Math. Program., Ser. B 95: 249–277 (2003)

Digital Object Identifier (DOI) 10.1007/s10107-002-0349-3

E.D. Andersen · C. Roos · T. Terlaky

On implementing a primal-dual interior-point method for conic quadratic optimization

Received: November 18, 2000 / Accepted: January 18, 2001
Published online: September 27, 2002 –

Abstract. Based on the work of the Nesterov and Todd on self-scaled cones an implementation of a primal-dual interior-point method for solving large-scale sparse conic quadratic optimization problems is presented. The main features of the implementation are it is based on a homogeneous and self-dual model, it handles rotated quadratic cones directly, it employs a Mehrotra type predictor-corrector extension and sparse linear algebra to improve the computational efficiency. Finally, the implementation exploits fixed variables which naturally occurs in many conic quadratic optimization problems. This is a novel feature for our implementation. Computational results are also presented to document that the implementation can solve very large problems robustly and efficiently.

Key words. conic optimization – interior-point methods – large-scale implementation

1. Introduction

Conic quadratic optimization is the problem of minimizing a linear objective function subject to the intersection of an affine set and the direct product of quadratic cones of the form

$$\left\{ x : x_1^2 \geq \sum_{j=2}^{n} x_j^2, \; x_1 \geq 0 \right\}. \tag{1}$$

The quadratic cone is also known as the second-order, the Lorentz, or the ice-cream cone.

Many optimization problems can be expressed in this form. Some examples are linear, convex quadratic, and convex quadratically constrained quadratic optimization. Other examples are the problem of minimizing the sum of norms and robust linear optimization. Various applications of conic quadratic optimization are presented in [11, 17].

Over the past 15 years there has been extensive research into interior-point methods for linear optimization. One result of this research is the development of a primal-dual interior-point algorithm [16, 20] which is highly efficient both in theory and in practice [7, 18]. Therefore, several authors have studied how to generalize this algorithm to other

E.D. Andersen: MOSEK APS, Fruebjergvej 3 Box 16, 2100 Copenhagen O, Denmark, e-mail: e.d.andersen@mosek.com

C. Roos: TU Delft, Mekelweg 4, 2628 CD Delft, The Netherlands, e-mail: c.roos@its.tudelft.nl

T. Terlaky: McMaster University, Department of Computing and Software, Hamilton, Ontario, Canada, L8S 4L7. e-mail: terlaky@mcmaster.ca

problems. An important work in this direction is the paper of Nesterov and Todd [23] which shows that the primal-dual algorithm maintains its theoretical efficiency when the nonnegativity constraints are replaced by a convex cone as long as the cone is homogeneous and self-dual, or in the terminology of Nesterov and Todd, a self-scaled cone. It has subsequently been pointed out by Güler [15] that the only interesting cones having this property are direct products of R_+, the quadratic cone, and the cone of positive semi-definite matrices.

In the present work we will mainly focus on conic quadratic optimization and an algorithm for this class of problems.

Several authors have already studied algorithms for conic quadratic optimization. In particular Tuschiya [31] and Monteiro and Tuschiya [21] have studied the complexity of different variants of the primal-dual algorithm. Schmieta and Alizadeh [27] have shown that many of the polynomial algorithms developed for semi-definite optimization immediately can be translated to polynomial algorithms for conic quadratic optimization.

Andersen [9] and Alizadeh and Schmieta [1] discuss implementations of algorithms for conic quadratic optimization. Although they present good computational results, the implemented algorithms have unknown complexity and cannot deal with primal or dual infeasible problems.

Sturm [29] reports that his code SeDuMi can solve conic quadratic and semi-definite optimization problems. Although as described in [28, 30] the implementation is based on the work of Nesterov and Todd, then only limited information is provided about how the code deals with the conic quadratic case.

The purpose of this paper is to present an implementation of a primal-dual interior-point algorithm for conic quadratic optimization which employs the best known algorithm (theoretically), which can handle large sparse problems, is robust and handles primal or dual infeasible problems in a theoretically satisfactory way.

The outline of the paper is as follows. First we review the necessary duality theory for conic optimization and introduce the so-called homogeneous and self-dual model. Next we develop an algorithm based on the work of Nesterov and Todd for the solution of the homogeneous model. After presenting the algorithm we discuss efficient solution of the Newton equation system which has to be solved at every iteration of the algorithm. Finally, we discuss our implementation and present our numerical results.

2. Conic Optimization

2.1. A motivating example

First a motivating example for the subsequent work is presented.

It is well-known that the convex quadratic constraint

$$\frac{1}{2}||x||^2 + ax \leq b$$

can be reformulated as

$$\begin{aligned} ax + z &= b, \\ y &= 1, \\ ||x||^2 &\leq 2zy, \; z, y \geq 0 \end{aligned} \tag{2}$$

which is the intersection of two affine constraints and a quadratic cone. Next, using the definitions

$$z := \frac{u+v}{\sqrt{2}} \text{ and } y := \frac{u-v}{\sqrt{2}}$$

formulation (2) is equivalent to

$$\begin{aligned} ax + \tfrac{u+v}{\sqrt{2}} &= b, \\ u - v &= \sqrt{2}, \\ ||x||^2 + v^2 &\leq u^2, \quad u \geq 0, \end{aligned}$$

that is also the intersection of two affine constraints and a quadratic cone.

This example presents the two most frequently occurring quadratic cones

$$\{(z, x) : \ ||x|| \leq z\} \text{ and } \{(z, y, x) : \ ||x||^2 \leq 2zy, \ z, y \geq 0\}.$$

Furthermore, observe that fixed variables occur naturally in the formulation e.g. (2). This observation will be exploited in Section 5.1.

2.2. *Duality*

In general a conic optimization problem can be expressed in the form

$$\begin{array}{lll} (P) \text{ minimize} & c^T x & \\ \quad\ \text{subject to} & Ax & = b, \\ & x \in K & \end{array} \tag{3}$$

where K is assumed to be a pointed closed convex cone. Moreover, we assume that $A \in R^{m \times n}$ and all other quantities have conforming dimensions. For convenience and without loss of generality we will assume that A is of full row rank. A primal solution x to (P) is said to be feasible if it satisfies all the constraints of (P). Problem (P) is feasible if it has at least one feasible solution. Otherwise the problem is infeasible. Problem (P) is said to be strictly feasible if (P) has a feasible solution such that $x \in \text{int}(K)$, where $\text{int}(K)$ denotes the interior of K.

Let

$$K_* := \{s : \ s^T x \geq 0, \ \forall x \in K\} \tag{4}$$

be the dual cone, then the dual problem corresponding to (P) is given by

$$\begin{array}{ll} (D) \text{ maximize} & b^T y \\ \quad\ \text{subject to} & A^T y + s = c, \\ & s \in K_*. \end{array} \tag{5}$$

A dual solution (y, s) is said to be feasible if it satisfies all the constraints of the dual problem. The dual problem (D) is feasible if it has at least one feasible solution. Moreover, (D) is strictly feasible if a dual solution (y, s) exists such that $s \in \text{int}(K_*)$.

The difference $c^T x - b^T y$ is called the duality gap whereas $x^T s$ is called the complementarity gap. If $x \in K$ and $s \in K_*$, then x and s are said to be complementary if the corresponding complementarity gap is zero.

If (x, y, s) is a primal-dual feasible solution, then *weak duality* holds i.e.

$$c^T x - b^T y = x^T s \geq 0.$$

Strong duality holds, i.e., there exist a primal and a dual feasible solution having zero duality gap if both (P) and (D) are strictly feasible. If

$$\exists (y, s) : \; s \in K_*, \; A^T y + s = 0, \; b^T y^* > 0 \tag{6}$$

then (P) is infeasible. Furthermore, if

$$\exists x : \; x \in K, \; Ax = 0, \; c^T x < 0 \tag{7}$$

then (D) is infeasible.

For a detailed discussion of duality theory in the conic case we refer the reader to [11, 13].

2.3. A homogeneous model

The primal-dual algorithm for linear optimization suggested in [16, 20] and generalized by Nesterov and Todd [23] does not handle primal or dual infeasible problems very well. Indeed, one assumption for the derivation of the algorithm is that both the primal and dual problems have strictly feasible solutions.

Therefore, following [12, 24, 28, 30] we employ a generalization of the Goldman-Tucker homogeneous model (or self-dual embedding) for linear optimization to the case of conic optimization. The so-called simplified version is

$$\begin{aligned} Ax - b\tau &= 0, \\ A^T y + s - c\tau &= 0, \\ -c^T x + b^T y - \kappa &= 0, \\ (x; \tau) \in \bar{K}, (s; \kappa) &\in \bar{K}_*. \end{aligned} \tag{8}$$

Here we use the notation that

$$\bar{K} := K \times R_+ \quad \text{and} \quad \bar{K}_* := K_* \times R_+.$$

Subsequently we call a solution to (8) complementary if the complementarity gap

$$x^T s + \tau\kappa$$

is zero.

Lemma 1. *Let $(x^*, \tau^*, y^*, s^*, \kappa^*)$ be any feasible solution to (8), then*

i)

$$(x^*)^T s^* + \tau^* \kappa^* = 0.$$

ii) If $\tau^* > 0$ then $(x^*, y^*, s^*)/\tau^*$ is a primal-dual optimal solution to (P) and (D).

iii) If $\kappa^* > 0$ then at least one of the strict inequalities

$$b^T y^* > 0 \tag{9}$$

and

$$c^T x^* < 0 \tag{10}$$

hold. If the first inequality holds then (P) is infeasible. If the second inequality holds then (D) is infeasible.

Proof. We leave this as an exercise to the reader.

This implies that any solution to the homogeneous model having

$$\tau^* + \kappa^* > 0 \tag{11}$$

is either a scaled optimal solution or a certificate of infeasibility. If no such solution exists, then a tiny perturbation to the problem data exists such that the perturbed problem has a solution satisfying (11) [11]. Hence, the problem is ill-posed. In the case of linear optimization this is never the case. Indeed in this case a so-called strictly complementary solution satisfying (11) and $x^* + s^* > 0$ always exist. However, for example for a primal and dual feasible conic quadratic problem having non-zero duality gap, then (11) cannot be satisfied. See [11] for a concrete example.

