A homogeneous interior-point algorithm for nonsymmetric convex conic optimization

Anders Skajaa · Yinyu Ye

Abstract A homogeneous 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 maximization problems and geometric programs; all formulated as nonsymmetric convex conic optimization problems.

A. Skajaa (✉)
Department of Applied Mathematics and Computer Science,
Technical University of Denmark, 2800 Kgs. Lyngby, Denmark
e-mail: andsk@imm.dtu.dk

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

Y. Ye
The International Center of Management Science and Engineering,
Nanjing University, Nanjing 210093, China

1 Introduction

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

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

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 non-empty 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 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 [16], 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 [17,18]. The practical efficiency of these algorithms has been widely verified, see e.g. [1,2,23].

In [9], Güler demonstrated that self-scaled cones are identical to those that are *symmetric*; a class that comprises just five cones of which only two are interesting for optimization. These cones are 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). Notice that linear programming is a subset of semidefinite programming.

Although these two 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 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 [25,27]. Another 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 [15]. 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 [19] to solve a homogeneous model of the general convex conic problem (PD). This leads to a method with some desirable properties but at the same time it has 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. The so-called doubly non-negative cone[1] is an example of the latter situation. A simple barrier for the primal cone is known, but no explicit barrier function for the dual cone is known.

Building on the algorithms of [15] and [19], we present in this paper a primal–dual interior-point algorithm for a homogeneous model of (PD). This approach has proven successful for self-scaled cones [2,22,26] 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 [19], 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 [2,12,23]. 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.

We will assume that $\mathcal{K}$ is a proper cone and 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 $\mathcal{K}$. The definition of the barrier parameter ν of F and many of the useful properties of these functions are listed in Appendix 1.

---

[1] A positive semidefinite matrix with all non-negative entries is called *doubly non-negative*.

This paper is organized in two main parts. In the first, which consists of Sects. 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 Sect. 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 that solve the following problem:

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

The motivation for doing so is summarized in the following.

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 section "Optimality and infeasibility certificate" of Appendix 2.

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.

Any useful algorithm aimed at solving (HSD) must therefore at least have the following two properties: If (PD) is both primal and dual feasible and has zero duality

gap, the algorithm must produce a solution (or approximate solution) to (HSD) with $\tau > 0$. Conversely, if (PD) is either primal or dual infeasible, the algorithm must produce a solution (or approximate solution) to (HSD) with $\kappa > 0$. In [11], it is thoroughly demonstrated which properties are sufficient for an algorithm to meet these goals. The algorithm that we later present in Sect. 4 indeed has these properties. However, since this paper concentrates on algorithmic aspects we omit the details regarding the various primal and dual infeasibility cases and what they mean for the homogeneous model and algorithm. Instead, we refer the reader to [19] and particularly [11] for a much more detailed discussion in this direction.

There is another desirable feature of the homogeneous model (HSD):

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

Proof See section "Self-duality" of Appendix 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. Specifically, there is no need to handle and store variables from the dual of (HSD) since they are identical to those of the primal.

Given the two lemmas above and the comments following Lemma 1, we can state the following desirable consequences of solving the homogeneous model (HSD) with our algorithm to be presented later:

- If the original problem (PD) is primal and dual feasible and has zero duality gap, an optimal primal–dual solution is found and a certificate of optimality is produced.
- If the original primal–dual pair (PD) is primal or dual infeasible, a certificate of this infeasibility is produced.
- The dimension of the problem is not essentially larger than that of the original primal–dual pair (PD) and does not require more computational effort to solve.
- The algorithm can be initialized in a point not necessarily feasible w.r.t. the linear constraints of (HSD).

When $\mathcal{K}$ is not equal to $\mathbb{R}_n^+$, it is possible that the pair (PD) is primal and dual feasible but has strictly positive duality gap. In this situation, a tiny perturbation to the problem data exists such that the perturbed problem has a solution with $\tau + \kappa > 0$. Thus, the problem is ill-posed. See [4] for further discussion and examples of this exceptional case.

