A Homogeneous Interior-Point Algorithm for Nonsymmetric Convex Conic Optimization

Anders Skajaa · Yinyu Ye

Abstract A homogeneous infeasible-start interior-point algorithm for solving nonsymmetric convex conic optimization problems is presented. Starting each iteration from the vicinity of the central path, the method steps in the approximate tangent direction and then applies a correction phase to locate the next well-centered primal-dual point. Features of the algorithm include that it makes use only of the primal barrier function, that it is able to detect infeasibilities in the problem and that no phase-I method is needed. We prove convergence to ϵ-accuracy in $\mathcal{O}(\sqrt{\nu}\log{(1/\epsilon)})$ iterations. To improve performance, the algorithm employs a new Runge-Kutta type second order search direction suitable for the general nonsymmetric conic problem. Moreover, quasi-Newton updating is used to reduce the number of factorizations needed, implemented so that data sparsity can still be exploited. Extensive and promising computational results are presented for the p-cone problem, the facility location problem, entropy problems and geometric programs; all formulated as nonsymmetric convex conic optimization problems.

Anders Skajaa
Department of Informatics and Mathematical Modelling
Technical University of Denmark, DK-2800, Kgs. Lyngby, Denmark.
E-mail: andsk@imm.dtu.dk

Yinyu Ye
Department of Management Science and Engineering
Stanford University, CA 94305-4121, USA.
E-mail: yinyu-ye@stanford.edu

1 Introduction

This paper is concerned with conic optimization problem pairs of the form

$$
\text{PRIMAL} \begin{cases} \min_x c^T x \\ \text{s.t.} \quad Ax = b \\ \qquad x \in \mathcal{K} \end{cases} \qquad \text{DUAL} \begin{cases} \max_{y,s} b^T y \\ \text{s.t.} \quad A^T y + s = c \\ \qquad s \in \mathcal{K}^*, \ y \in \mathbb{R}^m \end{cases} \qquad \text{(PD)}
$$

where $x, c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, \mathcal{K} \subset \mathbb{R}^n$ is a proper cone (i.e. it is convex, pointed, closed and has nonempty interior) and $\mathcal{K}^* = \{ s \in \mathbb{R}^n : s^T x \geq 0, \ \forall x \in \mathcal{K} \}$ is its dual cone, which is also proper. We are further assuming that $m \leq n$ and that $\text{rank}(A) = m$.

If $\mathcal{K}$ is the positive orthant $\mathbb{R}^n_+$, then (PD) is a linear programming (LP) problem in standard form and its dual. Solution methods for LP have been studied for long in different settings and until the emergence of interior-point methods (IPMs), the most prominent method was the simplex method, developed by Dantzig in the 1940s. The introduction of IPMs is usually ascribed to Karmarkar [10] in 1984 and since then, research in the area has been extensive.

In [17], it was studied how to extend the ideas of IPMs to the nonlinear case. If $\mathcal{K}$ admits a *self-scaled* barrier function $F \colon \mathcal{K}^\circ \mapsto \mathbb{R}$, problems of the type (PD) are efficiently solvable using long-step symmetric primal-dual IPMs [18,19]. The practical efficiency of these algorithms has been widely verified, see e.g. [1,2,25].

In [9], Güler demonstrated that self-scaled cones are identical to those that are *symmetric*; a class that encompasses just five cones of which only three are interesting for optimization. These cones are the positive orthant (leading to LP), the Lorentz cone (leading to quadratic cone programming which generalizes quadratic programming and second order cone programming) and the positive semidefinite cone (leading to semidefinite programming).

Although these three self-scaled cones allow for modelling of a great variety of constraints [4], many important types of constraints do not fall in this class. Examples include entropy type constraints: $x \log x \leq t$, p-cone constraints: $\|x\|_p \leq t$, and constraints arising in geometric programming [5]. Some of these constraints can be modelled using self-scaled cones, but this usually requires the introduction of many extra variables and constraints [4].

Theoretically, one can solve problems involving *any* convex constraint using a purely primal short-step barrier method and still obtain an algorithm with the best-known worst-case computational complexity. Such an algorithm is, however, known to be practically inefficient compared to a long-step primal-dual IPM. Other approaches are also possible and special algorithms for certain sub-classes of problems exist [27,30]. An approach known to be effective for general convex problems is to solve the monotone complementarity problem, see for example [3].

It may be beneficial to model nonsymmetric constraints more directly using *non*-self-scaled cones (nonsymmetric cones) such as the power cone or the exponential cone. This approach was employed by Nesterov in [16]. He proposed a method that mimics the ideas of a long-step primal-dual IPM for symmetric

cones by splitting each iteration into two phases. First, a pure primal correction phase is used to find a primal central point x and a scaling point w. These points are used to compute a feasible dual point s such that an *exact* scaling relation is satisfied: $s = \nabla^2 F(w)x$. Second, a truly symmetric primal-dual step in the approximate tangent direction is taken (a prediction step). This algorithm, however, assumes the existence of a strictly feasible primal-dual point and *requires* a strictly feasible initial primal point to start.

If knowledge of both the primal and the dual barrier function, their gradients and Hessians is assumed, truly primal-dual symmetric search directions can be constructed. This approach was used in [20] to solve a homogenous model of the general convex conic problem (PD). This leads to a method with some desirable properties but at the same time two crucial disadvantages: Firstly, the linear systems that must be solved in each iteration are twice the size compared to algorithms for self-scaled cones therefore increasing total computation time by a factor of $2^3 = 8$ for problems of equal dimension. Secondly, it can be difficult or impossible to find an expression for the dual barrier and its derivatives.

Building on the algorithms of [16] and [20], we present in this paper an infeasible-start primal-dual interior-point algorithm for a homogeneous model of (PD). This approach has proven successful for self-scaled cones [29, 2, 24] because it implies several desirable properties, among which are the ability to detect infeasibility in the problem pair and the ease of finding a suitable starting point, eliminating the need for a phase-I method. Unlike the algorithm in [20], our algorithm uses only the primal barrier function and therefore our linear systems are no larger than those appearing in IPMs for self-scaled cones.

In addition to the advantages induced by using a homogeneous model, we suggest the following improvements to reduce computational load. The Mehotra second order correction [12] is known to significantly improve practical performance of IPMs for linear and quadratic conic problems [12, 2, 25]. With the same goal in mind, we suggest a new way to compute a search direction containing second order information for the general (possibly non-self-scaled) conic problem. This search direction is inspired by Runge-Kutta methods for ordinary differential equations. Further, we employ BFGS-updating of the Hessian of the barrier function to reduce the number of full matrix factorizations needed. It is shown how this can be done in a way retaining the possibility to exploit sparsity in A.

For all problems that we consider, $\mathcal{K}$ will have the form $\mathcal{K} = \mathcal{K}_1 \times \cdots \times \mathcal{K}_K$ where each $\mathcal{K}_j$ is either a *three*-dimensional proper cone or $\mathbb{R}_+$. We assume that a logarithmically homogeneous self-concordant barrier function F for $\mathcal{K}$, its gradient ∇F and its Hessian $\nabla^2 F$ are available and can be efficiently computed for all x in the interior of $\mathcal{K}$. Such a barrier function has many useful properties, some of which we have listed in appendix A.

This paper is organized in two main parts. In the first, which consists of Sections 2 through 4, we discuss theoretical issues, present our algorithm and prove that the method converges in $\mathcal{O}(\sqrt{\nu} \log(1/\epsilon))$ iterations. We state all theoretical results in the main text, emphasizing asymptotic complexity

behavior, but divert all proofs to the appendix to keep the main text clean and free of technical details. Sections 5 and 6 make up the second part. Here, we present and discuss details related to the implementation of our algorithm. We introduce heuristic methods to increase convergence speed and then present an extensive series of computational results substantiating the effectiveness and practical applicability of our algorithm. We finally draw conclusions in Section 7.

2 Homogeneous and self-dual model

If there exist $x \in \mathcal{K}^\circ$ such that $Ax = b$ and $s \in (\mathcal{K}^*)^\circ, y \in \mathbb{R}^m$ such that $A^T y + s = c$, then strong duality holds for the primal-dual problem pair (PD). In this case, any primal optimal x and dual optimal (y, s) must satisfy

$$
\begin{aligned}
Ax - b &= 0 \\
-A^T y - s + c &= 0 \\
x^T s &= 0 \\
x \in \mathcal{K}, \ s \in \mathcal{K}^*, \ y &\in \mathbb{R}^m
\end{aligned}
\tag{1}
$$

We propose solving a homogeneous model of problems (PD). We therefore introduce two extra non-negative scalar variables τ and κ and seek to find x, τ, y, s, κ such that

$$
\left.
\begin{aligned}
\text{minimize} \quad & 0 \\
\text{subject to} \quad & Ax \ -b\tau \quad\quad\ = 0 \\
& -A^T y \quad\quad +c\tau \ -s \ = 0 \\
& b^T y \ -c^T x \quad\quad -\kappa = 0 \\
& (x, \tau) \in \mathcal{K} \times \mathbb{R}_+, \ (s, \kappa) \in \mathcal{K}^* \times \mathbb{R}_+, \ y \in \mathbb{R}^m
\end{aligned}
\right\}
\tag{HSD}
$$

The motivation for doing this is summarized in the following two lemmas.

Lemma 1 *Assume* (x, τ, y, s, κ) *solves* (HSD). *Then*

1. (x, τ, y, s, κ) *is* complementary. *That is:* $x^T s + \tau\kappa = 0$.
2. *If* $\tau > 0$ *then* $(x, y, s)/\tau$ *is optimal for* (PD).
3. *If* $\kappa > 0$ *then one or both of* $b^T y > 0$ *and* $c^T x < 0$ *hold. If the first holds, then* (PD) *is primal-infeasible. If the second holds, then* (PD) *is dual-infeasible.*

Proof See appendix B.1.

Lemma 1 shows that any solution to (HSD) with $\tau + \kappa > 0$ provides either an optimal solution to our original problems (PD) or a certificate of infeasibility of (one of) the original problems. See [11] for further details.

The lemma below shows that there is another desirable feature of the homogeneous model (HSD):

Lemma 2 *The optimization problem* (HSD) *is self-dual.*

Proof See appendix B.2.

Lemma 2 implies that we can apply a primal-dual interior-point algorithm to the problem (HSD) *without* doubling the dimension of the problem — i.e. there is no need to handle and store variables from the dual of (HSD) since they are identical to those of the primal.

The advantages of solving the homogeneous and self-dual model (HSD) therefore include

– It solves the original primal-dual problem pair (PD) without assuming anything concerning the existence of optimal or feasible solutions.
– The dimension of the problem is not essentially larger than that of the original primal-dual pair (PD).
– If the original primal-dual pair (PD) is infeasible, a certificate of this infeasibility is produced.
– As we shall see, the algorithm to solve (HSD) can be initialized in a point not necessarily feasible w.r.t. the linear constraints of (HSD).