Hence, instead of solving (P) and (D) directly we will suggest an algorithm that solves (8).

3. Conic quadratic optimization

In the remaining part of this work we restrict our attention to cones that can be formulated as the product of R_+ and the quadratic cones. To be specific, we will work with the following three cones:

Definition 1. *R_+:*

$$R_+ := \{x \in R : \ x \geq 0\}. \tag{12}$$

Quadratic cone:

$$K_q := \{x \in R^n : \ x_1^2 \geq ||x_{2:n}||^2, \ x_1 \geq 0\}, \tag{13}$$

where $x_{2:n} := (x_2, \ldots x_n)^T$.

Rotated quadratic cone:

$$K_r := \{x \in R^n : \ 2x_1 x_2 \geq ||x_{3:n}||^2, \ x_1, x_2 \geq 0\}. \tag{14}$$

These three cones are homogeneous and self-dual, see Definition 4. Without loss of generality it can be assumed that

$$K = K^1 \times \ldots \times K^k$$

i.e. cone K is the direct product of several individual cones each one of the type (12), (13), or (14), respectively. Furthermore, let x be partitioned according to the cones i.e.

$$x = \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^k \end{bmatrix} \quad \text{and} \quad x^i \in K^i \subseteq R^{n^i}.$$

Associated with each cone are two matrices

$$Q^i, T^i \in R^{n^i \times n^i}$$

that are defined in Definition 2.

Definition 2. *i.) If K^i is R_+, then*

$$T^i := 1 \quad and \quad Q^i = 1. \tag{15}$$

ii.) If K^i is the quadratic cone, then

$$T^i := I_{n^i} \quad and \quad Q^i := diag(1, -1, \ldots, -1). \tag{16}$$

iii.) If K^i is the rotated quadratic cone, then

$$T^i := \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \cdots 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \cdots 0 \\ 0 & 0 & 1 \cdots 0 \\ \vdots & \vdots & \vdots \ddots \vdots \\ 0 & 0 & 0 \cdots 1 \end{bmatrix} \tag{17}$$

and

$$Q^i := \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 \end{bmatrix}. \tag{18}$$

It is an easy exercise to verify that each Q^i and T^i are orthogonal. Hence

$$Q^i Q^i = I \quad \text{and} \quad T^i T^i = I.$$

The definition of the Q matrices allows an alternative way of stating the quadratic cone because assume K^i is the quadratic cone then

$$K^i = \{x^i \in R^{n^i} : (x^i)^T Q^i x^i \geq 0, \ x_1^i \geq 0\}$$

and if K^i is a rotated quadratic cone, then

$$K^i = \{x^i \in R^{n^i} : (x^i)^T Q^i x^i \geq 0, \ x_1^i, x_2^i \geq 0\}.$$

If the ith cone is a rotated quadratic cone, then

$$T^i x^i \in K_r \ \Leftrightarrow \ x^i \in K_q.$$

This demonstrates that the rotated quadratic cone is identical to the quadratic cone under a linear transformation.

For algorithmic purposes the complementarity conditions between the primal and dual solutions are needed. Using the notation that if v is a vector, then capital V denotes the related "arrow head" matrix

$$V := \mathrm{mat}\,(v) = \begin{bmatrix} v_1 & v_{2:n}^T \\ v_{2:n} & v_1 I \end{bmatrix}.$$

The complementarity conditions can now be stated compactly as presented in Lemma 2.

Lemma 2. *Let $x, s \in K$ then x and s are complementary, i.e. $x^T s = 0$, if and only if*

$$X^i S^i e^i = S^i X^i e^i = 0, \quad i = 1, \ldots, k, \tag{19}$$

where $X^i := mat\left(T^i x^i\right)$, $S^i := mat\left(T^i s^i\right)$ and $e_i = (1, 0, \ldots, 0)^T \in R^{n_i}$.

Proof. See the Appendix.

Subsequently let X and S denote two block diagonal matrices having X^i and S^i along the diagonal i.e.,

$$X := \mathrm{diag}(X^1, \ldots, X^k) \quad \text{and} \quad S := \mathrm{diag}(S^1, \ldots, S^k).$$

Given $v \in \mathrm{int}(K)$, then it is easy to verify the following useful formula

$$\mathrm{mat}\,(v)^{-1} = V^{-1} = \frac{1}{v_1^2 - ||v_{2:n}||^2} \begin{bmatrix} v_1 & -v_{2:n}^T \\ -v_{2:n} & \left(v_1 - \frac{||v_{2:n}||^2}{v_1}\right) I + \frac{v_{2:n} v_{2:n}^T}{v_1} \end{bmatrix}.$$

3.1. The central Path

The guiding principle in primal-dual interior-point algorithms is to follow the so-called central path towards an optimal solution. The central path is a smooth curve connecting an initial interior point and a complementary solution. Formally, let an initial point $(x^{(0)}, \tau^{(0)}, y^{(0)}, s^{(0)}, \kappa^{(0)})$ be given such that

$$(x^{(0)}; \tau^{(0)}), (s^{(0)}; \kappa^{(0)}) \in \mathrm{int}(\bar{K})$$

then the set of nonlinear equations

$$\begin{aligned} Ax - b\tau &= \gamma(Ax^{(0)} - b\tau^{(0)}), \\ A^T y + s - c\tau &= \gamma(A^T y^{(0)} + s^{(0)} - c\tau^{(0)}), \\ -c^T x + b^T y - \kappa &= \gamma(-c^T x^{(0)} + b^T y^{(0)} - \kappa^{(0)}), \\ XSe &= \gamma \mu^{(0)} e, \\ \tau\kappa &= \gamma \mu^{(0)}, \end{aligned} \tag{20}$$

defines the central path of the homogeneous model parameterized by $\gamma \in [0, 1]$. Here $\mu^{(0)}$ is given by the expression

$$\mu^{(0)} := \frac{(x^{(0)})^T s^{(0)} + \tau^{(0)} \kappa^{(0)}}{k + 1}$$

and e by the expression

$$e := \begin{bmatrix} e^1 \\ \vdots \\ e^k \end{bmatrix}.$$

The first three blocks of equations in (20) are feasibility equations whereas the last two blocks of equations are the relaxed complementarity conditions.

In general it is not possible to compute a point on the central path exactly. However, using Newton's method a point in a neighborhood of the central path can be computed efficiently. Nesterov and Todd [25] proves that

$$(x^T s + \tau\kappa)\left(\sum_{i=1}^{k} \frac{(x^i)^T s^i}{(x^i)^T Q^i (x^i)(s^i)^T Q^i (s^i)} + \frac{1}{\tau\kappa}\right) \geq (k+1)^2 \tag{21}$$

for all $(x; \tau), (s; \kappa) \in \operatorname{int}(\bar{K})$. The inequality (21) holds as equality if and only if the point (x, τ, s, κ) is on the central path. Therefore, for some $\beta \in [0, 1]$ the set of points satisfying

$$\begin{aligned} \tfrac{(x^i)^T Q^i (x^i)(s^i)^T Q^i (s^i)}{(x^i)^T s^i} &\geq \beta \tfrac{(x^T s + \tau\kappa)}{k+1}, \; i = 1, \ldots, k, \\ \tau\kappa &\geq \beta \tfrac{(x^T s + \tau\kappa)}{k+1} \end{aligned} \tag{22}$$

defines a neighborhood. In the case all cones are linear this neighbor corresponds to the widely used one-sided ∞-norm neighborhood [33, p. 9]. Moreover, Tuncel [32] demonstrates that the equality (21) can be generalized, so it holds for nonsymmetric cones as long as an appropriate barrier functions are known for the primal and dual cones.

Formally, we define our neighborhood as follows

$$\mathcal{N}(\beta) := \left\{ (x; \tau), (s; \kappa) \in \operatorname{int}(\bar{K}) : \begin{aligned} \tfrac{(x^i)^T Q^i (x^i)(s^i)^T Q^i (s^i)}{(x^i)^T s^i} &\geq \beta\mu, \; \forall i, \\ \tau\kappa &\geq \beta\mu \end{aligned} \right\},$$

where

$$\mu := \frac{x^T s + \tau\kappa}{k+1}$$

and $\beta \in [0, 1]$. Clearly, the size of $\mathcal{N}(\beta)$ increases with a decrease in β. Moreover, $\mathcal{N}(1)$ coincides with the central path.

3.2. Scaling

For later use we need the definition of a scaling matrix.

Definition 3. *$W^i \in R^{n^i \times n^i}$ is a scaling matrix if it satisfies the conditions*

$$\begin{aligned} W^i &\succ 0, \\ W^i Q^i W^i &= Q^i, \end{aligned}$$

where $W^i \succ 0$ means W^i is symmetric and positive definite.

A scaled point $\bar{x}, \bar{s}$ is obtained by the transformation

$$\bar{x} := \Theta W x \quad \text{and} \quad \bar{s} := (\Theta W)^{-1} s,$$

where

$$W := \begin{bmatrix} W^1 & 0 & \cdots & 0 \\ 0 & W^2 & \vdots & \vdots \\ \vdots & \cdots & \ddots & 0 \\ 0 & \cdots & 0 & W^k \end{bmatrix}$$

and

$$\Theta = \operatorname{diag}(\theta_1 1_{n^1}; \ldots ; \theta_k 1_{n^k}).$$

1_{n^i} is the vector of all ones having the length n^i and $\theta \in R^k_+$. Hence, W is a block diagonal matrix having the W^is along the diagonal and Θ is a diagonal matrix.

It is shown in Lemma 3 that scaling does not change anything. For example, if the original point is in the interior of the cone K, the scaled point is in the interior too. Similarly, if the original point belongs to a certain neighborhood, then the scaled point belongs to the same neighborhood.