3 Nonsymmetric path following

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

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

where F again is a logarithmically homogeneous self-concordant barrier function[2] (LHSCB) with barrier parameter ν. The Karush–Kuhn–Tucker (KKT) conditions of

---

[2] See Appendix 1 for a list of properties of this class of functions.

problem (2) 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 $u(\mu) = (x(\mu), y(\mu), s(\mu))$. Using relation (20) from Appendix 1, it is easy to see that central path points satisfy $c^T x(\mu) - b^T y(\mu) = x(\mu)^T s(\mu) = \nu\mu$. 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 system (3), we will denote it $p(\mu)$.

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 [24]. Thus no particular emphasis is put on either the primal or the dual problem, which is a desirable feature of an algorithm. If $\mathcal{K}$ is self-scaled, a symmetric $u'(\mu)$ can be computed using a *scaling point* [17,18]. 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 occurring in a symmetric IPM. A further disadvantage is that one must be able to compute both the primal and the dual barriers, their gradients and Hessians, which may prove difficult.

Nesterov showed in [15] that a scaling point determined during an iterative centering (correction) procedure can be used to compute a symmetric search direction $u'(\mu)$. Let us briefly describe the concept underlying the algorithm from [15]. The following *proximity* measure is needed:

For (HSD), need to replace x by (x,τ) and y by (y,κ). Then F, F* and ν changes accordingly

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

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 1 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** $\Psi(x, y, s) > \eta$
   $\qquad (x, y, s) := (x, y, s) + \hat{\alpha} p(\mu)$
   end while

where α in step 1 is chosen so that $\Psi(x, y, s) < \beta$ after step 1 and $\hat{\alpha}$ is chosen to be $\lambda/(1+\lambda)$, where λ is the Newton decrement. In [15], it is proved that with appropriate choices of η, β and α, the above algorithm converges in $\mathcal{O}(\sqrt{\nu} \log{(1/\epsilon)})$ iterations. This method uses only the Hessian $\nabla^2 F(\cdot)$ of the primal barrier but still the *value* of the dual barrier $F^*(\cdot)$. Two serious practical drawbacks of the 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.

The approach of [19] is instead to compute a symmetric $u'(\mu)$ by using both the Hessian of the primal and the dual barriers. Again, the iteration complexity result $\mathcal{O}(\sqrt{\nu} \log{(1/\epsilon)})$ is obtained and the two practical drawbacks of [15] are alleviated by the use of a homogeneous model. Two major disadvantages of the method of [19] are that one must know (or be able to compute) Hessians of both barriers and that the linear systems that must be solved are double in size.

Our goal in this paper is to construct an efficient algorithm utilizing the main ideas of [15] and [19], but adapted to be efficient for the homogeneous model (HSD) *without* using the Hessians of the primal *and* the dual barrier. In fact, our method does not make *any* use of the dual barrier — not even the function value. Unlike [15] and [19], our prediction direction $u'(\mu)$ will *not* be exactly symmetric unless the iterate is exactly on the central path, which is rarely the case. However, it turns out to be sufficient that we ensure that the iterate is "close to" the central path. This will guarantee a high enough quality of the prediction direction. In exchange for dropping the "exact symmetric" tangent direction we obtain a method that does not suffer from any of the above mentioned drawbacks of the methods in either of [15] and [19] while still maintaining the $\mathcal{O}(\sqrt{\nu} \log{(1/\epsilon)})$ iteration complexity result.

Thus, compared to [19], this work represents the following improvements:

1. We need only to know the primal barrier function, its gradient and Hessian (no need for the dual barrier and its derivatives).
2. The linear systems that need to be solved in each iteration are half the dimension (i.e. a factor 8 faster in terms of computation time).

Likewise, in relation to [15], this work represent the following improvements:

1. We need only to know the primal barrier function, its gradient and Hessian (no need for the dual barrier function value).
2. We do not require a feasible starting point (no phase-I method needed).
3. Our method detects infeasibilities in the problem.