3 Nonsymmetric path following

Path following methods are usually motivated by considering a family of barrier problems parametrized by $\mu > 0$:

$$\min_x c^T x + \mu F(x), \quad \text{s.t. } Ax = b, \ x \in \mathcal{K}^\circ. \tag{2}$$

The Karush-Kuhn-Tucker (KKT) conditions of this problem are: If $x \in \mathcal{K}^\circ$ is optimal for (2), then there exist $s \in (\mathcal{K}^*)^\circ$ and $y \in \mathbb{R}^m$ so that

$$\begin{aligned} Ax - b &= 0 \\ -A^T y - s + c &= 0 \\ s + \mu \nabla F(x) &= 0 \\ x \in \mathcal{K}, \ s \in \mathcal{K}^*, \ &y \in \mathbb{R}^m \end{aligned} \tag{3}$$

The points that satisfy (3) are known as the primal-dual *central path*. Let us denote a point in this set by $u(\mu) = (x(\mu), y(\mu), s(\mu))$. It is easy to see that they satisfy $c^T x - b^T y = x^T s = \nu \mu$. The idea of a path-following method is to loosely track $u(\mu)$ towards $u(0)$, thus obtaining a point eventually being approximately optimal for (PD), compare (3) to (1).

Experience shows that it is most efficient to take steps that are combinations of two directions: 1. The direction approximately tangent to the central path (the predictor direction), that is, the direction $u'(\mu)$ and 2. the direction pointing towards the central path as the current iterate may not be exactly on the central path. This correction direction is the Newton step for the equations (3), we will denote it $p(\mu)$. The reason for using this combination is that along u', the function $x^T s$ decreases fast thus bringing the iterate closer to optimality. In order to maintain centrality, i.e. not drift to far away from $u(\mu)$, the correction direction is included as a component in the final search direction.

If the iterate is not exactly on the central path, the search direction $u'(\mu)$ can still be computed so that it is symmetric. Here *symmetric* refers to the search direction (and thus the iterates) being *the same* regardless of whether the roles of the primal and dual problems in (PD) are interchanged [26]. Thus no particular emphasis is put on either the primal or the dual problem, which is a desireable feature of an algorithm. If $\mathcal{K}$ is self-scaled, a symmetric $u'(\mu)$ can be computed using the known *scaling point* [18,19]. If the cone is not self-scaled (nonsymmetric), a symmetric $u'(\mu)$ can be computed by using both the Hessian of the primal and the dual barrier. As discussed in the introduction, this, however, leads to an algorithm that must solve linear systems double the size of those occuring in a symmetric IPM. If the iterate is sufficiently close to the central path, Nesterov showed in [16] that a scaling point determined during a centering (correction) procedure can be used to compute a symmetric search direction $u'(\mu)$.

To present this conceptual algorithm in more detail, we need to introduce a measure of how close an iterate is to the central path. For this purpose, [16] uses the following *proximity measure*:

$$\Psi(x, y, s) = F(x) + F^*(s) + \nu \ln \frac{x^T s}{\nu} + \nu$$

which is ≥ 0 and $= 0$ only if (x, y, s) is on the central path. Here, F^* denotes the dual barrier of F, see appendix A for properties of these two functions.

The general algorithm can then be outlined as below. Assume we start with an initial point $(x, y, s) \in \mathcal{K} \times \mathbb{R}^m \times \mathcal{K}^*$ with $\Psi(x, y, s) < \eta$. Then

---

Repeat

1. $(x, y, s) := (x, y, s) + \alpha u'(\mu)$
   $\mu = x^T s / \mu$.
2. **while** $\Psi(x, y, s) > \eta$
      $(x, y, s) := (x, y, s) + \bar{\alpha} p(\mu)$
   end while

---

where α in step 1 is chosen so that $\Psi(x, y, s) < \beta$ after step 1 and $\bar{\alpha}$ is chosen to be $\lambda/(1 + \lambda)$, where λ is the Newton decrement.

By computing a symmetric $u'(\mu)$ using both the Hessian of the primal and the dual barriers, [20] proves that with appropriate choices of η, β and α, the above algorithm converges in $\mathcal{O}(\sqrt{\nu} \log(1/\epsilon))$ iterations when applied to the homogeneous model (HSD). By computing $u'(\mu)$ using the scaling point found iteratively during step 2, [16] proves the same worst-case complexity estimate when the above algorithm is applied directly to (PD), i.e. not a homogenized version. It uses only the Hessian of the primal barrier thus not suffering from much inflated linear systems. However, a two serious practical drawbacks of the latter method are that it assumes that the original problems are strictly

feasible and that it *requires* a strictly feasible initial primal point to start therefore needing a phase-I method.

Notice that we could, with an appropriate redefinition of A, b, c and K also directly write (HSD) as a purely primal problem similar to the primal part of (PD) and then apply the algorithm from [16]. This would again result in an algorithm with the standard $\mathcal{O}(\sqrt{\nu}\log(1/\epsilon))$ iteration complexity. However, such a redefinition would also result in an algorithm using the Hessians of the primal *and* the dual barrier again making such an algorithm impractical.

Our goal in this paper is to construct an efficient algorithm utilizing the main ideas of [16] and [20], but adapted to be efficient for the homogeneous model (HSD) without using the Hessians of the primal *and* the dual barrier.

We are also aware of the apparent gap between IPM complexity theory and state-of-the-art implementations, see e.g. the introduction of [16] for a discussion about this issue in the case of convex conic programming. In the realm of interior-point algorithms, it is often the case in practice that methods with inferior complexity estimates convincingly outperform algorithms with best-known complexity estimates. See e.g. [1,25] for implementations of such fast algorithms for the case of self-scaled cones. Furthermore, in industry-standard software, heuristic techniques to speed up convergence rates are often employed, although they invalidate the proofs of convergence in the purely theoretical sense. A standard example of such a practice is PDIPMs for linear programming in which it is common to use different primal and dual step lengths. Since a similar discrepancy between theory and practice might be present for the case of a nonsymmetric cone, we expect to be able to improve the performance of our algorithm by employing techniques similar to those used to accelerate the fastest PDIPMs for self-scaled problems.

4 Homogeneous algorithm

4.1 Notation

To simplify notation, we will make the following redefinitions:

New notation	Old meaning	New notation	Old meaning
x	(x, τ)	$\mathcal{K}$	$\mathcal{K} \times \mathbb{R}_+$
s	(s, κ)	$\mathcal{K}^*$	$\mathcal{K}^* \times \mathbb{R}_+$
$F(x)$	$F(x) - \log \tau$	ν	$\nu + 1$
$F^*(s)$	$F^*(s) - \log \kappa$		

We will aggregate all variables as $z = (x, y, s) \in \mathcal{F}$ where $\mathcal{F} := \mathcal{K} \times \mathbb{R}^m \times \mathcal{K}^*$ and define the *complementarity gap* of z by $\mu(z) := (x^T s)/\nu$. We will write $g_x = \nabla F(x)$ and $H_x = \nabla^2 F(x)$ and make use of the following local norms:

$$\|u\|_x = \|H_x^{1/2} u\|, \quad \|s\|_x^* = \|H_x^{-1/2} s\|$$

where $\| \cdot \|$ is a norm. See also appendix A for more properties of these local norms. In our new notation, we can write the homogeneous model simply as

$$G(y, x) - (0, s) = 0, \quad z \in \mathcal{F} \tag{4}$$

where G is the skew-symmetric matrix

$$G := \begin{pmatrix} 0 & A & -b \\ -A^T & 0 & c \\ b^T & -c^T & 0 \end{pmatrix} \tag{5}$$

4.2 The central path in the homogeneous model

First, let us define

$$\psi(x, s, t) := s + t g_x \overset{(18)}{=} s - t H_x x. \tag{6}$$

We initialize our algorithm in $z^0 \in \mathcal{F}$. Denote $\mu^0 = \mu(z^0)$. Parametrized by $\gamma \in [0, 1]$, we define the central path of the homogenized problem (4) by the points z_γ that satisfy

$$G(y_\gamma, x_\gamma) - (0, s_\gamma) = \gamma \left(G(y^0, x^0) - (0, s^0) \right) \tag{7}$$
$$\psi(x_\gamma, s_\gamma, \gamma \mu^0) = 0 \tag{8}$$

In the homogeneous model, the central path connects the point z^0 (at $\gamma = 1$) with a solution of the problem (4) as $\gamma \to 0$. Therefore, the main idea of the algorithm, as in other path-following algorithms, is to approximately track the central path towards a solution.

For a fixed parameter $\eta \in [0, 1]$ to be chosen later, we will be using the set

$$\mathcal{N}(\eta) = \{ z = (x, y, s) \in \mathcal{F} : \quad \| \psi(x, s, \mu(z)) \|_x^* \le \eta \mu(z) \} \tag{9}$$

which, in view of (8), can be considered a measure of the proximity to the feasible central path — that is, the path that would arise from using z^0 in (7)–(8) such that $G(y^0, x^0) - (0, s^0) = 0$.

In the case of LP with the usual barrier $F(x) = - \sum_j \log x_j$, we remark that equation (8) is the same as the familiar $Xs = \gamma \mu^0 e$ where $X = \text{diag}(x)$ and $e = (1, \ldots, 1)$. Similarly, the definition of $\mathcal{N}(\eta)$ in (9) reduces to $\|Xs - \mu e\| \le \eta \mu(z)$.

4.3 Prediction

The direction z' tangent to the central path (also called the predictor direction) is determined by differentiating (7)–(8) with respect to γ. For equation (8), this yields

$$s'_\gamma = -\mu^0 g_{x_\gamma} - \gamma \mu^0 H_{x_\gamma} x'_\gamma$$

and by (8), we have $\gamma^{-1}s_\gamma = -\mu^0 g_{x_\gamma}$, which we insert and get

$$s_\gamma' + \gamma\mu^0 H_{x_\gamma} x_\gamma' = \gamma^{-1} s_\gamma.$$

The same operation on (7) gives the equations defining the direction z':

$$G(y', x') - (0, s') = -\left(G(y, x) - (0, s)\right) \tag{10}$$
$$s' + \mu(z) H_x x' = -s \tag{11}$$

where we have dropped the argument γ for readability and put $\mu(z)/\mu^0 = \gamma$. Notice also that we have rescaled the equations by $-\gamma$ to make the notation consistent with the general IPM litterature. This does not change the direction z', only its magnitude. Determining the direction z' thus amounts to solving the system of linear equations (10)–(11).

In the rest of this section, we will use the notation

$$z^+ = (x^+, y^+, s^+) = (x + \alpha x', y + \alpha y', s + \alpha s') = z + \alpha z'$$
$$\psi = \psi(x, s, \mu(z))$$
$$\psi^+ = \psi(x^+, s^+, \mu(z^+))$$
$$z' = \text{solution of (10)–(11)}.$$

The next lemma explains how the linear residuals and the complementarity gap are reduced along the predictor direction.

Lemma 3 *The direction z' satisfies*

$$G(y^+, x^+) - (0, s^+) = (1 - \alpha)\left(G(y, x) - (0, s)\right)$$
$$\mu(z^+) = (1 - \alpha)\mu(z) + (1 - \alpha)\alpha\nu^{-1}\psi^T x'$$

Proof See appendix C.1.

The first relation shows that the linear residuals are reduced by the factor $1-\alpha$ along the direction z'. The complimentarity gap μ is reduced in a slightly more complicated way depending on the vector ψ. If z is precisely on the central path, $\psi = 0$, so $\mu(z^+) = (1 - \alpha)\mu(z)$ and also the complementarity gap is reduced by the factor $1-\alpha$. As we shall see, we can, similarly to other interior-point algorithms, choose $\alpha = \Omega(1/\sqrt{\nu})$ so that $\mu(z^+) \leq (1 - \Omega(1/\sqrt{\nu}))\mu(z)$. Here, we use the "big-$\Omega$"-notation meaning that α is asymptotically bounded below by $1/\sqrt{\nu}$ times a positive (possibly small) constant as $\nu \to \infty$.

