0	1.0e-10	2.3e+17
filter48_socp	931 | 49 | 48	47234	6.8e-19	151	6	1.0e-7	Inf
filtinf1	945 | 49 | 49	48794	5.0e-07	820	209	1.0e-7	Inf
qssp6	144 | 91(3) |	2751	7.2e-02	187	6	1.0e-9	1.4e+12
qssp9	314 |190(3) |	6438	7.2e-02	352	41	1.0e-10	4.2e+14

problem we list the number and the dimensions of the cones (e.g., 4 | 793(3) | indicates that the problem has 4 linear variables and 793 three-dimensional SOCP cones and no SDP variables) the number of nonzeros in A, the ratio between the smallest and the largest (in absolute value) nonzero elements of A and the following information from the last SeDuMi iteration: the number of elements of L that were greater than 100, the number of elements of L that were greater than $1,000$, the relative duality gap and the ratio between the largest and the smallest elements of the diagonal matrix D of LDL^T. This last number provides an estimate of the condition number of LDL^T to contrast it with the size of the elements of L ("Inf" in the last column indicates that the value was listed as **Inf** in Matlab).

For all of the problems tested, none of the elements of L ever exceeded $10,000$ in absolute value. For most of the problems the number of elements exceeding $1,000$ was quite small, often zero. The only exception was **filtinf1**, which contained SDP constraints, and therefore our analysis is not applicable (at least not without major revision). Notice that even though in most of the problems in the list the matrix A is badly scaled, the size of elements of L still remains reasonable. It seems that the bad scaling of A does not cause the elements of L to blow up, as long as the elements of A form "clusters" according the their order of magnitude rather than spanning the whole spectrum of magnitudes. As an example of "bad behavior" we generated another instance of problem **rand1000_500** allowing the order of magnitude of the elements of A to vary uniformly between the 1 and -7. In this case some of the elements of L exceeded 10^4, whereas for the well-scaled case, in which the order of the magnitude of the elements of A varied between 1 and -1, all elements of L were less than 100.

Problems **sched** were not solved as accurately as the other problems in the list. These problems are those that are marked "scaled" in the test set. In spite of this the matrix A was still badly scaled. A full explanation of the numerical instability of these problems is beyond the issue of using PFCF, since other

Table 4. Time with/without PFCF

problem name	conic structure lin | quad | sdp	m	time full CF	time PFCF
filtinf1	945 | 49 | 49	983	205.96	7.81
filter48_socp	931 | 49 | 48	969	116.27	8.72
rand1000q	| 1000 |	600	5.87	1.85
rand1000_500..	400 | 1000,500,5,3,3 |	1100	62.02	33.33
sched50_50	2501 | 2475 |	2526	-	12.64
sched100_50	5002 | 4742 |	4843	-	45.17
sched100_100	10002 | 8236 |	8337	-	91.11
sched200_100	20002 | 17885 |	18086	-	297

implementations based on the normal equations approach also suffer a loss of accuracy on these problems [29].

Finally, to demonstrate the benefit of using the PFCF approach, we solved all problems in Table 4 that contain at least one large second-order cone with SeDuMi with and without using the product-form factorization. In Table 4 we list the results of these runs. The column titled **full CF** lists solution times (in seconds) when standard Cholesky factorization was applied to the full matrix. The column titled **PFCF** lists the solution times obtained when the PFCF was used. We also list the dimensions and conic structure of the problems.

One can easily see that a PFCF approach yielded a significant improvement on the **filter** problems. These problems contain dense LP columns that were also removed. On random problems, although there was less improvement, it was still substantial. Since random problems typically have a lot of fill-in due to the randomness of the sparsity structure, these improvement factors can be viewed as "worst case" improvements. Finally for the **sched** problems, Matlab ran out of memory when trying to store the almost completely dense matrix AFA^T. Clearly, even if enough memory is available, solving **sched** without treating large second-order cones in a special way would be prohibitively expensive.

We do not present a computational comparison of the PFCF method to the Sherman-Morrison-Woodbury approach for dealing with low-rank dense blocks, since all of our computations were done using SeDuMi, which does not implement the latter approach. From our earlier work on LP [11], we know that the Sherman-Morrison-Woodbury approach is approximately twice as fast as the PFCF method, but it was not implemented in SeDuMi because of the numerical instability that we mentioned in the introduction.

Acknowledgements. We are grateful to the anonymous referees for their helpful suggestions on earlier versions of the manuscript.