Lemma 3. *i)* $(x^i)^T s^i = (\bar{x}^i)^T \bar{s}^i$.
ii) $\theta_i^2 (x^i)^T Q^i x^i = (\bar{x}^i)^T Q^i \bar{x}^i$.
iii) $\theta_i^{-2} (s^i)^T Q^i s^i = (\bar{s}^i)^T Q^i \bar{s}^i$.
iv) $x \in K \Leftrightarrow \bar{x} \in K$ *and* $x \in int(K) \Leftrightarrow \bar{x} \in int(K)$.
v) Given a $\beta \in (0, 1)$ *then*

$$(x, \tau, s, \kappa) \in \mathcal{N}(\beta) \Rightarrow (\bar{x}, \tau, \bar{s}, \kappa) \in \mathcal{N}(\beta).$$

Proof. See the Appendix.

4. The search direction

The main algorithmic idea in a primal-dual interior-point algorithm is to trace the central path loosely.

Indeed given an initial interior solution

$$(x^{(0)}, \tau^{(0)}, y^{(0)}, s^{(0)}, \kappa^{(0)})$$

in the neighborhood of the central path and a fixed γ then we will use a search direction

$$(d_x, d_\tau, d_y, d_s, d_\kappa)$$

from the Monteiro-Zhang family of search directions that is defined by the linear equation system

$$\begin{aligned} Ad_x - bd_\tau &= (\gamma - 1)(Ax^{(0)} - b\tau^{(0)}), \\ A^T d_y + d_s - cd_\tau &= (\gamma - 1)(A^T y^{(0)} + s^{(0)} - c\tau^{(0)}), \\ -c^T d_x + b^T d_y - d_\kappa &= (\gamma - 1)(-c^T x^{(0)} + b^T y^{(0)} - \kappa), \\ \bar{X}^{(0)} T(\Theta W)^{-1} d_s + \bar{S}^{(0)} T \Theta W d_x &= -\bar{X}^{(0)} \bar{S}^{(0)} e + \gamma \mu^{(0)} e, \\ \tau^{(0)} d_\kappa + \kappa^{(0)} d_\tau &= -\tau^{(0)} \kappa^{(0)} + \gamma \mu^{(0)}, \end{aligned} \tag{23}$$

where

$$T := \begin{bmatrix} T^1 & 0 & \cdots & 0 \\ 0 & T^2 & \vdots & \vdots \\ \vdots & \cdots & \ddots & 0 \\ 0 & \cdots & 0 & T^k \end{bmatrix}.$$

Given A is of full row rank and an appropriate choice of the scaling ΘW it can be shown that the scaled Monteiro-Zhang search direction is uniquely defined.

This search direction corresponds to applying Newton's method to (20) in a scaled space and then scaling the resulting search direction back to the original space. The scaling $\Theta W = I$ corresponds to the unscaled Newton search direction which has the drawback that it is not always well defined and is expensive to compute compared to the choice of the scaling mentioned in sequel.

A new point is obtained by moving in the direction $(d_x, d_\tau, d_y, d_s, d_\kappa)$ as follows

$$\begin{bmatrix} x^{(1)} \\ \tau^{(1)} \\ y^{(1)} \\ s^{(1)} \\ \kappa^{(1)} \end{bmatrix} = \begin{bmatrix} x^{(0)} \\ \tau^{(0)} \\ y^{(0)} \\ s^{(0)} \\ \kappa^{(0)} \end{bmatrix} + \alpha \begin{bmatrix} d_x \\ d_\tau \\ d_y \\ d_s \\ d_\kappa \end{bmatrix} \tag{24}$$

for some step size $\alpha \in [0, 1]$. This is a promising idea because given $\gamma \in [0, 1[$ the new point is closer to being feasible to the homogeneous model and complementarity. This is the significance of Lemma 4.

Lemma 4. *Given (23) and (24) then*

$$\begin{aligned} Ax^{(1)} - b\tau^{(1)} &= (1 - \alpha(1 - \gamma))(Ax^{(0)} - b\tau^{(0)}), \\ A^T y^{(1)} + s^{(1)} - c\tau^{(1)} &= (1 - \alpha(1 - \gamma))(A^T y^{(0)} + s^{(0)} - c\tau^{(0)}), \\ -c^T x^{(1)} + b^T y^{(1)} - \kappa^{(1)} &= (1 - \alpha(1 - \gamma))(-c^T x^{(0)} + b^T y^{(0)} - \kappa^{(0)}), \\ d_x^T d_s + d_\tau d_\kappa &= 0, \\ (x^{(1)})^T s^{(1)} + \tau^{(1)} \kappa^{(1)} &= (1 - \alpha(1 - \gamma))((x^{(0)})^T s^{(0)} + \tau^{(0)} \kappa^{(0)}). \end{aligned} \tag{25}$$

Proof. Follows easily by using elementary linear algebra.

One important issue left open is the choice of the scaling ΘW. Monteiro and Tuschiya [21] has proven polynomial complexity of the resulting algorithm for several different choices of scaling, an appropriate choice of γ and step size α. However, the best results are obtained for the NT scaling suggested in [23]. In the NT scaling ΘW is chosen such that

$$\Theta W x = \bar{x} = \bar{s} = (\Theta W)^{-1} s, \tag{26}$$

holds that is equivalent to require the scaled primal and dual points is identical. Note relation (26) implies

$$s = (W\Theta^2 W)x. \tag{27}$$

In case of NT scaling both Θ and W can be computed cheaply for each of our cones as demonstrated in Lemma 5.

Lemma 5. *Assume that $x^i, s^i \in int(K^i)$ then*

$$\theta_i^2 = \sqrt{\frac{(s^i)^T Q^i s^i}{(x^i)^T Q^i x^i}}. \tag{28}$$

Moreover, if K^i is

i) a positive half-line R_+, then

$$\begin{aligned} W^i &= \tfrac{1}{\theta_i}((X^i)^{-1}S^i)^{\frac{1}{2}} \\ &= 1. \end{aligned}$$

ii) a quadratic cone, then

$$\begin{aligned} W^i &= \begin{bmatrix} w_1^i & \left(w_{2:n^i}^i\right)^T \\ w_{2:n^i}^i & I + \frac{w_{2:n^i}^i \left(w_{2:n^i}^i\right)^T}{1+w_1^i} \end{bmatrix} \\ &= -Q^i + \frac{(e^i + w^i)(e^i + w^i)^T}{1 + (e^i)^T w^i}, \end{aligned} \tag{29}$$

where

$$w^i = \frac{\theta_i^{-1} s^i + \theta_i Q^i x^i}{\sqrt{2}\sqrt{(x^i)^T s^i + \sqrt{(x^i)^T Q^i x^i (s^i)^T Q^i s^i}}}. \tag{30}$$

Furthermore,

$$(W^i)^2 = -Q^i + 2w^i (w^i)^T. \tag{31}$$

iii) a rotated quadratic cone, then

$$W^i = -Q^i + \frac{(T^i e^i + w^i)(T^i e^i + w^i)^T}{1 + (e^i)^T T^i w^i}, \tag{32}$$

where w^i is given by (30). Furthermore,

$$(W^i)^2 = -Q^i + 2w^i (w^i)^T. \tag{33}$$

Proof. In case of the quadratic cone the Lemma is derived in [31], but we prefer to include a proof here for the sake of completeness. See the Appendix for details.

Lemma 6. *Let W^i be given as in Lemma 5 then*

$$(W^i\theta_i^2 W^i)^{-1} = \theta_i^{-2} Q^i (W^i)^2 Q^i.$$

Proof. Using Definition 3 we have that $W^i Q^i W^i = Q^i$ and $Q^i Q^i = I$ which implies $(W^i)^{-1} = Q^i W^i Q^i$ and $(W^i)^{-2} = Q^i (W^i)^2 Q^i$.

One observation which can be made from Lemma 5 and Lemma 6 is that the scaling matrix W can be stored using an n dimensional vector because only the vector w^i has to be stored for each cone. Furthermore, any multiplication with W or W^2 or their inverses can be carried out in $O(n)$ complexity. This is an important fact that should be exploited in an implementation.

4.1. Choice of the step size

After the search direction has been computed then a step size has to be chosen. In general the step size α should be chosen such that

$$(x^{(0)} + \alpha d_x, \tau^{(0)} + \alpha d_\tau, s^{(0)} + \alpha d_s, \kappa^{(0)} + \alpha d_\kappa) \in \mathcal{N}(\beta) \tag{34}$$

where $\beta \in (0, 1)$ is a fixed constant. In our implementation we have chosen a decreasing sequence of α's in the interval [0, 1]. The largest α in the sequence satisfying (34) is chosen as the step size.

4.2. Adapting mehrotra's predictor-corrector method

Several important issues have not been addressed so far. In particular nothing has been stated about the choice of γ. In theoretical work on primal-dual interior-point algorithms γ is usually chosen as a constant close to one but in practice this leads to slow convergence. Therefore, in the linear case Mehrotra [19] suggested a heuristic which chooses γ dynamically depending on how much progress can be made in the pure Newton (affine scaling) direction. Furthermore, Mehrotra suggested using a second-order correction of the search direction which increases the efficiency of the algorithm significantly in practice [18].

In this section we discuss how these two ideas proposed by Mehrotra can be adapted to the primal-dual method based on the Monteiro-Zhang family of search directions.