Lemma 4 *Assume $z \in \mathcal{N}(\eta)$. Then we can choose $\alpha = \Omega(1/\sqrt{\nu})$ so that $x^+ \in \mathcal{K}$ and $s^+ \in \mathcal{K}^*$*

Proof See appendix C.3.

Lemma 5 *Assume $z \in \mathcal{N}(\eta)$. If $\eta \leq 1/6$, then we can choose $\alpha = \Omega(1/\sqrt{\nu})$ so that*

$$z^+ \in \mathcal{N}(2\eta)$$

Proof See appendix C.4.

Algorithm 1 Nonsymmetric Predictor-Corrector Algorithm

Input: Barrier function F, $\eta \leq 1/6$, and initial point $z \in \mathcal{F} \cap \mathcal{N}(\eta)$.

  $\bar{\alpha} := 1/84$

  Repeat

    Set $\mu := \mu(z)$

    Stopping

      If stopping criteria satisfied: **terminate**.

    Prediction

      Solve (10)–(11) for z'

      Choose largest α so that $z + \alpha z' \in \mathcal{F} \cap \mathcal{N}(2\eta)$

      Set $z := z + \alpha z'$ and $\mu = \mu(z)$.

    Correction

      Solve (13)–(14) for $\bar{z}$

      Set $z := z + \bar{\alpha}\bar{z}$

      Solve (13)–(14) for $\bar{z}$

      Set $z := z + \bar{\alpha}\bar{z}$

  End

4.4 Correction phase

Given a point $z = (x, y, s) \in \mathcal{N}(2\eta)$, the goal of the correction phase is to find a new point $z^+ = (x^+, y^+, s^+)$ which is close to the central path. That is, we want to find z^+ so that $\|\psi(x^+, s^+, \mu(z))\|_x^* \leq \eta\mu(z)$. At the same time, we would like that the linear residuals remain unchanged. To achieve this, we apply Newton's method to the equation

$$\psi(x, s, \mu(z^+)) = 0 \tag{12}$$
$$G(y, x) - (0, s) = G(y^+, x^+) - (0, s^+)$$

The Newton step $(\bar{x}, \bar{y}, \bar{s})$ for these of equations is

$$G(\bar{y}, \bar{x}) - (0, \bar{s}) = 0 \tag{13}$$
$$\bar{s} - \mu(z)H_x\bar{x} = -\psi(x, s, \mu) \tag{14}$$

We then apply

$$z := z + \bar{\alpha}\bar{z} \tag{15}$$

recursively until $\|\psi(x, s, \mu(z^+))\| \leq \eta\mu(z^+)$. The following Lemma shows that this process terminates quickly.

Lemma 6 *If $\eta \leq 1/6$, then the correction process* (15) *terminates in at most two steps.*

Proof See appendix D.

Algorithm 2 Aggresive step implementation

Input: Barrier function F, $0 < \eta \leq \beta < 1$, and initial point $z \in \mathcal{F} \cap \mathcal{N}(\eta)$.

  Repeat
    Set $\mu := \mu(z)$
    Stopping
      If stopping criteria satisfied: **terminate**.
    Prediction
      Solve (10)–(11) for z'
      Choose largest α so that $z + \alpha z' \in \mathcal{F} \cap \mathcal{N}(\beta)$
      Set $z := z + \alpha z'$ and $\mu = \mu(z)$.
    Correction
      Repeat
        Solve (13)–(14) for $\bar{z}$
        Choose $\bar{\alpha}$ to approximately minimize $\|\psi\|_x^*$ along $\bar{z}$
        Set $z := z + \bar{\alpha}\bar{z}$
      Until $z \in \mathcal{F} \cap \mathcal{N}(\eta)$.
  End

4.5 Complexity of algorithm

From (12), we see that the linear residuals do not change during the correction phase. We can now gather the pieces and prove the following theorem.

Theorem 1 *Algorithm 1 terminates with a point $z = (x, y, s)$ that satisfies*

$$\mu(z) \leq \epsilon\mu(z^0) \quad and \quad \|G(y, x) - (0, s)\| \leq \epsilon\|G(y^0, x^0) - (0, s^0)\|$$

in no more than $\mathcal{O}\left(\sqrt{\nu}\log\left(1/\epsilon\right)\right)$ iterations.

Proof See appendix E.

We remark that we have emphasized only the asymptotic analysis. The technical details can be found in the proofs in the appendix. In several places, it may be possible to improve the constants in the leading terms but as the above analysis serves only to demonstrate asymptotic worst-case behavior, this is of minor importance. As discussed in the introduction, the gap between worst-case complexity analysis and actual performace of interior-point methods is often significant. In order for an interior-point method to be practical and competitive, the implementation must deviate somewhat from the pure theoretical algorithm. In the next section, we describe how such an efficient algorithm can be implemented.

5 Implementation

Our implementation is outlined in Algorithm 2. As is common practice in implementations of interior-point methods, we allow for a much longer prediction step, for example $\beta \geq 0.80$. This leads to faster convergence once we get close to the optimal point. Indeed we do observe what appears to be superlinear convergence in this region.

It should be noted, however, that we can no longer be certain that two correction steps will be enough to reach a sufficiently centered point. Therefore, we continue taking correction steps until the centrality condition $\|\psi\|_x^* \leq \eta\mu$ is satisfied. As the computational experiments later show, for the problems we have solved, rarely more than one or two correction steps are needed. We can further reduce the cost of the correction phase by using quasi-Newton updating as we explain in the next section.

5.1 Quasi-Newton updating in the correction phase

Solving either for a prediction or a correction step requires the factorization of the sparse $n \times n$ matrix H_x and of the possibly sparse $m \times m$ matrix $Q = AH_x^{-1}A^T$. To reduce the total number of factorizations needed in the correction phase, we suggest taking J quasi-Newton steps for each normal correction step.

Let us show how this can be done computationally efficient without destroying sparsity in the KKT-system, which is an essential requirement in practical applications.

Let B and M denote the current quasi-Newton approximation of the *inverses* of H and Q respectively. Conceptually, we update B to B^+ using BFGS updating (see e.g. [22]), a rank-2 updating scheme: $B^+ = B + k^{(v)}vv^T + k^{(w)}ww^T$. In order to keep the ability to exploit sparsity of A and Q, we do not actually store B or M but simply the Cholesky factors of the most recent H and Q and the sequence of BFGS update vectors. More specifically, for $q \leq J$, let $B^{(q)}$ be the q'th update of H^{-1}, i.e.

$$B^{(q)} = C^{-1}C^{-T} + \Psi\Lambda\Psi^T$$

where $\Psi = [v^{(1)}, \ldots, v^{(q)}, w^{(1)}, \ldots, w^{(q)}], \Lambda = \text{diag}(k_1^{(v)}, \ldots, k_q^{(v)}, k_1^{(w)}, \ldots, k_q^{(w)})$. Then we compute products such as $B^{(q)}r$ by means of

$$B^{(q)}r = C^{-1}(C^{-T}r) + \Psi\left(\Lambda(\Psi^Tr)\right).$$

For M, the situation is similar:

$$\begin{aligned}
M^{(q)} &= \left(AB^{(q)}A^T\right)^{-1} \\
&= \left(A(H^{-1} + \Psi\Lambda\Psi^T)A^T\right)^{-1} \\
&= \left(Q + \Phi\Lambda\Phi^T\right)^{-1}
\end{aligned}$$

where $\Phi = A\Psi$. By the Sherman-Morrison-Woodbury formula, we get

$$M^{(q)} = Q^{-1} - Q^{-1}\Phi\left(\Lambda^{-1} + \Phi^TQ^{-1}\Phi\right)^{-1}\Phi^TQ^{-1}.$$

We can thus compute products like $M^{(q)}r$ by

$$
\begin{aligned}
M^{(q)}r &= Q^{-1}\left(I - \Phi\left(\Lambda^{-1} + \Phi^T Q^{-1}\Phi\right)^{-1}\Phi^T Q^{-1}\right)r \\
&= D^{-1}D^{-T}\left(r - \Phi\left(\Lambda^{-1} + \Phi^T D^{-1}D^{-T}\Phi\right)^{-1}\Phi^T D^{-1}D^{-T}r\right)
\end{aligned}
$$

where we remark that 1) only two columns are added to Φ in each iteration so that only two new backsubstitutions in the operation $D^{-T}\Phi$ are needed, 2) Λ is diagonal and thus cheap to invert and 3) the matrix $\left(\Lambda^{-1} + \Phi^T D^{-1}D^{-T}\Phi\right)$ is only of size $2q \times 2q$ and is therefore also cheap to invert.

We then alternate between taking J BFGS steps and one full Newton correction step, starting with BFGS steps and terminate when $\|\psi\|_x^* \leq \eta\mu$. The resulting BFGS search direction is a descent direction for the function $\|\psi\|$, so by using a backtracking line search along these directions, we can not make the objective worse by proceding in this way. On the other hand, we have no theoretical guarantee that BFGS steps improve the objective value. However, as the computational experiments will demonstrate, it is often the case that enough centrality can be achieved after just a few BFGS steps.

The norm $\|v\|_x^*$ is computed as $(v^T H_x^{-1} v)^{1/2}$. Computing this number requires the evaluation and factorization of H_x. But since H_x is blockdiagonal, this operation is cheap. In fact, it is possible simply to analytically compute H_x^{-1} at each x, since H_x is block diagonal with block sizes 3×3.

We finally remark that whether or not it is beneficial to take BFGS steps, and if it is, how many should be taken, depends on the cost of building and Cholesky factorizing $AH_x^{-1}A^T$ relative to the cost of subsequent backsubstitutions, of which the needed amount is increased if BFGS steps are used. This ratio depends on the dimension and sparsity pattern of A — quantities about which we know nothing beforehand. However, since the dimension and sparsity pattern of $AH_x^{-1}A^T$ do not vary with x, it is possible to determine this ratio at initialization time. Thus we can determine an upper bound on J before the main loop of the algorithm.

5.2 Higher order predictor direction

It is well known that the Mehrotra second order correction [12] term significantly improves performance of interior-point methods for symmetric cones. This technique is used in virtually all competitive industry standard interior-point implementations. Hoping to achieve a similar improvement in performace, we suggest computing a higher order prediction step as described in the following.

Let us denote the central path point with complementarity gap μ by $z(\mu)$, which corresponds to $\mu = \gamma\mu^0$ in equations (7)–(8). By an appropriate definition of a matrix $K(z)$ and a vector $u(z)$, dependent on the current iterate $z = (x, y, s)$, it is clear that the equations (10)–(11) defining z' can be written

$$
K(z)z'(\mu) = u(z) \quad \text{or} \quad z'(\mu) = K(z)^{-1}u(z) =: f(z).
$$

The central path is thus the solution of the ordinary differential equation defined by $z'(\mu) = f(z)$. A step in the predictor direction, i.e. the direction z', is then the same as taking one Euler step for this ODE. We can obtain a direction that contains, for example, second order information by computing a stage-2 Runge-Kutta direction d_2, remembering that each evaluation of f requires solving a system of the type $Kz' = u$. Such a direction is defined by

$$d_2 = h \left(1 - \frac{1}{2\theta} \right) f(z) + h \frac{1}{2\theta} f(\zeta)$$
$$\zeta = (\zeta_x, \zeta_y, \zeta_s) = z(\mu) + \theta h f(z)$$

where h is the stepsize possible in the direction $f(z)$ and $\theta \in (0, 1]$ is a parameter. The choices $\theta = 1/2$ and $\theta = 1$ correspond to the classical midpoint and trapezoidal rules respectively [6].