References

1. F. Alizadeh and D. Goldfarb, "Second-order cone programming", *Mathematical Programming*, **95(1)**, (2003) pp. 3-51.

2. F. Alizadeh and S. Schmieta, "Optimization with semidefinite, quadratic and linear constraints", Technical Report, RUTCOR, Rutgers University, NJ, 1997.
3. E. Andersen, C. Roos, and T. Terlaky, "On implementing a primal-dual interior-point method for conic quadratic optimization", Helsinki School of Economics and Business Administration, Working paper W-274 (2000).
4. K. D. Andersen, "A modified Schur complement method for handling dense columns in interior point methods for linear programming", *ACM Transactions on Mathematical Software*, **22(3)**, (1996), pp. 348-356.
5. J. M. Bennett, "Triangular factors of modified matrices", *Numerisches Mathematik* **7**, (1965), PP. 217-221.
6. I. C. Choi, C. L. Monma, and D. F. Shanno, "Further development of primal-dual interior point methods", *ORSA J. on Computing*, **2(4)**, (1990), pp. 304-311.
7. R. Fletcher and M. J. D. Powell, "On the Modification of LDL^T factorization", *Mathematics of Computation*, **28(128)**, (1974) pp. 1067-1087.
8. W.M. Gentleman, "Least squares computations by Givens transformations without square roots", *J. Inst. Math. Appl.*, **12**, (1973) pp. 329-336.
9. Ph. E. Gill, W. Murray, and M. A. Saunders, "Methods for computing and modifying the LDV factors of a matrix", *Mathematics of Computation*, **29**, (1975), pp. 1051-10.
10. D. Goldfarb, "The conic simplex method", CORC Report, IEOR Dept., Columbia University, New York, NY, 2002.
11. D. Goldfarb and K. Scheinberg, "A product-form Cholesky factorization method for handling dense columns in interior point methods for linear programming", *Mathematical Programming*, **99**, (2004), pp. 1-34.
12. G. H. Golub and C. F. Van Loan, *Matrix Computations*, third edition, The John Hopkins University Press, Baltimore, 1996.
13. N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM publications, Philadelphia, 1996.
14. K. A. McShane, C. L. Monma, and D. F. Shanno, "An implementation of a primal-dual method for linear programming", *ORSA J. on Computing*, **1(2)**, (1989) pp. 70-83.
15. C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Edgewood Cliffs, NJ, 1974.
16. M.S. Lobo, L. Vandenberghe, S. Boyd and H. Lebret, "Applications of second order cone programming", *Linear Algebra Appl.*, **284**, (1998), pp. 193-228.
17. Z.-Q. Luo, J. Sturm, and S. Zhang "Superlinear convergence of a symmetric primal-dual path following algorithms for semidefinite programming", *SIAM J. on Optimization*, **8(1)**, (1998), pp. 59-81.
18. R.D.C. Monteiro, "Primal-dual path-following algorithms for semidefinite programming", *SIAM J. Optimization*, **7**, (1997) pp. 663-678.
19. A. S. Nemirovskii and K. Scheinberg "Extension of Karmarkar's algorithm to convex quadratically constrained quadratic problems", *Mathematical Programming* **72**, (1996) pp. 273-289.
20. Yu. E. Nesterov and A. S. Nemirovskii, *Interior point Methods in Convex Programming: Theory and Application*, Society of Industrial and Applied Mathematics, Philadelphia, 1994.
21. Yu. E. Nesterov and M. J. Todd, "Self-Scaled Barriers and Interior-Point Methods for Convex Programming", *Mathematics of Operations Research* **22**, (1997) pp. 1-42.
22. Yu. E. Nesterov and M. J. Todd, "Primal-Dual Interior-Point Methods for Self-Scaled Cones", *SIAM Journal on Optimization* **8**, (1998) pp. 324-364.
23. G. Pataki and S. Schmieta, "http://dimacs.rutgers.edu/Challenges/Seventh/Instances", 1999.
24. K. Scheinberg and S. Wright, "Notes on modified Cholesky factorizations and Schur complements for linear systems from interior point methods", working paper, 1998.
25. S.H. Schmieta and F. Alizadeh, "Associative and Jordan algebras, and polynomial time interior-point algorithms for symmetric cones", *Math. of O.R.* **26(3)**, (2001) pp. 543-564.
26. J.F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones", *Optimization Methods and Software* **11-12**, (1999) pp. 625-653.
27. J. F. Sturm, "Avoiding numerical cancellation in the interior point method for solving semidefinite programs", *Mathematical Programming, Series B* **95(2)** pp. 219–247, 2003.
28. T. Tsuchiya, "A Polynomial Primal-Dual Path-Following Algorithm for Second-Order Cone Programming", Technical Report Research Memorandum No. 649, The Institute of Statistical Mathematics, Tokyo, Japan, 1997.

29. R. H. Tütüncü, K. C. Toh and M. J. Todd, "Solving semidefinite-quadratic-linear programs using SDPT3", *Mathematical Programming, Series B* **95(2)**, pp. 189–217, 2003.
30. S. J. Wright, "Modified Cholesky factorizations in interior point algorithms for linear programming", *SIAM J. on Optimization* **9(4)**, (1999), pp. 1159-1191.
31. Y. Zhang, "On extending primal-dual interior-point algorithms from linear programming to semidefinite programming", *SIAM J. Optimization*, **8**, (1998) pp. 356-386.