Mehrotra's predictor-corrector method utilizes the observation that

$$\text{mat}\,(Tx) + \text{mat}\,(Td_x) = \text{mat}\,(T(x + d_x))$$

and hence

$$\begin{aligned}(X + D_x)(S + D_s)e &= \text{mat}\,(T(x + d_x))\,\text{mat}\,(T(s + d_s))\,e \\ &= XSe + SD_xe + XD_se + D_xD_se,\end{aligned}$$

where

$$D_x := \text{mat}\,(Td_x) \quad \text{and} \quad D_s := \text{mat}\,(Td_s)\,.$$

When Newton's method is applied to the perturbed complementarity conditions

$$XS = \gamma\mu^{(0)}e,$$

then the quadratic term

$$D_x D_s e \tag{35}$$

is neglected and the search direction is obtained by solving the resulting system of linear equations. Instead of neglecting the quadratic term Mehrotra suggests to estimate it using the pure Newton direction. Indeed, Mehrotra suggested computing the primal-dual affine scaling direction

$$(d_x^n, d_\tau^n, d_y^n, d_s^n, d_\kappa^n)$$

first which is the unique solution of (23) for $\gamma = 0$. Next this direction is used to estimate the quadratic term as follows

$$D_x D_s e \approx D_x^n D_s^n e \quad \text{and} \quad d_\tau d_\kappa \approx d_\tau^n d_\kappa^n.$$

In the framework of the Monteiro-Zhang family of search directions this implies that the linearized complementarity conditions in (23) are replaced by

$$\begin{aligned} \bar{X}^{(0)}T(\Theta W)^{-1}d_s + \bar{S}^{(0)}T\Theta W d_x &= -\bar{X}^{(0)}\bar{S}^{(0)}e + \gamma\mu^{(0)}e - \bar{D}_x^n\bar{D}_s^n e, \\ \tau^{(0)}d_\kappa + \kappa^{(0)}d_\tau &= -\tau^{(0)}\kappa^{(0)} + \gamma\mu^{(0)} - d_\tau^n d_\kappa^n, \end{aligned}$$

where

$$\bar{D}_x^n := \operatorname{mat}\left(T\Theta W d_x^n\right) \quad \text{and} \quad \bar{D}_s^n := \operatorname{mat}\left(T(\Theta W)^{-1}d_s^n\right).$$

Note that even though the corrector term is included in the right-hand-side, it can be proved that the final search direction satisfies all the properties stated in Lemma 4.

Mehrotra also suggests to use the pure Newton direction for a dynamic choice of γ based on how much progress can be made in the pure Newton direction. Now let $\alpha_n^{\max}$ be the maximum step size to the boundary which can be taken along the pure Newton direction. According to Lemma 4 this implies that the residuals and the complementarity gap are reduced by a factor of

$$1 - \alpha_n^{\max}.$$

Then it seems reasonable to choose γ small if $\alpha_n^{\max}$ is large. The heuristic

$$\gamma = \min(\delta, (1 - \alpha_n^{\max})^2)(1 - \alpha_n^{\max})$$

achieve this, where $\delta \in [0, 1]$ is a fixed constant.

4.3. *Free variables*

Until now we have assumed that no free variables occur in the problem where a free variable is a variable which is not member of quadratic cone and is not bounded below by 0. This is of course not always the case in practice. We will not delve into the details about how we deal with free variables but refer the reader to [2].

5. Computing the search direction

The computationally most expensive part of the primal-dual algorithm is the computation of the search direction which requires the solution of a linear equation system of the form

$$\begin{aligned} Ad_x - bd_\tau &= r^1, \\ A^T d_y + d_s - cd_\tau &= r^2, \\ -c^T d_x + b^T d_y - d_\kappa &= r^3, \\ \bar{X}^{(0)} T(\Theta W)^{-1} d_s + \bar{S}^{(0)} T\Theta W d_x &= r^4, \\ \tau^{(0)} d_\kappa + \kappa^{(0)} d_\tau &= r^5, \end{aligned} \tag{36}$$

where r represents an arbitrary right-hand side.

Before proceeding with the details of the computation of the search direction we recall that any matrix-vector product involving the matrices $\bar{X}^{(0)}$, $\bar{S}^{(0)}$, T, W, Θ, W^2, and Θ^2, or their inverses, can be carried out in $O(n)$ complexity. Hence, these operations are computationally cheap operations and will not be considered further.

Using elementary linear algebra, it can be shown that if

$$\begin{aligned} g^2 &= (A(W\Theta^2 W)^{-1} A^T)^{-1}(b + A(\Theta W)^2 c), \\ g^1 &= -(W\Theta^2 W)^{-1}(c - A^T g^2) \end{aligned} \tag{37}$$

and

$$\begin{aligned} h^2 &= (A(W\Theta^2 W)^{-1} A^T)^{-1}(r^1 + A(\Theta W)^2(r^2 - (\Theta W T\bar{X}^{(0)})^{-1} r^4)), \\ h^1 &= -(W\Theta^2 W)^{-1}(r^2 - \Theta W(\bar{X}^{(0)})^{-1} r^4 - A^T h^2), \end{aligned} \tag{38}$$

then

$$\begin{aligned} d_\tau &= \frac{r^3 - c^T h^1 + b^T h^2}{(\tau^{(0)})^{-1}\kappa^{(0)} + c^T g^1 - b^T g^2}, \\ d_x &= g^1 + h^1 d_\tau, \\ d_y &= g^2 + h^2 d_\tau, \\ d_s &= \Theta W T(\bar{X}^{(0)})^{-1}(r^4 - \bar{S}^{(0)} T\Theta W d_x), \\ d_\kappa &= (\tau^{(0)})^{-1}(r^5 - \kappa^0 d_\tau). \end{aligned} \tag{39}$$

Hence, we have reduced the computation of the search direction to computing

$$(A(\Theta W)^{-2} A^T)^{-1}$$

or equivalently to solve a linear equation system of the form

$$A(W\Theta^2 W)^{-1} A^T z = r. \tag{40}$$

The matrix $A(W\Theta^2 W)^{-1} A^T$ is symmetric and positive definite and as observed in [1] the system (40) can be solved efficiently by using a sparse Cholesky decomposition approach. However, in the case A contains dense columns or some cones contains many variables the Cholesky approach is inefficient due to $A(W\Theta^2 W)^{-1} A^T$ becoming dense [14]. However, the dense columns and big cones can be handled efficiently with the Schur complement approach or a product-form Cholesky approach [14]. The product-form Cholesky approach has superior numerical stability properties but it is computationally more expensive than the Schur complement approach. The method for handling dense columns and big cones we have implemented is based on the modified Schur complement approach as presented in [8].

Further details about how the Cholesky decomposition is computed can be found in [4, 5].

5.1. Exploiting structure in the constraint matrix

In practice most optimization problems have some structure in the constraint matrix which can be exploited to speed up the computations. In our implementation we exploit the following two types of constraint structures.

Upper bound constraints: If a variable x_j which is not member of any quadratic cone has both a lower and an upper bound, then an additional constraint of the form

$$x_j + x_k = u$$

must be introduced where x_k is a slack variable and therefore occurs in only one constraint.

Singleton constraints: As shown in Section 2.1, then constraints of the form

$$x_j = b$$

frequently arise where x_j only occurs in one constraint. Moreover, we will assume that x_j does not belong to a linear cone because such a variable can simply be substituted out of the problem.

Exploitation of the upper bound constraints are trivial and similar to the pure linear optimization case as discussed in for example [6]. Therefore, we will not discuss this case further and the subsequent discussion is limited to the singleton constraints case only.

After a suitable reordering of the variables and the constraints, we may assume A has the form

$$A = \begin{bmatrix} 0 & A_{12} \\ I & 0 \end{bmatrix}$$

where

$$[\, I \;\; 0 \,]$$

corresponds to the set of singleton constraints. This observation can be exploited when computing g and h from (37) and (38). The vectors g and h are both solutions to equations system of the form

$$\begin{bmatrix} -H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} g^1 \\ g^2 \end{bmatrix} = \begin{bmatrix} r^1 \\ r^2 \end{bmatrix}, \tag{41}$$

where r^1 and r^2 represents arbitrary right-hand sides.

Next, assume that the vector g and the right-hand side of the system (41) has been partitioned according to the partitioning of the matrix A and according to the partitioning of

$$H := \begin{bmatrix} -H_{11} & -H_{12} \\ -H_{21} & -H_{22} \end{bmatrix} = -(W\Theta^2 W)^{-1}.$$

This implies that g is given as the solution to a system of the form

$$\begin{bmatrix} -H_{11} & -H_{12} & 0 & I \\ -H_{21} & -H_{22} & A_{12}^T & 0^T \\ 0 & A_{12} & 0 & 0 \\ I & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} g_1^1 \\ g_2^1 \\ g_1^2 \\ g_2^2 \end{bmatrix} = \begin{bmatrix} r^{11} \\ r^{12} \\ r^{21} \\ r^{22} \end{bmatrix}.$$

This large system can be reduced to the two small systems

$$\begin{bmatrix} -H_{22} & A_{12}^T \\ A_{12} & 0 \end{bmatrix} \begin{bmatrix} g_1^2 \\ g_2^1 \end{bmatrix} = \begin{bmatrix} r^{12} \\ r^{21} \end{bmatrix} + \begin{bmatrix} H_{21} r^{11} \\ 0 \end{bmatrix} \tag{42}$$

and

$$\begin{bmatrix} -H_{11} & I^T \\ I & 0 \end{bmatrix} \begin{bmatrix} g_1^1 \\ g_2^2 \end{bmatrix} = \left(\begin{bmatrix} r^{11} \\ r^{22} \end{bmatrix} - \begin{bmatrix} -H_{12} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} g_1^2 \\ g_2^1 \end{bmatrix} \right) \tag{43}$$

that has to be solved in the order as stated. First, observe that the second system (43) is trivial to solve, whereas system (42) is easily solved if the inverses or appropriate factorizations of the matrices

$$H_{22}$$

and

$$A_{21} H_{22}^{-1} A_{21}^T \tag{44}$$

are known. Next, observe that A_{21} is of full row rank because A is of full row rank. Finally, due to H is positive definite, the matrix H_{22} is positive definite as well, which implies that the matrix (44) is positive definite.

Since the matrix H_{22} is identical to H, except some rows and columns have been removed, then H_{22} is also a block diagonal matrix where each block originate from a block in H. Subsequently we will show that the inverse of each block in H_{22} can be computed efficiently.

In the discussion we will assume that H_{22} consists of one block only. It should be obvious how to extend the discussion to the case of multiple blocks. Any block in H can be written in the form

$$-Q + 2ww^T,$$

where we have dropped the cone subscript i for convenience. Here Q is either of the form (16) or (18) and w is a vector with appropriate dimension. Next we will partition the block to obtain

$$-\begin{bmatrix} Q_{11} & Q_{21} \\ Q_{12} & Q_{22} \end{bmatrix} + 2 \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}^T.$$

After dropping the appropriate rows and columns we assume we are left with the block

$$H_{22} = -Q_{22} + 2w_2 w_2^T,$$

for which we have to compute an inverse of. First assume that Q_{22} is nonsingular. Then by using the Sherman-Morrison-Woodbury formula[1] we obtain

$$(-Q_{22} + 2w_2 w_2^T)^{-1} = -Q_{22}^{-1} - 2\frac{Q_{22}^{-1} w_2 w_2^T Q_{22}^{-1}}{1 - 2w_2^T Q_{22}^{-1} w_2}$$

that is the required explicit representation for the inverse of the H_{22} block. It can be verified that in this case $Q_{22}^{-1} w_2$ can be computed in $O(n)$ complexity.