We are also aware of the apparent gap between IPM complexity theory and state-of-the-art implementations, see e.g. the introduction of [15] 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. [1,23] 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 use of the following notation. For the concatenation of two vectors v and w, we will sometimes use the MATLAB-inspired notation $(v; w)$ and otherwise the usual $\begin{pmatrix} v \\ w \end{pmatrix}$. We will further simplify notation by writing

$$\bar{x} = \begin{pmatrix} x \\ \tau \end{pmatrix} = (x; \tau), \quad \bar{s} = \begin{pmatrix} s \\ \kappa \end{pmatrix} = (s; \kappa)$$

$$\bar{F}(\bar{x}) = F(x) - \log \tau, \quad \bar{F}^*(\bar{s}) = F^*(s) - \log \kappa$$

and

$$\bar{\mathcal{K}} = \mathcal{K} \times \mathbb{R}_+, \quad \bar{\mathcal{K}}^* = \mathcal{K}^* \times \mathbb{R}_+, \quad \bar{\nu} = \nu + 1$$

This notation is consistent with that of [19]. Notice that $\bar{F}$ and $\bar{F}^*$ are logarithmically homogeneous self-concordant barrier functions for the cones $\bar{\mathcal{K}}$ and $\bar{\mathcal{K}}^*$ respectively.

We will also use a higher level of aggregation: $z = (\bar{x}; y; \bar{s}) = (x; \tau; y; s; \kappa) \in \mathcal{F} := \bar{\mathcal{K}} \times \mathbb{R}^m \times \bar{\mathcal{K}}^*$ and define the *complementarity gap* of z by $\mu(z) := (\bar{x}^T \bar{s})/\bar{\nu}$. We will write $g_{\bar{x}} = \nabla \bar{F}(\bar{x})$ and $H_{\bar{x}} = \nabla^2 \bar{F}(\bar{x})$ and make use of the following local norms:

$$\|u\|_{\bar{x}} = \|H_{\bar{x}}^{1/2} u\|, \quad \|v\|_{\bar{x}}^* = \|H_{\bar{x}}^{-1/2} v\|, \quad \text{for } u \in \bar{\mathcal{K}} \quad \text{and} \quad v \in \bar{\mathcal{K}}^*,$$

where $\| \cdot \|$ denotes the standard Euclidean norm. See also Appendix 1 for more properties of these local norms. In our new notation, we can write the homogeneous model simply as

$$G \begin{pmatrix} y \\ \bar{x} \end{pmatrix} - \begin{pmatrix} 0 \\ \bar{s} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad z = \begin{pmatrix} \bar{x} \\ y \\ \bar{s} \end{pmatrix} \in \mathcal{F} \tag{4}$$

where G is the skew-symmetric matrix

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

Equations such as (4) will usually be written as $G(y; \bar{x}) - (0; \bar{s}) = (0; 0)$ to save vertical space. Notice that the expression $G(y; \bar{x})$ involves a *multiplication* between G and $(y; \bar{x})$ and the parenthesis thus do not denote arguments to a function. This latter situation will be clear from the context.

4.2 The central path in the homogeneous model

For $\bar{x} \in \bar{\mathcal{K}}, \bar{s} \in \bar{\mathcal{K}}^*$ and a scalar t, let us define the function

$$\psi(\bar{x}, \bar{s}, t) := \bar{s} + t g_{\bar{x}}. \tag{6}$$

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

$$G(y_\gamma; \bar{x}_\gamma) - (0, \bar{s}_\gamma) = \gamma \left(G(y^0; \bar{x}^0) - (0; \bar{s}^0) \right) \tag{7}$$

$$\psi(\bar{x}_\gamma, \bar{s}_\gamma, \gamma \mu^0) = 0 \tag{8}$$

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 $\eta \in [0, 1]$, we define the set

$$\mathcal{N}(\eta) = \left\{ z = (\bar{x}; y; \bar{s}) \in \mathcal{F} : \quad \|\psi(\bar{x}, \bar{s}, \mu(z))\|_{\bar{x}}^* \leq \eta \mu(z) \right\} \tag{9}$$

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

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 $\bar{X}\bar{s} = \gamma \mu^0 e$ where $\bar{X} = \text{diag}(\bar{x})$ and $e = (1, \ldots, 1)$. Similarly, the inequality in (9) reduces to $\|\bar{X}\bar{s} - \mu e\| \leq \eta \mu(z)$.