Our experience shows that this approach reduces the *total* number of iterations as well as the number of factorizations needed to reach an optimal solution, even though two factorizations are needed to compute d_2.

We can, however, restrict ourselves to just one factorization by using in place of H_{ζ_x} the BFGS update of H_x. In section 5.1, we showed how to implement such a procedure efficiently.

5.3 Initial point

The initial point $z^0 = (x^0, y^0, s^0)$ is required to satisfy $z^0 \in \mathcal{F} \cap \mathcal{N}(\eta)$. We therefore choose some $x^0 \in \mathcal{K}^\circ$ and set $s^0 = -g_{x^0}$. We then get

$$\nu \mu(z^0) = (x^0)^T s^0 = -(x^0)^T g_{x^0} \overset{(19)}{=} \nu$$

and hence $\mu(z^0) = 1$. Therefore, this z^0 is exactly on the central path, i.e. $z^0 \in \mathcal{N}(0) \subset \mathcal{N}(\eta)$.

5.4 Termination

A point (x, y, s) that satisfies the bounds in Theorem 1 solves to ϵ-accuracy the homogeneous model (HSD). However, we are interesting in either a certificate of infeasibility or a solution of (PD). Therefore, we need to use stopping criteria able to detect one of these two situations. In this section we therefore briefly return to the "extended" notation used prior to Section 4.1.

Assume (x, τ, y, s, κ) is the current iterate and consider the following inequalities:

$$\|Ax - \tau b\|_\infty \le \epsilon \cdot \max\{1, \|[A, b]\|_\infty\} \tag{P}$$

$$\|A^T y + s - c\tau\|_\infty \le \epsilon \cdot \max\{1, \|[A^T, I, -c]\|_\infty\} \tag{D}$$

$$|-c^T x + b^T y - \kappa| \le \epsilon \cdot \max\{1, \|[-c^T, b^T, 1]\|_\infty\} \tag{G}$$

$$|c^T x/\tau - b^T y/\tau| \le \epsilon \cdot (1 + |b^T y/\tau|) \tag{A}$$

$$\tau \le \epsilon \cdot 10^{-2} \cdot \max\{1, \kappa\} \tag{T}$$

$$\tau \le \epsilon \cdot 10^{-2} \cdot \min\{1, \kappa\} \tag{K}$$

$$\mu \le \epsilon \cdot 10^{-2} \cdot \mu^0 \tag{M}$$

We then terminate and conclude as follows:

(OPT)	$(P) \wedge (D) \wedge (A) \Rightarrow$	Feasible and optimal solution found
(INFEAS)	$(P) \wedge (D) \wedge (G) \wedge (T) \Rightarrow$	Problem primal or dual infeasible
(ILLP)	$(K) \wedge (M) \Rightarrow$	Problem ill-posed

In case (OPT), the optimal solution $(x, y, s)/\tau$ is returned. If we find (INFEAS), the problem is dual infeasible if $c^T x < 0$ and primal infeasible if $b^T y > 0$. The number $\epsilon > 0$ is a user-specified tolerance.

6 Computational experiments

In this section we present results from running our algorithm, which we will denote by NPC, on different test problems. We first introduce the nonsymmetric cones needed for our test problems and then present the test problems. Finally, we include tables with numerical results and discussion.

6.1 Two three-dimensional nonsymmetric cones

In the rest of this paper, we will be considering problems involving the following two nonsymmetric convex cones, both three dimensional.

The three-dimensional *exponential cone* is defined by

$$\mathcal{K}_{\exp} = \text{closure}\{(x_1; x_2; x_3) \in \mathbb{R} \times \mathbb{R}_+ \times \mathbb{R}_{++} : \exp(x_1/x_3) \le x_2/x_3\}$$

for which we are using the barrier function

$$F_{\exp}(x) = -\log(x_3 \log(x_2/x_3) - x_1) - \log x_2 - \log x_3$$

with barrier parameter $\nu = 3$.

The three-dimensional *power cone* is defined by

$$\mathcal{K}_\alpha = \{(x_1; x_2; x_3) \in \mathbb{R} \times \mathbb{R}_+^2 : |x_1| \le x_2^\alpha x_3^{1-\alpha}\}$$

where $\alpha \in [0,1]$ is a parameter. Notice that $\mathcal{K}_{1/2}$ is the standard rotated quadratic cone. For all other $\alpha \in (0,1)$, $\mathcal{K}_\alpha$ is not symmetric. In [7], it was proved that the function

$$F_\alpha(x) = -\log\left(x_2^{2\alpha} x_3^{2-2\alpha} - x_1^2\right) - (1-\alpha)\log x_2 - \alpha \log x_3$$

is a logarithmically homogeneous self-concordant barrier with parameter $\nu = 3$ for $\mathcal{K}_\alpha$. It is this barrier function we are using in our experiments. Nesterov proposed in [16] a barrier function for the three-dimensional power cone with parameter $\nu = 4$. Our computational experience shows that F_α is better in practice which is in accordance with theory.

6.2 Test problems

In this section, e will denote the vector of all ones. The dimension of e will be clear from the context.

6.2.1 p-cone problem

Given $A \in \mathbb{R}^{M \times N}$ and $b \in \mathbb{R}^M$, the p-cone problem is the problem

$$\min_x \quad \|x\|_p, \quad \text{s.t. } Ax = b.$$

In [15], it is shown that this is equivalent to

$$\begin{aligned}
\min_{x,y,t} \quad & t \\
\text{s.t.} \quad & Ax = b, \quad e^T y = t \\
& (x_j; y_j; t) \in K_{(1/p)}, \quad j = 1, \ldots, M.
\end{aligned}$$

6.2.2 Facility location problem

Given M points (locations) in $\mathbb{R}^N$: $C^{(j)}, j = 1, \ldots, M$, we want to find the point z with the minimal sum of weighted distances to the locations $C^{(j)}$ measured in p_j-norms, $p_j \geq 1$. That is

$$\min_z \quad \sum_{j=1}^M a_j \|z - C^{(j)}\|_{p_j} \tag{16}$$

where $a_j \geq 0$ are the weights. We can then formulate (16) in conic form:

$$\begin{aligned}
\min_{z^+, z^-, v, w, u} \quad & \sum_{j=1}^M a_j u_1^{(j)} \\
\text{s.t.} \quad & v^{(j)} = z^+ - z^- - C^{(j)} & j = 1, \ldots, M \\
& e^T w^{(j)} = u_1^{(j)}, \quad u_1^{(j)} = u_2^{(j)} = \cdots = u_N^{(j)} & j = 1, \ldots, M \\
& (v_i^{(j)}; w_i^{(j)}; u_i^{(j)}) \in \mathcal{K}_{1/p_j} & j = 1, \ldots, M, \ i = 1, \ldots, N \\
& z^+ \geq 0, \quad z^- \geq 0
\end{aligned}$$

6.2.3 Geometric programming

This is a problem of the type

$$
\begin{aligned}
\min_{\boldsymbol{x}} \quad & f^{(0)}(\boldsymbol{x}) \\
\text{s.t.} \quad & g^{(j)}(\boldsymbol{x}) = 1, \quad j = 1, \ldots, M \\
& f^{(j)}(\boldsymbol{x}) \leq 1, \quad j = 1, \ldots, P
\end{aligned}
$$

where $g^{(j)}$ are monomials and $f^{(j)}$ are posynomials:

$$
g(\boldsymbol{x}) = k_j \boldsymbol{x}^{\boldsymbol{b}^{(j)}}, \quad f^{(j)}(\boldsymbol{x}) = \sum_{i=1}^{N_j} d_i \boldsymbol{x}^{\boldsymbol{a}_i^{(j)}}
$$

where we have used the notation

$$
\boldsymbol{x}^{\boldsymbol{v}} := \prod_{i=1}^{n} x_i^{v_i}, \quad x_i > 0.
$$

With the j'th posynomial $f^{(j)}$, we associate

- the matrix $\boldsymbol{A}^{(j)} := \left(\boldsymbol{a}_1^{(j)}, \boldsymbol{a}_2^{(j)}, \ldots, \boldsymbol{a}_{N_j}^{(j)} \right)^T \in \mathbb{R}^{N_j \times N}$,
- the vector $\boldsymbol{d}^{(j)} = (d_1^{(j)}, \ldots, d_{N_j}^{(j)})^T \in \mathbb{R}^{N_j \times 1}$ and
- the vector $\boldsymbol{c}^{(j)} = \log\left(\boldsymbol{d}^{(j)}\right) = \left(\log\left(d_1\right), \ldots, \log\left(d_{N_j}\right)\right)^T \in \mathbb{R}^{N_j \times 1}$

Similarly, we associate with the j'th monomial $g^{(j)}$

- the vector $\boldsymbol{b}^{(j)}$
- the scalar $k^{(j)}$
- the scalar $h^{(j)} = \log\left(k^{(j)}\right)$

Using the change of variables $u_i = \log\left(x_i\right) \Leftrightarrow x_i = \exp(u_i)$ for all i, we can write the problem in conic form:

$$
\begin{aligned}
\min_{\boldsymbol{u}_+, \boldsymbol{u}_-, \boldsymbol{w}, \boldsymbol{v}, \boldsymbol{y}, t^{(0)}} \quad & t^{(0)} \\
\text{s.t.:} \quad & \boldsymbol{B}(\boldsymbol{u}_+ - \boldsymbol{u}_-) + \boldsymbol{h} = 0 \\
& \boldsymbol{w}^{(j)} = \boldsymbol{A}^{(j)}(\boldsymbol{u}_+ - \boldsymbol{u}_-) + \boldsymbol{c}^{(j)} && j = 0, \ldots, P \\
& \boldsymbol{e}^T \boldsymbol{v}^{(j)} = t^{(j)}, \quad \boldsymbol{y}^{(j)} = \boldsymbol{e} && j = 0, \ldots, P \\
& \boldsymbol{u}_+, \boldsymbol{u}_-, t^{(0)} \geq 0 \\
& \left(w_i^{(j)}; v_i^{(j)}; y_i^{(j)} \right) \in K_{\exp} && j = 0, \ldots, P, \ i = 1, \ldots, N_j
\end{aligned}
$$

where $\boldsymbol{h} = (h^{(1)}, \ldots, h^{(M)})^T \in \mathbb{R}^{M \times 1}$ and $\boldsymbol{B} = \left(\boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(M)}\right)^T \in \mathbb{R}^{M \times N}$.

Table 1 *Parameters used in computational experiments.*

Parameter	J	θ	η	β	ϵ
Value	3	0.70	0.50	0.80	10^{-6}

6.2.4 Entropy maximization

Given $A \in \mathbb{R}^{M \times N}$, $b \in \mathbb{R}^M$ and $d \in \mathbb{R}^N_+$, the entropy maximization problem is

$$\min_{x} \quad \sum_{j=1}^{N} d_j x_j \log x_j$$
$$\text{s.t.} \quad Ax = b$$
$$x_j \geq 0, \quad j = 1, \ldots, N$$

which can be formulated as

$$\min_{x,u} \quad -d^T u$$
$$\text{s.t.} \quad Ax = b, \ v = e$$
$$(u_j; v_j; x_j) \in \mathcal{K}_{\exp}, \quad j = 1, \ldots, N.$$

6.3 Computational results

The remaining tables in this section show the number of iterations (it), the *total* number of factorizations made (ch), the average number of *full* correction steps per iteration (ce) and the termination status (st). `opt` means that an optimal solution was found and `ipr/idu` means a primal/dual infeasibility certificate was found. For all computational experiments, we used the parameters displayed in Table 1.