In most cases Q_{22} is a nonsingular matrix, because it is only singular if the block corresponds to a rotated quadratic cone and either x_1 or x_2 but not both variables are fixed. This implies that in the case Q_{22} is singular then it can be assumed that Q_{22} has the form

$$Q_{22} = \begin{bmatrix} 0 & 0 \\ 0 & -I \end{bmatrix}.$$

Now let

$$w_2 = \begin{bmatrix} \bar{w}_1 \\ \bar{w}_2 \end{bmatrix}$$

where $\bar{w}_1$ is a scalar and it can be verified that $\bar{w}_1 > 0$. It is now easy to verify that

$$-Q_{22} + 2w_2 w_2^T = FF^T$$

where

$$F := \begin{bmatrix} \sqrt{2}\bar{w}_1 & 0 \\ \sqrt{2}\bar{w}_2 & I \end{bmatrix} \quad \text{and} \quad F^{-1} = \begin{bmatrix} \frac{1}{\sqrt{2}\bar{w}_1} & 0 \\ -\frac{\bar{w}_2}{\bar{w}_1} & I \end{bmatrix}.$$

Hence,

$$\begin{aligned}
(-Q_{22} + 2w_2 w_2^T)^{-1} &= (FF^T)^{-1} \\
&= \begin{bmatrix} \frac{1+2||\bar{w}_2||^2}{2\bar{w}_1^2} & -\frac{\bar{w}_2^T}{\bar{w}_1} \\ -\frac{\bar{w}_2}{\bar{w}_1} & I \end{bmatrix} \\
&= \begin{bmatrix} \frac{1+2||\bar{w}_2||^2}{2\bar{w}_1^2} & 0 \\ 0 & I \end{bmatrix} - \frac{1}{\bar{w}_1}\left(\begin{bmatrix} 0 \\ \bar{w}_2 \end{bmatrix} e_1^T + e_1 \begin{bmatrix} 0 \\ \bar{w}_2 \end{bmatrix}^T\right)
\end{aligned}$$

that is the explicit representation of the inverse of the H_{22} block we were looking for.

In summary, instead of computing a factorization of $A(W\Theta^2 W)^{-1}A^T$ it is sufficient to compute a factorization of the potentially much smaller matrix (44) and then just do some additional cheap linear algebra.

[1] If the matrices B and matrix $B + vv^T$ are nonsingular, then $1 + v^T B^{-1} v$ is nonzero and $(B + vv^T)^{-1} = B^{-1} - \frac{B^{-1}vv^T B^{-1}}{1+v^T B^{-1} v}$.

6. Starting and stopping

6.1. Starting point

In our implementation we use the simple starting point:

$$x^{i(0)} = s^{i(0)} = T^i e_1^i,$$

$y^{(0)} = 0$ and $\tau^{(0)} = \kappa^{(0)} = 1$. This choice of starting point implies that

$$(x^{(0)}, \tau^{(0)}, s^{(0)}, \kappa^{(0)}) \in \mathcal{N}(1).$$

6.2. Stopping criteria

An important issue is when to terminate the interior-point algorithm. Obviously the algorithm cannot be terminated before a feasible solution to the homogeneous model has been obtained. Therefore, to measure the infeasibility the following measures

$$\begin{aligned}
\rho_P^{(k)} &:= \tfrac{||Ax^{(k)} - b\tau^{(k)}||_\infty}{\max(1, ||[A, b]||_\infty)}, \\
\rho_D^{(k)} &:= \tfrac{||A^T y^{(k)} + s^{(k)} - c\tau^{(k)}||_\infty}{\max(1, ||[A^T, I, -c]||_\infty)}, \\
\rho_G^{(k)} &:= \tfrac{|-c^T x^{(k)} + b^T y^{(k)} - \kappa^{(k)}|}{\max(1, ||[-c^T, b^T, 1]||_\infty)}
\end{aligned}$$

are employed which are scaled primal, dual, and gap infeasibility measures, respectively. Also define

$$\rho_A^{(k)} := \frac{|c^T x^{(k)} - b^T y^{(k)}|}{\tau^{(k)} + |b^T y^{(k)}|} = \frac{|c^T x^{(k)}/\tau^{(k)} - b^T y^{(k)}/\tau^{(k)}|}{1 + |b^T y^{(k)}/\tau^{(k)}|} \tag{45}$$

that measures the number of significant digits in the objective value. The kth iterate is considered nearly feasible and optimal if

$$\rho_P^{(k)} \le \bar{\rho}_P, \quad \rho_D^{(k)} \le \bar{\rho}_D, \quad \text{and} \quad \rho_A^k \le \bar{\rho}_A,$$

where $\bar{\rho}_P, \bar{\rho}_D, \bar{\rho}_A \in (0, 1]$ are (small) user specified constants. In this case the solution

$$(x^*, y^*, s^*) = (x^{(k)}, y^{(k)}, s^{(k)})/\tau^{(k)}$$

is reported to be an optimal solution to (P).

The algorithm is also terminated if

$$\rho_P^{(k)} \le \bar{\rho}_P, \quad \rho_D^{(k)} \le \bar{\rho}_D, \quad \rho_G^{(k)} \le \bar{\rho}_G, \quad \text{and} \quad \tau^{(k)} \le \bar{\rho}_I \max(1, \kappa^{(k)}),$$

where $\bar{\rho}_I \in]0, 1[$ is a small user specified constant. In this case a feasible solution to the homogeneous model with a small τ has been computed. Therefore, it is concluded that the problem is primal or dual infeasible. If $b^T y^{(k)} > 0$, then the primal problem is concluded to be infeasible and if $c^T x^{(k)} < 0$, then the dual problem is concluded to be infeasible. Moreover, the algorithm is terminated if

$$\mu^{(k)} \le \bar{\rho}_\mu \mu^{(0)} \quad \text{and} \quad \tau^{(k)} \le \bar{\rho}_I \min(1, \kappa^{(k)})$$

and the problem is reported to be ill-posed. The parameters $\bar{\rho}_A, \bar{\rho}_G, \bar{\rho}_\mu, \bar{\rho}_I \in (0, 1]$ are all user specified constants.

7. Implementation

The algorithm for solving conic quadratic optimization problems have now been presented in details and we will therefore turn our attention to a few implementational issues.

7.1. Input format

In our implementation we allow the user to specify conic models in the form

$$\begin{array}{lrcccl} \text{minimize} & & c^T x + c^f & & & \\ \text{subject to} & l^c \leq & Ax & \leq & u^c, & \\ & l^x \leq & x & \leq & u^x, & \\ & & x^i \in K^i, & & i = 1, \ldots, k & \end{array} \tag{46}$$

where x^i is a vector comprised of parts of the decision variables x. Each decision variable is allowed to be the member of at most one vector x^i.

If the conic constraint

$$x^i \in K^i \tag{47}$$

is excluded from the problem (46), then the problem is a linear optimization problem that can be specified using the standard MPS input format employed by virtually all optimization software packages. See [22] for details about the MPS format.

It is therefore natural to extend the MPS format to allow for specification of conic constraints of the form (47). We have done that by introducing a new section type in the MPS format named `CSECTION`. In Figure 1 an example of a `CSECTION` is presented which specifies the rotated quadratic cone

$$x_5^2 + x_6^2 \leq 2x_1x_3, \;\; x_1, x_3 \geq 0.$$

In Figure 1 `k1` is a name assigned to the cone and the key word `RQUAD` denotes the cone type. An alternative to `RQUAD` is `QUAD` which implies the `CSECTION` specifies a quadratic cone.

It should be obvious how this extension allows specification of multiple conic quadratic constraints.

7.2. Presolving the problem

Before the problem is optimized it is preprocessed to remove obvious redundancies using the techniques presented in [3]. For example fixed variables are removed, obviously

```
CSECTION      k1                          RQUAD
    x1
    x3
    x5
    x6
```

Fig. 1. An example of a `CSECTION`

Table 1. Algorithmic parameters

Constant	Value	Section
β	10^{-8}	3.1
δ	0.5	4.2
$\bar{\rho}_P$	10^{-8}	6.2
$\bar{\rho}_D$	10^{-8}	6.2
$\bar{\rho}_A$	10^{-8}	6.2
$\bar{\rho}_G$	10^{-8}	6.2
$\bar{\rho}_I$	10^{-10}	6.2
$\bar{\rho}_\mu$	10^{-10}	6.2

redundant constraints are removed, linear dependencies in A are removed. Finally, some of the linear free variables are substituted out of the problem.

7.3. Scaling the problem

Before optimizing the problem, the constraints are scaled so that the maximal element in each linear constraint is one in an absolute sense. This improves numerical stability for badly scaled problems.

8. Computational results

In this section we turn our attention to evaluate the practical efficiency of the presented algorithm.

The algorithm has been implemented in the programming language `C` and is one of the several optimizers available in the software packages MOSEK optimization tools and the MOSEK optimization toolbox[2] for MATLAB version 2.

During the computational testing all the algorithmic parameters are held constant at the values listed in Table 1.

The computational test is performed on a 1.2GHz PIII PC(DELL Inspiron 8100 laptop) having 0.5GB of RAM and running the Windows XP operating system.

In Table 2 and 3 the test problems are shown along with the size of the problems before and after the presolve procedure has been applied to the problems.

The problems belonging to the `nb*`, and `sched*` families are DIMACS Challenge problems [26] that have been converted from SeDuMi format to the MOSEK extended MPS format. The problems `c-qssp*` and `c-nql*` are the original versions of the `qssp*` and `nql*` problems submitted by one of the authors to the DIMACS challenge. We use the original versions of the problems instead of the reformulated DIMACS versions because our opinion is that the original formulation is better. For example, the original formulation does not contain splitted free variables. The `dttd*` family of problems are multi load truss topology design problems, see [11]. The `c-traffic*` problems arises from a model developed by C. Roos to study a traffic phenomena. The

[2] See http://www.mosek.com for further details about MOSEK.

Table 2. Name of the test problems

Name	Problem
c-antoon1	1
dttd7-7-2	2
dttd7-7-5	3
dttd9-9-2	4
dttd9-9-5	5
dttd11-11-2	6
dttd11-11-5	7
dttd13-13-2	8
nb	9
nb_L1	10
nb_L2	11
nb_L2_bessel	12
c-nql30	13
c-nql60	14
c-nql90	15
c-nql180	16
c-qssp30	17
c-qssp60	18
c-qssp90	19
c-qssp180	20
c-than1	21
c-than2	22
c-than3	23
c-than4	24
c-than5	25
c-traffic-12	26
c-traffic-24	27
c-traffic-36	28
sched_100_100_orig	29
sched_100_100_scaled	30
sched_100_50_orig	31
sched_100_50_scaled	32
sched_200_100_orig	33
sched_200_100_scaled	34
sched_50_50_orig	35
sched_50_50_scaled	36

remaining problems have been obtained by the authors from various sources. The problems in the `c-nql*` and `c-qssp*` families have previously been solved in [10] and are dual problems of minimum sum of norm problems.

It is evident from Table 3 that the presolve procedure in some cases is effective in reducing the problem size. In particular the `c-nql*`, `c-than*`, and `c-traffic*` problems are reduced significantly by the presolve procedure. It can also be observed that the presolve is currently ineffective if all the variables are member of a quadratic cone. This is for example true for the `sched*` problems.

The purpose of the subsequent Table 4 is to show various optimizer related statistics i.e. the size of the problems actually solved and performance related statistics. In some cases additional constraints and variables are added to the problems to state the problems in the required conic quadratic form. The first two columns of Table 4 show the number of constraints and the number of quadratic cones in each problem. Next the total number of variables, the number variables which have both a finite lower and upper bound, and the number of variables that are members of a quadratic cone are shown. Finally, the number of interior-point iterations performed to optimize the problems, the time spend in the interior-point optimizer, and the total solution time are shown.

Table 3. The test problems

Pro-	Before presolve			After presolve		
blem	Constraints	Variables	Nz(A)	Constraints	Variables	Nz(a)
1	5557	6508	14165	3102	5456	11307
2	1326	6931	12833	1326	6931	12833
3	5046	17326	33815	5046	17326	33815
4	3495	19225	36758	3495	19225	36758
5	13542	48061	96701	13542	48061	96701
6	7648	43231	84427	7648	43231	84427
7	29926	108076	221875	29926	108076	221875
8	14745	84709	167936	14745	84709	167936
9	123	2383	191519	123	2383	191519
10	915	3176	192312	915	3176	192312
11	123	4195	402285	122	4194	402284
12	123	2641	208817	122	2640	208816
13	2780	3601	16969	1018	1840	26487
14	10960	14401	68139	4093	7535	107927
15	24540	32401	153509	9397	17259	242996
16	97680	129601	614819	37737	69659	972084
17	1800	5674	34959	1799	5673	34950
18	7200	22144	141909	7199	22143	141900
19	16200	49414	320859	16199	49413	320850
20	64800	196024	1289709	64799	196023	1289700
21	229864	264557	944944	188092	222785	820844
22	4068	4861	16884	3096	3889	14868
23	181	520	1524	158	497	2262
24	400	541	1884	220	361	1620
25	312208	374461	1320564	237316	299569	1170148
26	2266	1760	4801	830	1248	4031
27	4570	3596	9829	1727	2601	9973
28	6874	5432	14857	2635	3965	14787
29	8338	18240	104902	8338	18240	104902
30	8337	18238	114899	8337	18238	114899
31	4844	9746	55291	4844	9746	55291
32	4843	9744	60288	4843	9744	60288
33	18087	37889	260503	18087	37889	260503
34	18086	37887	280500	18086	37887	280500
35	2527	4979	25488	2527	4979	25488
36	2526	4977	27985	2526	4977	27985
Sum	936548	1597150	7549458	724015	1391568	7765866

The conclusion that can be made based on Table 4 is that even though some of the problems are large then the solution time is small. Moreover, as can be expected from an interior-point method the number of iterations are quite low and tend to grow slowly with problem size.

Finally, in Table 5 we show feasibility and optimality related measures. The columns ρ_e^p and ρ_e^d report the infinity norm violations of the solution in the primal and dual equalities respectively. The columns ρ_b^p and ρ_b^d report the solutions maximal violation of the simple bounds in the primal and dual problems, respectively. Next, the columns ρ_c^p and ρ_c^d report the solutions maximal violation of the conic quadratic constraints in the primal and dual problems. For a quadratic cone the violation is given by

$$\max\{0.0, -(x_1 - \|x_{2:n}\|)\}. \tag{48}$$

For a rotated quadratic cone we use the fact that if x^i is in a rotated quadratic cone then $T^i x^i$ is in a quadratic cone. Therefore, the violation for a rotated quadratic cone is computed as the violation of $T^i x^i$ using formula (48). Finally, the optimal primal

Table 4. Problems and results of the optimizer

Problem	Constraints: Linear	Constraints: Q. conic	Variables: Total	Variables: Upper	Variables: Q. conic	Optimizer: Iterations	Optimizer: Time (s) Interior-point	Optimizer: Time (s) Total
1	3602	951	6358	801	2302	34	1.5	1.5
2	1326	2310	6931	0	6930	32	1.6	1.6
3	5046	5775	17326	0	17325	31	7.3	7.3
4	3495	6408	19225	0	19224	32	6.0	6.0
5	13542	16020	48061	0	48060	40	33.3	33.3
6	7648	14410	43231	0	43230	39	19.6	19.6
7	29926	36025	108076	0	108075	40	91.5	91.5
8	14745	28236	84709	0	84708	39	39.6	39.6
9	123	793	2383	0	2379	21	4.7	4.7
10	915	793	3176	0	2379	17	4.8	4.8
11	122	839	4195	0	4191	13	9.6	9.6
12	122	839	2641	0	2637	12	2.9	2.9
13	1018	900	2740	0	2699	14	1.2	1.2
14	4093	3600	11135	0	10799	15	7.0	7.0
15	9397	8100	25359	0	24299	15	19.1	19.1
16	37737	32400	102059	0	97199	18	142.8	142.8
17	1799	1891	7564	0	7564	15	1.5	1.5
18	7199	7381	29524	0	29524	19	10.3	10.3
19	16199	16471	65884	0	65884	23	34.0	34.0
20	64799	65341	261364	0	261364	23	222.4	222.4
21	188092	55696	278481	0	278480	15	133.5	133.5
22	3096	1296	5185	0	5184	12	0.7	0.7
23	158	120	497	0	480	14	0.1	0.1
24	220	120	481	0	480	11	0.1	0.1
25	237316	99856	399425	0	399424	17	218.0	218.0
26	1248	429	2106	825	1276	15	0.4	0.4
27	2601	897	4395	1725	2668	26	1.7	1.7
28	3965	1365	6695	2625	4060	24	2.8	2.8
29	8338	2	18240	0	8238	27	6.0	6.0
30	8337	1	18238	0	8236	37	8.3	8.3
31	4844	2	9746	0	4744	24	2.7	2.7
32	4843	1	9744	0	4742	25	2.9	2.9
33	18087	2	37889	0	17887	30	17.5	17.5
34	18086	1	37887	0	17885	32	19.0	19.0
35	2527	2	4979	0	2477	26	1.1	1.1
36	2526	1	4977	0	2475	19	0.9	0.9
Sum	727137	409274	1690906	5976	1599508	846	1076.1	1076.1

objective value and the number of figures that are identical in the optimal primal and dual objective values are reported.

In general the feasibility measures are good. However, they are quite large for the `sched*` problems. The reason is that in these problems $||b||$ is very large and hence the reported infeasibility measures reported in Table 5 are small in a relative sense. Finally, the required 8 figures accuracy in the objective value is not obtained for all the problems. The reason is the optimizer stopped prematurely due to numerical problems (i.e., inaccurate search direction) or some of the optimality is lost during the postsolve. (The postsolve is the inverse process of the presolve.)

9. Conclusions

The present work discusses a primal-dual interior-point method designed to solve large-scale sparse conic quadratic optimization problems. The main theoretical features of the