For entropy maximization problems and geometric programs, we compare our algorithm to the purpose-built solvers in Mosek [13]. For p-cone problems, we compare our algorithm to `SeDuMi` (see [24]) when called through `CVX` (see [8]). We intentionally compare only the number of Choleschy factorizations performed by each algorithm. This is to eliminate from the comparisons the cpu-time consumed by software overhead. Therefore, it is reasonable to measure only the dominating operations, i.e. the Choleschy factorizations.

6.3.1 p-cone problems

Table 2 shows results from solving a series of p-cone problems. The data A and b are from the NETLIB collection of linear programs. We see that NPC performs very well compared to `SeDuMi`. CVX solves the problem by approximating the original p-cone problem by an approximately equivalent self-scaled problem. The resulting self-scaled problem is then solved using `SeDuMi`. As

Table 2 *Computational results for p-cone problems. Data $A \in \mathbb{R}^{M \times N}$ and b from* NETLIB. *$sp(A)$ denotes the sparsity of A.*

Problem		NPC						CVX/SeDuMi			
name & size	p	m	n	it	ch	ce	st	m	n	ch	st
`bandm`	1.13	777	1416	9	19	1.1	opt	6913	14632	21	opt
$M = 305$	1.57	777	1416	11	23	1.1	opt	8801	18408	26	opt
$N = 472$	2.09	777	1416	14	29	1.1	opt	9745	20296	27	opt
$sp(A) = 1.73\%$	4.71	777	1416	23	37	0.6	opt	10689	22184	26	opt
	7.39	777	1416	24	43	0.8	opt	11633	24072	26	opt
`blend`	1.13	188	342	9	19	1.1	opt	1670	3534	21	opt
$M = 74$	1.57	188	342	9	20	1.2	opt	2126	4446	22	opt
$N = 114$	2.09	188	342	9	16	0.8	opt	2354	4902	20	opt
$sp(A) = 6.19\%$	4.71	188	342	11	19	0.7	opt	2582	5358	20	opt
	7.39	188	342	13	21	0.6	opt	2810	5814	21	opt
`bore3d`	1.13	565	1002	7	8	0.1	opt	4907	10354	6	opt
$M = 231$	1.57	565	1002	7	8	0.1	opt	6243	13026	6	opt
$N = 334$	2.09	565	1002	7	8	0.1	opt	6911	14362	6	opt
$sp(A) = 1.87\%$	4.71	565	1002	7	8	0.1	opt	7579	15698	6	opt
	7.39	565	1002	7	8	0.1	opt	8247	17034	6	opt
`scagr25`	1.13	1142	2013	11	18	0.6	opt	9865	20801	21	opt
$M = 471$	1.57	1142	2013	10	21	1.1	opt	12549	26169	21	opt
$N = 671$	2.09	1142	2013	11	21	0.9	opt	13891	28853	20	opt
$sp(A) = 0.55\%$	4.71	1142	2013	10	16	0.6	opt	15233	31537	19	opt
	7.39	1142	2013	10	21	1.1	opt	16575	34221	17	opt
`sctap1`	1.13	960	1980	10	21	1.1	opt	9540	20460	22	opt
$M = 300$	1.57	960	1980	10	20	1.0	opt	12180	25740	25	opt
$N = 660$	2.09	960	1980	9	22	1.4	opt	13500	28380	21	opt
$sp(A) = 0.95\%$	4.71	960	1980	9	23	1.6	opt	14820	31020	18	opt
	7.39	960	1980	9	20	1.2	opt	16140	33660	21	opt
`share1b`	1.13	370	759	10	21	1.1	opt	3659	7843	21	opt
$M = 117$	1.57	370	759	12	20	0.7	opt	4671	9867	26	opt
$N = 253$	2.09	370	759	13	24	0.8	opt	5177	10879	24	opt
$sp(A) = 3.98\%$	4.71	370	759	13	23	0.8	opt	5683	11891	23	opt
	7.39	370	759	13	24	0.8	opt	6189	12903	24	opt
`share2b`	1.13	258	486	9	20	1.2	opt	2364	5022	19	opt
$M = 96$	1.57	258	486	9	18	1.0	opt	3012	6318	20	opt
$N = 162$	2.09	258	486	9	16	0.8	opt	3336	6966	20	opt
$sp(A) = 5.00\%$	4.71	258	486	11	22	1.0	opt	3660	7614	20	opt
	7.39	258	486	11	20	0.8	opt	3984	8262	20	opt
`stocfor1`	1.13	282	495	9	16	0.8	opt	2427	5115	19	opt
$M = 117$	1.57	282	495	8	17	1.1	opt	3087	6435	20	opt
$N = 165$	2.09	282	495	9	19	1.1	opt	3417	7095	22	opt
$sp(A) = 2.60\%$	4.71	282	495	18	30	0.7	opt	3747	7755	25	opt
	7.39	282	495	22	29	0.3	opt	4077	8415	26	opt

discussed in the introduction, this modelling of a nonsymmetric problem by symmetric cones requires the introduction of extra variables and constraints. The table shows for each of the two solution methods, the number of rows m and columns n of the final linear constraint matrix (corresponding to A in (PD)). These results clearly demonstrate the advantage of modelling this inherently nonsymmetric problem (the p-norm is not a self-dual norm when $p \neq 2$) directly by using a nonsymmetric cone. As seen from the table, the size

of the problem built by `CVX` is much greater, in some instances by as much as 17 times, than the size of the problem solved by NPC. Notice also that the latter problem, unlike the first, is independent of p.

In terms of iterations, NPC uses about 40% less than `SeDuMi`. The total number of factorizations for the two methods is about the same. However, as described above, `SeDuMi` factorizes much larger matrices. Therefore we may conclude for these problems, that the direct modelling method coupled with a nonsymmetric solver like NPC is clearly superior to `CVX/SeDuMi`.

6.3.2 Facility location problems

Table 3 shows the performances of our algorithm when run on random instances of the facility location problem. For each pair (N, M), we generated 10 instances each with $C^{(j)}$ chosen at random from the standard normal distribution. For each instance, M different p_j were chosen as the maximum of 1.0 and a sample from a normal distribution with mean 2.0 and variance 0.25. The a_j were chosen randomly from a uniform distribution on $[0, 1]$. The column labelled $\bar{p}$ shows the number $M^{-1} \sum_{j=1}^{M} p_j$ averaged over the 10 instances. This number should be close to 2.0.

We see that our algorithm uses in the region 10–20 iterations and the number of Cholesky factorizations never exceeds 32. On average slightly more than 0.50 full centering steps are needed in each iteration. These results can be loosely compared with the computational results in [7, Table 4.1, page 142]. There, a dual variant of the algorithm of [16] is used to solve the same kind of problem. Overall, our algorithm performs better, both in terms of iterations and factorizations.

Table 3 *Results for facility location problems. The algorithm always terminated after reaching optimality as all problem instances were feasible by construction.*

	Problem				NPC	
N	M	ν	$\bar{p}$	it	ch	ce
3	4	44	2.03	11.1	18.2	0.65
3	8	88	1.96	14.1	22.3	0.61
10	4	128	2.07	13.2	20.1	0.54
3	12	132	1.93	15.3	23.4	0.56
3	20	220	2.09	17.1	27.5	0.64
19	4	236	2.00	13.8	21.0	0.54
10	8	256	1.98	15.6	23.4	0.51
10	12	384	2.06	16.0	25.1	0.58
32	4	392	2.03	13.4	20.9	0.56
19	8	472	1.98	15.2	23.1	0.53
10	20	640	1.99	18.7	30.5	0.66
19	12	708	1.99	15.3	25.9	0.70
32	8	784	2.04	14.0	23.3	0.67
32	12	1176	2.05	16.4	27.0	0.65
19	20	1180	2.01	19.7	30.5	0.60
32	20	1960	1.98	17.7	31.5	0.79

Table 4 *Results for geometric programs.*

name	n	dod	it	ch	ce	st	ch	st
	Problem			NPC			mskgpopt	
beck751	7	10	16	30	0.9	opt	18	opt
beck752	7	10	15	29	0.9	opt	28	opt
beck753	7	10	13	27	1.1	opt	10	opt
bss2	2	1	9	13	0.4	opt	5	opt
car	37	104	15	28	0.9	opt	46	opt
demb761	11	19	12	22	0.8	ipr	10	opt
demb762	11	19	9	19	1.1	opt	11	opt
demb763	11	19	10	20	1.0	opt	11	opt
demb781	2	1	7	10	0.4	opt	7	opt
fang88	11	16	9	18	1.0	opt	11	opt
fiac81a	22	50	11	22	1.0	opt	16	opt
fiac81b	10	9	12	21	0.8	ipr	10	opt
gptest	4	1	8	12	0.5	opt	5	opt
jha88	30	274	17	34	1.0	opt	13	opt
mra01	61	844	16	30	0.9	opt	58	opt
mra02	126	3494	30	57	0.9	opt	53	opt
rijc781	4	1	8	12	0.5	opt	5	opt
rijc782	3	5	10	18	0.8	opt	8	opt
rijc783	4	7	12	23	0.9	opt	7	opt
rijc784	4	3	13	19	0.5	rnd	6	opt
rijc785	8	3	9	16	0.8	opt	9	opt
rijc786	8	3	9	16	0.8	opt	6	opt
rijc787	7	40	12	23	0.9	opt	36	opt

6.3.3 Geometric programs

Table 4 shows results from applying our algorithms to a set of geometric programs supplied to us by MOSEK. The column labelled `dod` denotes the *degree of difficulty* of the problem [5]. For a particular problem instance j, let $I_j^{\mathcal{A}}$ and $C_j^{\mathcal{A}}$ be the number of iterations and Choleschy factorization respectively used by algorithm $\mathcal{A}$ to solve instance j and let us define the ratio of sums $\mathcal{S} = (\sum_j C_j^{\mathrm{NPC}})/(\sum_j C_j^{\mathrm{NPC}})$. Further let $\mathcal{R}_j^{\mathrm{it}} = I_j^{\mathrm{NPC}}/I_j^{\mathrm{MOSEK}}$ and $\mathcal{R}_j^{\mathrm{ch}} = C_j^{\mathrm{NPC}}/C_j^{\mathrm{MOSEK}}$. If we let an overbar denote arithmetic mean and a tilde denote geometric mean over all j, we then find

$$(S, \ \overline{\mathcal{R}^{\mathrm{it}}}, \ \overline{\mathcal{R}^{\mathrm{ch}}}, \ \widetilde{\mathcal{R}^{\mathrm{it}}}, \ \widetilde{\mathcal{R}^{\mathrm{ch}}}) = (1.3, \ 1.1, \ 1.9, \ 0.94, \ 1.7).$$

For these problems we therefore conclude that our algorithm performs somewhat inferiorly to MOSEK, using less iterations but cummulatively 30% more Choleschy factorization than MOSEK.

6.3.4 Entropy problems

Table 5 shows results from solving a set of real-world entropy problems supplied to us by MOSEK. Generally the problems have many variables compared to the number of constraints resulting in a very "fat" constraint matrix A. For these problems we compare our algorithms to the commercial solver from

Table 5 *Computational results for entropy problems.*

Problem			NPC				mskenopt	
name	N	M	it	ch	ce	st	ch	st
prob	17	15	9	15	0.7	opt	8	opt
prob2	18	14	9	18	1.0	opt	8	opt
ento46	130	21	25	50	1.0	opt	42	opt
ento47	255	21	23	49	1.1	opt	54	opt
ento28	740	16	38	78	1.1	opt	63	opt
ento30	740	21	38	84	1.2	opt	55	opt
ento31	740	21	38	84	1.2	opt	55	opt
ento22	794	28	28	60	1.1	ipr	14	ipr
ento21	931	28	55	112	1.0	ipr	18	ipr
a_tb	1127	25	38	87	1.3	opt	97	opt
ento23	1563	28	34	73	1.1	ipr	14	ipr
ento20	1886	28	41	94	1.3	opt	21	ipr
a_12	2183	37	46	104	1.3	opt	47	opt
ento12	2183	37	26	60	1.3	ipr	13	ipr
a_13	3120	37	44	99	1.2	ipr	20	ipr
a_23	3301	37	31	86	1.8	ipr	20	ipr
a_34	3905	37	36	83	1.3	ipr	17	ipr
a_14	3986	37	47	109	1.3	ipr	20	ukn
a_35	4333	37	43	90	1.1	ipr	18	ipr
a_bd	4695	26	44	102	1.3	opt	78	opt
ento2	4695	26	44	102	1.3	opt	78	opt
a_24	5162	37	36	90	1.5	ipr	23	ipr
ento3	5172	28	49	126	1.6	opt	146	opt
ento50	5172	28	49	126	1.6	opt	146	opt
a_15	5668	37	84	176	1.1	opt	34	ipr
a_25	6196	37	61	137	1.2	opt	122	opt
a_36	7497	37	40	99	1.5	ipr	18	ipr
a_45	7667	37	54	120	1.2	opt	23	ipr
ento26	7915	28	43	107	1.5	opt	131	opt
a_16	8528	37	89	204	1.3	opt	135	opt
a_26	9035	37	39	108	1.8	opt	113	opt
ento45	9108	37	51	128	1.5	opt	149	opt
a_46	9455	37	40	102	1.6	ipr	20	ipr
a_56	9702	37	65	158	1.4	opt	123	opt
ento25	10142	28	116	250	1.2	opt	149	opt
entodif	12691	40	50	130	1.6	opt	155	opt
ento48	15364	31	16	52	2.2	opt	47	opt

MOSEK, which solves the monotone complementarity problem [3] corresponding to the entropy problem. We see that, except for a few of the problems, our algorithm compares somewhat infavorably to MOSEK. With the notation defined in Section 6.3.3, we find

$$(S, \; \overline{\mathcal{R}^{\mathrm{it}}}, \; \overline{\mathcal{R}^{\mathrm{ch}}}, \; \widetilde{\mathcal{R}^{\mathrm{it}}}, \; \widetilde{\mathcal{R}^{\mathrm{ch}}}) = (1.6, \; 1.2, \; 2.8, \; 0.93, \; 2.1).$$

That is, although NPC uses fewer iterations, it uses cummulatively about 60% more Choleschy factorizations to solve the entire set of problems when compared to MOSEK.

We remark that the solvers from MOSEK for entropy problems and geometric programs are two *different* solvers, each purpose-built to solve those particular problems. Our algorithm, on the other hand, is a general purpose algorithm with no tuning of parameters to a particular problem. From simple experiments we know that tuning the parameters η and β for each type

of problem, we could improve the computational performance of our algorithm. However, since we believe in the importance of practical applicability we choose to fix the parameters and instead let our algorithm enjoy a very high degree of versatility. In that light, and considering the fact that MOSEK is an industry-grade implementation, we believe our algorithm compares very well.

7 Conclusions

In this paper, we have presented a homogeneous primal-dual interior-point algorithm for nonsymmetric convex conic optimization. Unlike previous work solving the homogenized convex conic problem, our algorithm makes use *only* of the primal barrier function thus making the algorithm widely applicable. We have proven the standard $\mathcal{O}(\sqrt{\nu}\log(1/\epsilon))$ worst-case complexity result and adapted to the nonsymmetric case techniques known to significantly improve efficiency of algorithms for self-scaled cones. These include quasi-Newton updating to reduce computational load and a Runge-Kutta type second order search direction. We demontrated how to efficiently implement these techniques without loosing the ability to exploit sparsity in the data matrix A. Finally we have presented extensive computational results that indicate the algorithm works well in practice.

By inspecting the tables in Section 6.3, we see that

- The performance of the algorithm depends a lot on the type of problem.
- For the p-cone problems, our algorithm superior in performance to SeDuMi called via CVX. These experiments clearly show the potential advantage of directly modelling nonsymmetric problems by using nonsymmetric cones.
- For the facility location problems, our algorithm compares favorably to an algorithm [7], which is a dual variant of the one presented in [16].
- For geometric programs, our algorithm compares somewhat infavorably to MOSEK.
- For entropy maximization problems, our algorithm again compares somewhat infavorably to MOSEK.

The computational results comparing our algorithm to MOSEK should, however, be seen in the light of the comments in Section 6.3.4 on page 23.

Comparing the kind of algorithm we have presented with a primal-dual IPM for self-scaled cones, we see that the major difference is the need for a seperate correction phase. Nesterov remarks in [16] that this process can be seen as the process of finding a scaling point, i.e. a point w such that $x = H_w s$. It seems reasonable that this is a more complex problem when the cone is not symmetric. We can not compute it analytically, so we need an iterative procedure.

This difference is interesting theoretically as well as practically. For the problems we have considered, the centering problem certainly is a relatively easy problem compared to the full problem, in the sense that we do not need

a very *accurately* centered point. We have seen in our experiments with our algorithm that rarely more a couple of correction steps are needed, some of which may be comparably inexpensive quasi-Newton steps.

A Properties of the barrier function

Here we list some properties of logarithmically homogeneous self-concordant barriers (LHSCB) that we use in this paper. Many more properties and proofs can be found in [18,19].

Let $\mathcal{K}^\circ$ denote the interior of $\mathcal{K}$. We assume that $F : \mathcal{K}^\circ \mapsto \mathbb{R}$ is a LHSCB for $\mathcal{K}$ with barrier parameter ν. This means that for all $x \in \mathcal{K}^\circ$ and $t > 0$,

$$F(tx) = F(x) - \nu \log t.$$

It follows that the conjugate of F, denoted F^* and defined for $s \in (\mathcal{K}^*)^\circ$ by

$$F^*(s) = \sup_{x \in \mathcal{K}} \{-s^T x - F(x)\}$$

is a LHSCB for the dual cone $\mathcal{K}^*$. Similarly to the notation used in [18,19], we write the local Hessian norms on $\mathcal{K}$ and $\mathcal{K}^*$ as:

$$\|g\|_x = \|H_x^{1/2} g\|, \quad \text{for } x \in \mathcal{K}^\circ$$
$$\|h\|_s^* = \|(H_s^*)^{1/2} g\|, \quad \text{for } s \in (\mathcal{K}^*)^\circ$$
$$\|h\|_x^* = \|H_x^{-1/2} h\|, \quad \text{for } x \in (\mathcal{K})^\circ,$$

where $H_s^* = \nabla^2 F^*(s)$. Notice the different definitions of $\|\cdot\|_y^*$ depending on whether y is in $\mathcal{K}$ or $\mathcal{K}^*$. Using this convention and that $-g_x \in (\mathcal{K}^*)^\circ$ and $H_{-g_x}^* = H_x^{-1}$, we see that

$$\|s\|_{-g_x}^* = \|(H_{-g_x}^*)^{-1/2} s\| = \|H_x^{1/2} s\| = \|s\|_x^* \tag{17}$$

For $x \in \mathcal{K}^\circ$, F satisfies

$$H_x x = -g_x \tag{18}$$
$$x^T g_x = -\nu \tag{19}$$
$$\|x\|_x^2 = \nu. \tag{20}$$

The *Dikin ellipsoids* are feasible [4]. That is:

$$x \in \mathcal{K}^\circ \quad \Rightarrow \quad W(x) = \{u, \|u - x\|_x \le 1\} \subseteq \mathcal{K} \tag{21}$$
$$s \in (\mathcal{K}^*)^\circ \quad \Rightarrow \quad W^*(s) = \{h, \|h - s\|_s^* \le 1\} \subseteq K^*. \tag{22}$$

B The homogeneous and self-dual model

B.1 Optimality and infeasibility certificate

Let G be defined by (5) and notice that G is skew-symmetric: $G = -G^T$.

1. Observe that we can write (HSD) as $G(y, x, \tau)^T - (0, s, \kappa)^T = 0$. Pre-multiplying this equation by $(y, x, \tau)^T$ gives $x^T s + \tau \kappa = 0$.
2. $\tau > 0$ implies $\kappa = 0$ and hence $b^T(y/\tau) - c^T(x/\tau) = 0$ and therefore $x^T s = 0$. Dividing the two first linear feasibility equations of (HSD) by τ, we obtain the linear feasibility equations of (1). Thus $(x, y, s)/\tau$ is optimal for (PD).
3. If $\kappa > 0$ then $\tau = 0$ so $Ax = 0$ and $A^T y + s = 0$. Further $c^T x - b^T y = -\kappa < 0$ so not both $c^T x$ and $-b^T y$ can be non-negative. Assume $-b^T x < 0$. If (PD) is primal-feasible then there exists $\bar{x} \in \mathcal{K}$ such that $A\bar{x} = b$. But then $0 > -b^T y = -\bar{x}^T A^T y = \bar{x}^T s \geq 0$, a contradiction. We can argue similarly if $c^T x < 0$,

and this completes the proof of Lemma 1.