Table 5. Feasibility measures and objective values

Pro-blem	Primal Feasibility ρ_e^p	ρ_b^p	ρ_c^p	Dual Feasibility ρ_e^d	ρ_b^d	ρ_c^d	Primal objective	Sig. fig.
1	4e-010	8e-004	0e+000	1e-010	0e+000	0e+000	5.903525014e-002	4
2	0e+000	5e-006	0e+000	5e-009	0e+000	0e+000	4.297394102e+003	8
3	0e+000	4e-005	0e+000	3e-008	0e+000	0e+000	7.016563290e+003	8
4	0e+000	7e-005	0e+000	2e-008	0e+000	0e+000	6.616024409e+003	8
5	0e+000	6e-005	0e+000	4e-008	0e+000	0e+000	1.657993991e+004	8
6	0e+000	1e-004	0e+000	2e-008	0e+000	0e+000	1.250831510e+004	8
7	0e+000	1e-004	0e+000	2e-008	0e+000	0e+000	1.367845588e+004	8
8	0e+000	5e-004	0e+000	3e-008	0e+000	0e+000	1.917127957e+004	8
9	0e+000	8e-010	0e+000	5e-008	0e+000	0e+000	-5.070309465e-002	12
10	0e+000	3e-010	0e+000	2e-009	0e+000	0e+000	-1.301227054e+001	8
11	0e+000	1e-005	0e+000	3e-006	0e+000	0e+000	-1.628965444e+000	7
12	0e+000	7e-010	0e+000	6e-010	0e+000	0e+000	-1.025695041e-001	8
13	3e-015	4e-012	0e+000	5e-009	0e+000	0e+000	-9.460280765e-001	9
14	3e-015	7e-006	0e+000	1e-009	0e+000	0e+000	-9.350556725e-001	9
15	4e-015	3e-005	0e+000	7e-009	0e+000	0e+000	-9.313885905e-001	9
16	4e-015	2e-005	0e+000	3e-009	0e+000	0e+000	-9.277161007e-001	9
17	0e+000	4e-016	0e+000	1e-009	0e+000	0e+000	-6.496675159e+000	10
18	0e+000	2e-015	0e+000	7e-009	0e+000	0e+000	-6.562696811e+000	9
19	0e+000	5e-013	0e+000	2e-007	0e+000	0e+000	-6.594401303e+000	11
20	0e+000	3e-013	2e-009	5e-009	0e+000	0e+000	-6.639434488e+000	9
21	6e-012	2e-007	0e+000	4e-009	0e+000	0e+000	-5.594102993e+000	12
22	5e-016	5e-008	0e+000	1e-008	0e+000	0e+000	-5.304443799e-001	10
23	8e-017	1e-006	0e+000	4e-011	0e+000	0e+000	7.763116434e-001	7
24	4e-016	1e-005	0e+000	5e-007	0e+000	0e+000	-7.762928565e-001	7
25	1e-016	2e-006	0e+000	3e-008	0e+000	0e+000	-4.637502076e-001	10
26	2e-014	3e-007	1e-008	3e-008	6e-009	0e+000	-8.634418052e+002	7
27	3e-014	5e-007	2e-008	4e-008	8e-009	0e+000	-2.683777027e+003	7
28	9e-014	7e-007	2e-008	9e-008	6e-009	0e+000	-5.390245694e+003	7
29	3e-014	3e+002	0e+000	2e-006	0e+000	0e+000	7.173671485e+005	8
30	6e-014	8e-001	0e+000	4e-009	0e+000	0e+000	2.733145701e+001	8
31	3e-011	4e+001	0e+000	3e+001	0e+000	0e+000	1.818898045e+005	8
32	1e-014	4e-001	0e+000	3e-008	0e+000	0e+000	6.716628515e+001	8
33	3e-011	1e+002	0e+000	2e-006	0e+000	0e+000	1.413597623e+005	8
34	1e-011	4e-001	0e+000	6e-008	0e+000	0e+000	5.181247057e+001	9
35	4e-012	9e-001	0e+000	2e-003	0e+000	0e+000	2.667297586e+004	8
36	7e-015	1e-002	0e+000	5e-007	0e+000	0e+000	7.852038453e+000	9

algorithm are it uses the Nesterov-Todd search direction and the homogeneous model. Moreover, the algorithm has been extended with a Mehrotra predictor-corrector scheme, treats the rotated quadratic cone without introducing additional variables and constraints and employs structure and sparsity exploiting linear algebra.

The presented computational results indicate that the suggested algorithm is capable of computing accurate solutions to very large sparse conic quadratic optimization problems in short time.

Acknowledgements. We would like to thank K. D. Andersen and A. Nemirovskii for helping us during the collection of the test problems.

A. Appendix

Definition 4. *Let K be a pointed and closed convex cone, then K is self-dual if*

$$K = K_*$$

and homogeneous if for any $x, s \in int(K)$

$$\exists B \in R^{n\times n} : \; B(K) = K, \; Bx = s.$$

Self-dual and homogeneous cones have been studied extensively in the literature and are called self-scaled by Nesterov and Todd, [23].

Proof of Lemma 2:

We first prove $X^i S^i = 0$ implies $(x^i)^T s^i = 0$. Observe that

$$\begin{aligned} e^T XSe &= \sum_{i=1}^{n} (e^i)^T X^i S^i e^i \\ &= \sum_{i=1}^{k} (T^i x^i)^T T^i s^i \\ &= x^T s. \end{aligned} \tag{49}$$

This implies that any solution satisfying (19) is also a complementary solution. Next we prove if $(x^i)^T s^i = 0$, then $X^i S^i e^i = 0$. In explicit form the complementarity conditions can be stated as

$$\begin{aligned} (x^i)^T s^i &= 0, \\ (T^i x^i)_1 (T^i s^i)_{2:n} + (T^i s^i)_1 (T^i x^i)_{2:n} &= 0. \end{aligned}$$

Note that $T^i x^i, T^i s^i \in K_q$. This implies if either $(T^i x^i)_1 = 0$ or $(T^i s^i)_1 = 0$ then (19) is true as claimed. Therefore, we assume that this is not the case. Since x and s are complementary one has

$$\begin{aligned} 0 &= x^T s \\ &= \sum_{i=1}^{k} (T^i x^i)^T T^i s^i \\ &\geq \sum_{i=1}^{k} (T^i x^i)_1 (T^i s^i)_1 - ||(T^i x^i)_{2:n}|| \, ||(T^i s^i)_{2:n}|| \\ &\geq \sum_{i=1}^{k} \sqrt{(x^i)^T Q^i x^i (s^i)^T Q^i s^i} \\ &\geq 0. \end{aligned}$$

The first inequality follows from the Cauchy-Schwartz inequality and the second inequality follows from $T^i x^i, T^i s^i \in K_q$. This implies that both $(x^i)^T Q^i x^i = 0$ and $(s^i)^T Q^i s^i = 0$ hold. Moreover, we conclude that

$$(T^i x^i)_1 (T^i s^i)_1 = ||(T^i x^i)_{2:n}|| \, ||(T^i s^i)_{2:n}||.$$

However, this can only be the case if

$$\exists \alpha : \; (T^i x^i)_{2:n} = \alpha (T^i s^i)_{2:n}$$

for some $\alpha \in R$. Therefore, considering the assumptions we have

$$\begin{aligned} 0 &= (x^i)^T s^i \\ &= (T^i x^i)_1 (T^i s^i)_1 + \alpha ||(T^i s^i)_{2:n^i}||^2 \end{aligned}$$

and

$$\alpha = -\frac{(T^i x^i)_1}{(T^i s^i)_1}$$

implying that the complementarity conditions (19) are satisfied.

Proof of Lemma 3:

i), ii), and iii) follow immediately from the definition of a scaling. iv) is proved next. In the case K^i is R_+ then the statement is obviously true. In the case K^i is the quadratic cone then due to $W^i Q^i W^i = I$ we have that

$$w_1^2 - ||w_{2:n}||^2 = 1,$$

where w denotes the first row of W^i. This implies

$$\begin{aligned} \bar{x}_i^1 &= (e^i)^T \bar{x}^i \\ &= (e^i)^T \Theta W^i x^i \\ &= \theta_i (w_1 x_1^i + w_{2:n}^T x_{2:n}^i) \\ &\geq \theta_i (w_1 x_1^i - ||w_{2:n^i}|| || x_{2:n}^i ||) \\ &= \theta_i (\sqrt{1 + ||w_{2:n^i}||^2} x_1^i) \\ &\geq \theta_i x_1^i \\ &\geq 0 \end{aligned}$$

and

$$(\bar{x}^i)^T Q^i \bar{x}^i = \theta_i (x^i)^T Q^i x^i \geq 0.$$

Hence, $x^i \in K^i$ implies $\bar{x}^i \in K^i$ for the quadratic cone. Similarly, it is easy to verify that $x^i \in \text{int}(K^i)$ implies $\bar{x}^i \in \text{int}(K^i)$. Now, assume that K^i is a rotated quadratic cone and $x^i, s^i \in K^i$. Let $\hat{x}^i := T^i x^i$ and $\hat{s}^i := T^i s^i$. Then $\hat{x}^i, \hat{s}^i \in K_q$. Therefore, a scaling $\hat{\theta}_i \hat{W}^i$ exists such that

$$\hat{s}^i = T^i s^i = (\hat{\theta}_i \hat{W}^i)^2 T^i x^i = (\hat{\theta}_i \hat{W}^i)^2 \hat{x}^i,$$

where

$$\begin{aligned} \bar{\theta}_i^2 &= \sqrt{\tfrac{(T^i s^i)^T \hat{Q}^i T^i s^i}{(T^i x^i)^T \hat{Q}^i T^i x^i}} \\ &= \theta_i^2. \end{aligned}$$

This implies

$$s^i = \hat{\theta}_i^2 T^i \hat{(} W^i)^2 T^i x^i$$

that shows $W^i = T^i \hat{W}^i T^i$. We know that $\hat{\theta}_i \hat{W}^i T^i x^i \in K_q$ and hence $\hat{\theta}_i T^i \hat{W}^i T^i x^i = \theta_i W^i x^i \in K_r$. Finally, vi) follows from v) and from the fact $\sqrt{x^T Q x s^T Q s} = \sqrt{\bar{x}^T Q \bar{x} \bar{s}^T Q \bar{s}}$.