B.2 Self-duality

The problem (HSD) can be written in the form

$$
\begin{aligned}
\min_{\bar{x}, \bar{x}_f} \quad & \bar{c}^T \bar{x} + \bar{c}_f^T \bar{x}_f \\
\text{s.t.} \quad & \bar{A}\bar{x} + \bar{A}_f \bar{x}_f = \bar{b} \\
& \bar{x} \in \bar{\mathcal{K}}, \bar{x}_f \text{ free.}
\end{aligned}
\tag{23}
$$

where

$$
\begin{aligned}
&\bar{c} = 0,\ \bar{c}_f = 0,\ \bar{b} = 0 \\
&\bar{x} = (x, \tau, s, \kappa),\ \bar{x}_f = y \\
&\bar{\mathcal{K}} = \mathcal{K} \times \mathbb{R}_+ \times \mathcal{K}^* \times \mathbb{R}_+
\end{aligned}
\quad,\quad
\bar{A} = \begin{pmatrix} A & -b & & \\ & c & -I & \\ -c^T & & & -1 \end{pmatrix},\quad
\bar{A}_f = \begin{pmatrix} 0 \\ -A^T \\ b^T \end{pmatrix}
$$

The dual of (23) problem is

$$
\max_{\bar{y}, \bar{s}} \quad \bar{b}^T \bar{y}
$$

$$
\text{s.t.} \quad \bar{A}^T \bar{y} + \bar{s} = \bar{c} \tag{24}
$$

$$
\bar{A}_f^T \bar{y} = \bar{c}_f \tag{25}
$$

$$
\bar{s} \in (\bar{\mathcal{K}})^*, \bar{y} \text{ free.} \tag{26}
$$

Let us split the variables in parts: $\bar{s} = (\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4)$ and $\bar{y} = (\bar{y}_1, \bar{y}_2, \bar{y}_3)$. We can then write (24) as:

$$
\begin{aligned}
\bar{s}_1 + A^T \bar{y}_1 - c\bar{y}_3 &= 0 \\
\bar{s}_2 - b^T \bar{y}_1 + c^T \bar{y}_2 &= 0 \\
\bar{s}_3 - \bar{y}_2 = 0, \quad \bar{s}_4 - \bar{y}_3 &= 0
\end{aligned}
\tag{27}
$$

From (27), we can immediately eliminate $\bar{y}_2$ and $\bar{y}_3$ since they are equal to $\bar{s}_3$ and $\bar{s}_4$ respectively. The constraint (25) is equivalent to $A\bar{s}_3 - b\bar{s}_4 = 0$ and (26) is the same as

$$
(\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4) \in \mathcal{K}^* \times \mathbb{R}_+ \times \mathcal{K} \times \mathbb{R}_+
$$

We now see that we are left with the dual problem

$$
\begin{aligned}
\text{maximize} \quad & 0 \\
\text{subject to} \quad & A\bar{s}_3 \quad -b\bar{s}_4 \qquad\quad = 0 \\
& -A^T \bar{y}_1 \qquad\quad +c\bar{s}_4 \ -\bar{s}_1 = 0 \\
& b^T \bar{y}_1 \ -c^T \bar{s}_3 \qquad\quad -\bar{s}_2 = 0 \\
& (\bar{s}_3, \bar{s}_4) \in \mathcal{K} \times \mathbb{R}_+, \ (\bar{s}_1, \bar{s}_2) \in \mathcal{K}^* \times \mathbb{R}_+, \ \bar{y}_1 \in \mathbb{R}^m
\end{aligned}
$$

This problem is clearly equivalent to the problem (HSD) through this identification of variables:

$$
\bar{s}_1 \sim s, \quad \bar{s}_2 \sim \kappa, \quad \bar{s}_3 \sim x, \quad \bar{s}_4 \sim \tau, \quad \bar{y}_1 \sim y
$$

and this proves Lemma 2.

C Prediction

The direction z' is defined by

$$G(y', x') - (0, s') = -(G(y, x) - (0, s)) \tag{28}$$

$$s' + \mu H_x x' = -s \tag{29}$$

C.1 Reduction of residuals

We first show:

$$1. \quad s^T x' + x^T s' + x^T s = \psi(z)^T x' \tag{30}$$

$$2. \quad (x + x')^T (s + s') = 0 \tag{31}$$

$$3. \quad (x')^T s' = -\psi(z)^T x'. \tag{32}$$

1. We get $s^T x' + x^T s' + x^T s \overset{(29)}{=} s^T x' + x^T(-s - \mu H_x x') + x^T s$, which, after reduction, gives $(x')^T(s - \mu H_x x) = \psi(z)^T x'$.
2. Equation (28) is equivalent to $G(y + y', x + x') - (0; s + s') = 0$. Pre-multiplying this equation by $(y + y', x + x')$ gives (31).
3. Follows from expanding (31) and using (30).

Now the lemma follows readily: We simply note that the first equation follows directly from elementary linear algebra. To show the second:

$$
\begin{aligned}
\nu\mu(z^+) \quad &= \quad (x + \alpha x')^T (s + \alpha s') \\
&= \quad x^T s + \alpha(s^T x' + x^T s') + \alpha^2 (x')^T s' \\
&\overset{(30)-(32)}{=} x^T s + \alpha(-x^T s + \psi(z)^T x') + \alpha^2(-\psi(z)^T x') \\
&= \quad (1 - \alpha)x^T s + \alpha(1 - \alpha)\psi(z)^T x'
\end{aligned}
$$

which after division by ν proves Lemma 3.

C.2 Bounds on s, s' and x'

Assume $\|\psi\|_x^* \leq \eta\mu$. By definition, $\psi = s - \mu H_x x$, which after left-multiplication by $H_x^{-1/2}$, taking norms and squaring both sides gives

$$
\begin{aligned}
(\|s\|_x^*)^2 &= (\|\psi\|_x^*)^2 + \mu^2 \|x\|_x^2 + 2\mu x^T \psi \\
&= (\|\psi\|_x^*)^2 + 2 + \mu^2 \nu \\
&\leq \mu^2 (\nu + \eta^2) \\
\|s\|_x^* &\leq \mu\sqrt{\eta^2 + \nu} \tag{33}
\end{aligned}
$$

where we used (20) and $x^T \psi = 0$.

This bound allows us to obtain bounds on x' and s': Left-multiplying (29) by $H_x^{-1/2}$, taking norms and squaring both sides gives

$$
\begin{aligned}
(\|s'\|_x^*)^2 + \mu^2 \|x'\|_x^2 = (\|s\|_x^*)^2 - 2\mu(x')^T s' &\overset{(32)}{=} (\|s\|_x^*)^2 + 2\mu(x')^T \psi \\
&\leq (\|s\|_x^*)^2 + 2\mu \|x'\|_x \|\psi\|_x^*
\end{aligned}
$$

by Cauchy-Schwarz inequality. Therefore

$$\mu^2\|x'\|_x^2 \le (\|s\|_x^*)^2 + 2\mu\|x'\|_x\|\psi\|_x^*$$

Now subtracting $2\mu\|x'\|_x\|\psi\|_x^*$ and adding $(\|\psi\|_x^*)^2$ to both sides, we get

$$\left(\mu\|x'\|_x - \|\psi\|_x^*\right)^2 \le (\|s\|_x^*)^2 + (\|\psi\|_x^*)^2$$

or

$$
\begin{aligned}
\|x'\|_x &\le \mu^{-1}\left(\|\psi\|_x^* + \sqrt{(\|s\|_x^*)^2 + (\|\psi\|_x^*)^2}\right) \\
&\le \mu^{-1}(\eta\mu + \sqrt{\mu^2(\eta^2+\nu) + \eta^2\mu^2}) \\
&= k_x := \eta + \sqrt{\eta^2+\nu}.
\end{aligned}
\tag{34}
$$

For s', we similarly have

$$
\begin{aligned}
(\|s'\|_x^*)^2 &\le (\|s\|_x^*)^2 + 2\mu\|x'\|_x\|s'\|_x^* \\
(\|s'\|_x^* - \mu\|x'\|_x)^2 &\le (\|s\|_x^*)^2 + \mu^2\|x'\|_x^2 \\
\|s'\|_x^* &\le k_x\mu + \sqrt{\mu^2(\eta^2+\nu) + k_x^2\mu^2} \\
&= k_s\mu
\end{aligned}
\tag{35}
$$

where $k_s := k_x + \sqrt{(\eta^2+\nu) + k_x^2}$.

C.3 Feasibility of z^+.

Define $\alpha_1 := k_x^{-1} = \Omega(1/\sqrt{\nu})$. Then for any $\alpha \le \alpha_1$, we have

$$\|x - (x + \alpha x')\|_x = \alpha\|x'\|_x \overset{(34)}{\le} \alpha k_x \le 1$$

and so from (21), we conclude $x + \alpha x' = x^+ \in \mathcal{K}$.

Now, define $\alpha_2 := (1-\eta)k_s^{-1} = \Omega(1/\sqrt{\nu})$. Then for $\alpha \le \alpha_2$, we have

$$
\begin{aligned}
\mu^{-1}\|s^+ + \mu g_x\|_{-g_x}^* &= \mu^{-1}\|s + \alpha s' + \mu g_x\|_{-g_x}^* \\
&= \mu^{-1}\|\psi + \alpha s'\|_{-g_x}^* \\
&\overset{(17)}{\le} \mu^{-1}\|\psi\|_x^* + \mu^{-1}\alpha\|s'\|_x^* \\
&\overset{(35)}{\le} \eta + \alpha k_s \le 1.
\end{aligned}
$$

Since $-g_x \in \mathcal{K}^*$, we have by (22) that $\mu^{-1}s^+ \in \mathcal{K}^*$ and therefore $s^+ \in \mathcal{K}^*$. Therefore, Lemma 4 holds with $\alpha = \min\{\alpha_1, \alpha_2\} = \Omega(1/\sqrt{\nu})$.

C.4 Bound on ψ^+.

First, let us repeat the general definition (6) of the function ψ:

$$\psi(x, s, t) = s + tg_x.$$

Consider for a fixed v_0 the function

$$\Phi_t(x) = x^T v_0 + t g_x$$

which is self-concordant with respect to x. Define its Newton step by $n_t(x) := -\nabla^2 \Phi_t(x) \nabla \Phi_t(x)$. Define also $m = \|n_{t_2}(x)\|_x$. From the general theory of self-concordant functions, the following inequality holds. If $m \leq 1$, then

$$\|n_{t_2}(x_2)\|_{x_2} \leq \left(\frac{m}{1-m}\right)^2. \tag{36}$$

For a proof of this relation, see e.g. Theorem 2.2.4 in [23].

With $v_0 = s^+$, $t_2 = \mu^+$ and $x_2 = x^+$, the inequality (36) is

$$\|\psi^+\|_{x^+}^* \leq \mu^+ \left(\frac{m}{1-m}\right)^2. \tag{37}$$

where $\mu^+ m = \|H_x^{-1}(s^+ + \mu^+ g_x)\|_x = \|s^+ + \mu^+ g_x\|_x^*$. From Lemma 3 and (34), we have

$$|\mu - \mu^+| = |-\alpha\mu + \alpha(1-\alpha)\nu^{-1}\psi^T x'|$$
$$\leq \mu\alpha\left(1 + (1-\alpha)\eta k_x \nu^{-1}\right). \tag{38}$$

By the assumption $\|\psi\|_x^* \leq \eta\mu$ combined with (34), we have $\psi^T x' \geq -\eta k_x \mu$. Therefore

$$\mu^+ = (1-\alpha)\mu + \alpha(1-\alpha)\nu^{-1}\psi^T x'$$
$$\geq \mu(1-\alpha)(1-\alpha\eta k_x \nu^{-1})$$
$$\mu/\mu^+ \leq (1-\alpha)^{-1}(1-\alpha\eta k_x \nu^{-1})^{-1} \tag{39}$$

Let us now obtain a bound on m.

$$\mu^+ m = \|s^+ + \mu^+ g_x\|_x^* \tag{40}$$
$$= \|\psi - (\mu - \mu^+)g_x + \alpha s'\|_x^*$$
$$\leq \|\psi\|_x^* + |\mu - \mu^+| \|g_x\|_x^* + \alpha\|s'\|_x^*$$
$$\leq \eta\mu + \mu\alpha\left(1 + (1-\alpha)\eta k_x \nu^{-1}\right)\sqrt{\nu} + \alpha k_s \mu$$
$$\leq \mu\left(\eta + \alpha k_s + \alpha(1 + (1-\alpha)\nu^{-1}\eta k_x)\sqrt{\nu}\right)$$
$$m \leq (\mu/\mu^+)(\eta + \alpha(\sqrt{\nu} + k_s + \eta k_x))$$
$$\leq (1-\alpha)^{-1}(1-\alpha\eta k_x \nu^{-1})^{-1}(\eta + \alpha(\sqrt{\nu} + k_s + \eta k_x))$$

where we used (35), (38), (39) and the assumption $\|\psi\|_x^* \leq \eta\mu$. Now the reader can verify that for $\eta \leq 1/6$ and $\nu \geq 2$, we have the implication

$$\alpha \leq \alpha_3 := \frac{1}{11\sqrt{\nu}} = \Omega(1/\sqrt{\nu}) \quad \Rightarrow \quad m^2/(1-m)^2 \leq 2\eta \leq 1/3 \tag{41}$$

Then by (37), we see that (41) implies $\|\psi^+\|_{x^+}^* \leq 2\eta\mu^+$ and hence $z^+ \in \mathcal{N}(2\eta)$ which finishes the proof of Lemma 5.