Proof of Lemma 5:

Equation (28) follows immediately from Lemma 3. It is trivial to compute W^i when K^i is R_+ is the case. Next assume that K^i is a quadratic cone. First, let us define

$$\tilde{w}_1 := w_1^i \quad \text{and} \quad \tilde{w}_2 := w_{2:n^i}^i,$$

then

$$\begin{aligned} W^i W^i &= \begin{bmatrix} \tilde{w}_1 & \tilde{w}_2^T \\ \tilde{w}_2 & I + \frac{\tilde{w}_2 \tilde{w}_2^T}{1+\tilde{w}_1} \end{bmatrix} \begin{bmatrix} \tilde{w}_1 & \tilde{w}_2^T \\ \tilde{w}_2 & I + \frac{\tilde{w}_2 \tilde{w}_2^T}{1+\tilde{w}_1} \end{bmatrix}^T \\ &= \begin{bmatrix} ||\tilde{w}||^2 & \left(1 + \tilde{w}_1 + \frac{||\tilde{w}_2||^2}{1+\tilde{w}_1}\right) \tilde{w}_2^T \\ \left(1 + \tilde{w}_1 + \frac{||\tilde{w}_2||^2}{1+\tilde{w}_1}\right) \tilde{w}_2 & \tilde{w}_2 \tilde{w}_2^T + \left(I + \frac{\tilde{w}_2 \tilde{w}_2^T}{1+\tilde{w}_1}\right)^2 \end{bmatrix} \\ &= -Q^i + 2w^i (w^i)^T, \end{aligned} \tag{50}$$

because

$$(w_1^i)^2 - ||w_{2:n^i}^i||^2 = 1$$

follows from the definition of Q^i and the fact $W^i Q^i W^i = Q^i$. When (50) is combined with (27) one has

$$s^i = \theta_i^2 (-Q^i + 2w^i (w^i)^T) x^i$$

and

$$(x^i)^T s^i = \theta_i^2 (-(x^i)^T Q^i x^i + 2((w^i)^T x^i)^2).$$

Therefore,

$$\begin{aligned} 2\theta_i^2 ((w^i)^T x^i)^2 &= (x^i)^T s^i + \theta_i^2 (x^i)^T Q^i x^i \\ &= (x^i)^T s^i + \sqrt{(x^i)^T Q^i x^i (s^i)^T Q^i s^i} \end{aligned}$$

and then

$$w^i = \frac{\theta_i^{-1} s^i + \theta_i Q^i x^i}{\sqrt{2} \sqrt{(x^i)^T s^i + \sqrt{(x^i)^T Q^i x^i (s^i)^T Q^i s^i}}}.$$

Clearly,

$$(x^i)^T s^i + \sqrt{(x^i)^T Q^i x^i (s^i)^T Q^i s^i} > 0$$

when $x^i, s^i \in \operatorname{int}(K)^i$. Now assume that K^i is the rotated quadratic cone. Let $\hat{x}^i := T^i x^i$ and $\hat{s}^i := T^i s^i$ that imply $\hat{x}^i, \hat{s}^i \in \operatorname{int}(K_q)$. Moreover, let us define

$$\hat{Q}^i := T^i Q^i T^i$$

and

$$\hat{W}^i := T^i W^i T^i.$$

Since $T^i T^i = I$ and $W^i Q^i W^i = Q^i$ we have $\bar{W}^i \hat{Q}^i \hat{W}^i = \hat{Q}^i$. Further, by definition we have $s^i = \theta_i^2 (W^i)^2 x^i$ that implies

$$\begin{aligned} \hat{s}^i &= T^i s^i \\ &= \theta_i^2 T^i (W^i)^2 x^i \\ &= \hat{\theta}_i^2 (T^i W^i T^i)^2 T^i x^i \\ &= \hat{\theta}_i^2 (\hat{W}^i)^2 \hat{x}^i \end{aligned} \tag{51}$$

because

$$\begin{aligned} \hat{\theta}_i^2 &= \sqrt{\frac{(T^i s^i)^T \bar{Q}^i T^i s^i}{(T^i x^i)^T \bar{Q}^i T^i x^i}} \\ &= \theta_i^2. \end{aligned}$$

Now $\hat{x}^i, \hat{s}^i \in \operatorname{int}(K_q)$ that implies that we can use relation (29) to compute the scaling matrix $\hat{W}^i$ in (51). Therefore,

$$\begin{aligned} \hat{w}^i &= \frac{\hat{\theta}_i^{-1} \hat{s}^i + \hat{\theta}_i \hat{Q}^i \hat{x}^i}{\sqrt{2}\sqrt{(\hat{x}^i)^T \hat{s}^i + \sqrt{(\hat{x}^i)^T \hat{Q}^i \hat{x}^i (T^i s^i)^T \hat{Q}^i T^i s^i}}} \\ &= \frac{\theta_i^{-1} T^i s^i + \theta_i T^i Q^i x^i}{\sqrt{2}\sqrt{(x^i)^T s^i + \sqrt{(x^i)^T Q^i x^i (s^i)^T Q^i s^i}}} \\ &= T^i w^i \end{aligned}$$

and

$$\begin{aligned} \hat{W}^i &= -\hat{Q}^i + \frac{(e_1^i + \hat{w}^i)(e_1^i + \hat{w}^i)^T}{1 + \hat{w}^i} \\ &= T^i \left(-Q^i + \frac{(T^i e_1^i + w^i)(T^i e_1^i + w^i)^T}{1 + (e_1^i)^T T^i w^i} \right) T^i \\ &= T^i W^i T^i \end{aligned}$$

from which (32) follows.

Clearly, W is symmetric in all cases and we leave it to the reader to prove that W is positive definite.

References

1. F. Alizadeh and S. H. Schmieta. Optimization with semidefinite, quadratic and linear constraints. Technical Report RRR 23-97, RUTCOR, Rutgers Center for Operations Research, P.O. Box 5062, New Brunswick, New Jersey, November 1997.
2. E. D. Andersen. Handling free variables in primal-dual interior-point methods using a quadratic cone approach. Technical report, 2002. in preparation.
3. E. D. Andersen and K. D. Andersen. Presolving in linear programming. *Math. Programming*, 71(2):221–245, 1995.
4. E. D. Andersen and K. D. Andersen. A parallel interior-point based linear programming solver for shared-memory multiprocessor computers: A case study based on the XPRESS LP solver. Technical Report CORE DP 9808, CORE, UCL, Belgium, 1997.

5. E. D. Andersen and K. D. Andersen. The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In H. Frenk, K. Roos, T. Terlaky, and S. Zhang, editors, *High Performance Optimization*, pages 197–232. Kluwer Academic Publishers, 2000.
6. E. D. Andersen, J. Gondzio, Cs. Mészáros, and X. Xu. Implementation of interior point methods for large scale linear programming. In T. Terlaky, editor, *Interior-point methods of mathematical programming*, pages 189–252. Kluwer Academic Publishers, 1996.
7. K. D. Andersen. *Minimizing a Sum of Norms (Large Scale solutions of symmetric positive definite linear systems)*. PhD thesis, Odense University, 1995.
8. K. D. Andersen. A Modified Schur Complement Method for Handling Dense Columns in Interior-Point Methods for Linear Programming. *ACM Trans. Math. Software*, 22(3):348–356, 1996.
9. K. D. Andersen. QCOPT a large scale interior-point code for solving quadratically constrained quadratic problems on self-scaled cone form. 1997.
10. K. D. Andersen, E. Christiansen, and M. L. Overton. Computing limit loads by minimizing a sum of norms. *SIAM Journal on Scientific Computing*, 19(2), March 1998.
11. A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. MPS/SIAM Series on Optimization. SIAM, 2001.
12. E. de Klerk, C. Roos, and T. Terlaky. Initialization in semidefinite programming via a self–dual, skew–symmetric embedding. *OR Letters*, 20:213–221, 1997.
13. C. Roos, E. D. Andersen, and T. Terlaky. Notes on duality in second order and p-order cone optimization. Technical report, 1999. Accepted for publication in Optimization.
14. D. Goldfarb, K. Scheinberg, and S. Schmieta. A product-form Cholesky factorization implementation of an interior-point method for second-order cone programming. Technical report.
15. O. Güler. Barrier functions in interior-point methods. *Math. Oper. Res.*, 21:860–885, 1996.
16. M. Kojima, S. Mizuno, and A. Yoshise. A primal-dual interior point algorithm for linear programming. In N. Megiddo, editor, *Progress in Mathematical Programming: Interior-Point Algorithms and Related Methods*, pages 29–47. Springer Verlag, Berlin, 1989.
17. M. S. Lobo, L. Vanderberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra Appl.*, pages 193–228, November 1998.
18. I. J. Lustig, R. E. Marsten, and D. F. Shanno. Interior point methods for linear programming: Computational state of the art. *ORSA J. on Comput.*, 6(1):1–15, 1994.
19. S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM J. on Optim.*, 2(4):575–601, 1992.
20. R. D. C. Monteiro and I. Adler. Interior path following primal-dual algorithms. Part I: Linear programming. *Math. Programming*, 44:27–41, 1989.
21. R. D. C. Monteiro and T. Tsuchiya. Polynomial convergence of primal-dual algorithms for the second-order cone program based on the MZ-family of directions. *Math. Programming*, 88(1):61–83, 2000.
22. J. L. Nazareth. *Computer Solution of Linear Programs*. Oxford University Press, New York, 1987.
23. Y. Nesterov and M. J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Math. Oper. Res.*, 22(1):1–42, February 1997.
24. Yu. Nesterov, M. J. Todd, and Y. Ye. Infeasible-start primal-dual methods and infeasibility detectors for nonlinear programming problems. *Math. Programming*, 84(2):227–267, February 1999.
25. Yu. E Nesterov and M. J. Todd. Primal-dual interior-point methods for self-scaled cones. *SIAM J. on Optim.*, 8:324–364, 1998.
26. G. Pataki and S. Schmieta. The DIMACS library of semidefinte-quadratic-linear programs. Technical report, Computational Optimization Research Center, Columbia University, November 1999.
27. S. H. Schmieta and F. Alizadeh. Associative algebras, symmetric cones and polynomial time interior point algorithms. Technical Report RRR 17-98, RUTCOR, Rutgers Center for Operations Research, P.O. Box 5062, New Brunswick, New Jersey, June 1998.
28. J. F. Sturm. *Primal-dual interior point approach to semidefinite programming*. PhD thesis, Tinbergen Institute, Erasmus University Rotterdam, 1997.
29. J. F. Sturm. SeDuMi 1.02, a MATLAB toolbox for optimizing over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.
30. J. F. Sturm. Theory and algorithms of semi-definite programming. In H. Frenk, K. Roos, T. Terlaky, and S. Zhang, editors, *High Performance Optimization*, pages 3–194. Kluwer Academic Publishers, 2000.
31. T. Tsuchiya. A polynomial primal-dual path-following algorithm for second-order cone programming. Technical report, The Institute of Statistical Mathematics, Tokyo, Japan, October 1997.
32. L. Tuncel. Generalization of primal-dual interior-point methods to convex optimization problems in conic form. *Foundations of Computational Mathematics*, 1:229–254, 2001.
33. S. J. Wright. *Primal-dual interior-point methods*. SIAM, Philadelphia, 1997.