D Correction phase

Assume $\|\psi(x, s, \mu)\|_x^* \leq \beta\mu$ where $\mu := \mu(z)$ with $z = (x, y, s)$. The equations defining the correction step $(\bar{x}, \bar{y}, \bar{s})$ are

$$G(\bar{y}, \bar{x}) - (0, \bar{s}) = 0 \tag{42}$$
$$\bar{s} - \mu H_x \bar{x} = -\psi(x, s, \mu) \tag{43}$$

and the next point is then $(x^+, y^+, s^+) := (x, y, s) + \bar{\alpha}(\bar{x}, \bar{y}, \bar{s})$. Left-multiplying (42) by $(\bar{y}, \bar{x})^T$, we get $\bar{x}^T \bar{s} = 0$. From (43), we then have

$$(\|\bar{s}\|_x^*)^2, \ \mu^2 \|\bar{x}\|_x^2 \leq (\|\bar{s}\|_x^*)^2 + \mu^2 \|\bar{x}\|_x^2 = (\|\psi(x, s, \mu)\|_x^*)^2 \leq \beta^2 \mu^2$$

and therefore

$$\|\bar{x}\|_x \leq \beta, \ \ \|\bar{s}\|_x^* \leq \beta\mu \tag{44}$$

From (43), we also have

$$\begin{aligned}
\|\psi(x, s, \mu) + \bar{\alpha}\bar{s}\|_x^* &= \|(1 - \bar{\alpha})\psi(x, s, \mu) - \bar{\alpha}\mu H_x \bar{x}\|_x^* \\
&\leq (1 - \bar{\alpha})\|\psi(x, s, \mu)\|_x^* + \bar{\alpha}\mu\|\bar{x}\|_x \\
&\leq (1 - \bar{\alpha})\beta\mu + \bar{\alpha}\mu\beta = \beta\mu
\end{aligned} \tag{45}$$

Where we used (44). Now define $m = (\mu^+)^{-1}\|s^+ + \mu^+ g_x\|_x^*$. Then estimating similarly to (40), we get

$$\begin{aligned}
\mu^+ m &\leq \|\psi(x, s, \mu) + (\mu^+ - \mu)g_x + \bar{\alpha}\bar{s}\|_x^* \\
&\leq \beta\mu(1 + \bar{\alpha}(\beta\nu^{-1/2} + 1))
\end{aligned}$$

and similarly to the computation in (39), we therefore find

$$\mu/\mu^+ \leq (1 - \bar{\alpha}\nu^{-1}\beta^2)^{-1}$$

so that alltogether

$$m \leq \beta(1 - \bar{\alpha}\nu^{-1}\beta^2)^{-1}(1 + \bar{\alpha}(\beta\nu^{-1/2} + 1)). \tag{46}$$

Now we can apply the theorem (36) with $v_0 = s^+$, $t = \mu$ and $x_2 = x^+$:

$$\|\psi(x^+, s^+, \mu^+)\|_{x^+}^* \leq \mu^+ \left(\frac{m}{1 - m}\right)^2 \tag{47}$$

The reader can verify that for $\bar{\alpha} \leq 1/84$, $\nu \geq 2$, $\beta \leq 2\eta \leq 1/3$, the bound (46) implies that when recursively using (47) *twice*, we obtain

$$\|\psi(x^+, s^+, \mu^+)\|_{x^+}^* \leq \frac{1}{2}\beta \leq \eta$$

and therefore $z^+ \in \mathcal{N}(\eta)$ which proves Lemma 6.

E Algorithm complexity

From Lemma 3, we have that the linear residuals $G(y, x) - (0, s)$ are reduced by a factor $(1 - \alpha)$ in each iteration. Since we can always take $\alpha = \Omega(1/\sqrt{\nu})$, we see that $G(y, x) - (0, s)$ decreases geometrically with a rate of $(1 - \Omega(1/\sqrt{\nu}))$ which implies that

$$\|G(y, x) - (0, s)\| \leq \epsilon\|G(y^0, x^0) - (0, s^0)\|$$

in $\mathcal{O}(\sqrt{\nu}\log(1/\epsilon))$ iterations.

To see that the same holds for $\mu(z)$, let us briefly use the following notation: z is the starting point, z^+ is the point after prediction and $z^{(j)}$ is the point after applying j correction steps starting in z^+. Then from Lemma 3 and (34), we have

$$\begin{aligned}
\mu(z^+) &\leq (1 - \alpha)\mu(z) + \alpha(1 - \alpha)\nu^{-1}\mu\eta k_x \\
&\leq \mu(z)(1 - \alpha)(1 + \alpha\eta k_x \nu^{-1}) \\
&= \mu(z)(1 - \Omega(1/\sqrt{\nu}))
\end{aligned} \tag{48}$$

Since $\bar{x}^T\bar{s} = 0$, we see from (43) that

$$(x^+)^T\bar{s} = \mu(z^+)\bar{x}^T g_{x^+} = \bar{x}^T\psi(x^+, s^+, \mu(z^+)) - \bar{x}^T s^+ \tag{49}$$

Therefore

$$
\begin{aligned}
\nu\mu(z^{(1)}) &= (x^+ + \bar{\alpha}\bar{x})^T(s^+ + \bar{\alpha}\bar{s}) \overset{(49)}{=} (x^+)^T(s^+) + \bar{\alpha}\bar{x}^T\psi(x^+, s^+, \mu(z^+)) \\
&\leq \nu\mu(z^+) + \bar{\alpha}\beta^2\mu(z^+) \\
&= \nu\mu(z^+)(1 + \bar{\alpha}\beta^2\nu^{-1})
\end{aligned}
$$

and hence

$$
\begin{aligned}
\mu(z^{(2)}) &\leq \mu(z^+)(1 + \bar{\alpha}\beta^2\nu^{-1})^2 \\
&\overset{(48)}{\leq} \mu(z)(1 - \Omega(1/\sqrt{\nu}))(1 + \bar{\alpha}\beta^2\nu^{-1})^2 \\
&= \mu(z)(1 - \Omega(1/\sqrt{\nu}))
\end{aligned}
$$

which shows that also $\mu(z)$ is decreased geometrically with a rate of $(1 - \Omega(1/\sqrt{\nu}))$. Therefore

$$\mu(z) \leq \epsilon\mu(z^0)$$

in $\mathcal{O}(\sqrt{\nu}\log(1/\epsilon))$ iterations, finishing the proof of Theorem 1.

References

1. Andersen, E. D., Andersen, K. D.: The MOSEK interior point optimization for linear programming: an implementation of the homogeneous algorithm. In: Frenk, H., Roos, K., Terlaky, T., Zhang, S.: High Performance Optimization, 197–232. Kluwer, Boston (1999).
2. Andersen, E. D., Roos, C., Terlaky, T.: On implementing a primal-dual interior-point method for conic quadratic optimization. Math. Program. 95(2), 249–277 (2003).
3. Andersen, E. D., Ye, Y.: On a homogeneous algorithm for the monotone complementarity problem. Math. Program. 84(2), 375–399 (1999).
4. Ben-Tal, A., Nemirovski, A. S.: Lectures on Modern Convex Optimization: Analysis, Algorithms and Engineering Applications. SIAM, Philadelphia (2001).
5. Boyd, S., Kim, S. J., Vandenberghe, L., Hassibi, A.: A Tutorial on Geometric Programming. Optim. Eng. 8, 67–127 (2007).
6. Butcher, J. C.: Numerical Methods for Ordinary Differential Equations. Wiley, 2nd edition (2008).
7. Chares, P. R.: Cones and Interior-Point Algorithms for Structured Convex Optimization involving Powers and Exponentials. PhD thesis, Uni. Catholique de Louvain, 2009.
8. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, version 1.21. http://cvxr.com/cvx, October 2010.
9. Güler, O.: Barrier Functions in Interior Point Methods. Math. Oper. Res. 21, 860–885 (1996).
10. Karmarkar, N.: A new polynomial-time algorithm for linear programming. Combinatorica 4, 373–395 (1984).
11. Luo, Z. Q., Sturm, J. F., Zhang, S.: Conic convex programming and self-dual embedding. Optim. Method. Softw. 14, 169–218 (2000).
12. Mehrotra, S.: On the Implementation of a Primal-Dual Interior Point Method. SIAM J. Optim. 2, 575–601 (1992).
13. MOSEK Optimization Software: Developed by MOSEK ApS. See www.mosek.com.
14. Nemirovski, A. S., Todd, M. J.: Interior-point methods for optimization. Acta Numerica 17, 191–234 (2008).
15. Nesterov, Y. E.: Constructing self-concordant barriers for convex cones. CORE Discussion Paper (2006/30).

16. Nesterov, Y. E.: Towards Nonsymmetric Conic Optimization. Optim. Method. Softw. 27, 893–917 (2012).
17. Nesterov, Y. E., Nemirovski, A. S.: Interior-Point Polynomial Algorithms in Convex Programming. SIAM (1994).
18. Nesterov, Y. E., Todd, M. J.: Self-Scaled Barriers and Interior-Point Methods for Convex Programming. Math. Oper. Res. 22, 1–42 (1997).
19. Nesterov, Y. E., Todd, M. J.: Primal-Dual Interior-Point Methods for Self-Scaled Cones. SIAM J. Optim. 8, 324–364 (1998).
20. Nesterov, Y. E., Todd, M. J., Ye, Y.: Infeasible-Start Primal-Dual Methods and Infeasibility Detectors for Nonlinear Programming Problems. Math. Program. 84, 227–267 (1999).
21. Netlib repository: Collection of linear programs. See `www.netlib.org/lp/`.
22. Nocedal, J., Wright, S. J.: Numerical Optimization. Springer, 2nd edition (2006).
23. Renegar, J.: A Mathematical View of Interior-Point Methods in Convex Optimization. SIAM, Philadelphia (1987).
24. Sturm, J. F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optim. Method. Softw. 12, 625–653 (1999).
25. Sturm, J. F.: Implementation of Interior Point Methods for Mixed Semidefinite and Second Order Cone Optimization Problems. Optim. Method. Softw. 17, 1105–1154 (2002).
26. Tuncel, L.: Primal-Dual Symmetry and Scale Invariance of Interior-Point Algorithms for Convex Optimization. Math. Oper. Res. 23, 708–718 (1998).
27. Tuncel, L.: Generalization Of Primal-Dual Interior-Point Methods To Convex Optimization Problems In Conic Form. Found. Comput. Math. 1, 229–254 (2001).
28. Wright, S. J.: Primal-Dual Interior-Point Methods. SIAM (1987).
29. Xu, X., Hung, P. F., Ye, Y.: A simplified homogeneous and self-dual linear programming algorithm and its implementation. Ann. Oper. Res. 62, 151–171 (1996).
30. Xue, G. Ye, Y.: An Efficient Algorithm for Minimizing a Sum of p-Norms. SIAM J. Optimiz. 10, 551–579 (1999).
31. Ye, Y.: Interior Point Algorithms: Theory and Analysis. Wiley (1997).