4.3 Prediction

The direction d_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

$$d_{\bar{s}_\gamma} = -\mu^0 g_{\bar{x}_\gamma} - \gamma \mu^0 H_{\bar{x}_\gamma} d_{\bar{x}_\gamma}$$

where $d_{\bar{x}_\gamma}$ denotes $\bar{x}_\gamma$ differentiated w.r.t. γ and similarly for other variables. By (8), we have $\gamma^{-1}\bar{s}_\gamma = -\mu^0 g_{\bar{x}_\gamma}$, which we insert and get

$$d_{\bar{s}_\gamma} + \gamma\mu^0 H_{\bar{x}_\gamma} d_{\bar{x}_\gamma} = \gamma^{-1}\bar{s}_\gamma.$$

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

$$G(d_y; d_{\bar{x}}) - (0; d_{\bar{s}}) = -(G(y; \bar{x}) - (0; \bar{s})) \tag{10}$$

$$d_{\bar{s}} + \mu(z)H_{\bar{x}}d_{\bar{x}} = -\bar{s} \tag{11}$$

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 literature. Determining the direction d_z thus amounts to solving the system of linear equations (10)–(11).

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

$$\begin{aligned}
z^+ &= (\bar{x}^+, y^+, \bar{s}^+) = (\bar{x} + \alpha d_{\bar{x}}, y + \alpha d_y, \bar{s} + \alpha d_{\bar{s}}) = z + \alpha d_z \\
\psi &= \psi(\bar{x}, \bar{s}, \mu(z)) \\
\psi^+ &= \psi(\bar{x}^+, \bar{s}^+, \mu(z^+)) \\
d_z &= \text{solution of } (10)-(11).
\end{aligned}$$

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

Lemma 3 *The direction d_z satisfies*

$$G(y^+; \bar{x}^+) - (0; \bar{s}^+) = (1-\alpha)(G(y; \bar{x}) - (0; \bar{s}))$$
$$\mu(z^+) = (1-\alpha)\mu(z) + (1-\alpha)\alpha\nu^{-1}\psi^T d_{\bar{x}}.$$

Proof See section "Reduction of residuals" of Appendix 3.

The first relation shows that the linear residuals are reduced by the factor $1 - \alpha$ along the direction d_z. The complementarity 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 $\alpha = \Omega(1/\sqrt{\bar{\nu}})$ so that $\mu(z^+) \le (1 - \Omega(1/\sqrt{\bar{\nu}}))\mu(z)$. Here, we use the "big-$\Omega$"-notation meaning that α is asymptotically bounded below by $1/\sqrt{\bar{\nu}}$ times a positive (possibly small) constant as $\nu \to \infty$.

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

Proof See section "Feasibility of z^+" of Appendix 3.

Lemma 5 *Assume $z \in \mathcal{N}(\eta)$. If $\eta \le 1/6$, then we can choose $\alpha = \Omega(1/\sqrt{\bar{\nu}})$ so that $z^+ \in \mathcal{N}(2\eta)$.*

Proof See section "Bound on ψ^+" of Appendix 3.

4.4 Correction phase

Given some point $z^+ = (\bar{x}^+, y^+, \bar{s}^+) \in \mathcal{N}(2\eta)$, the goal of the correction phase is to find a new point $z = (\bar{x}, y, \bar{s})$ which is close to the central path and satisfy the same linear constraints as z^+. That is, we want to find z so that $\|\psi(\bar{x}, \bar{s}, \mu(z))\|_{\bar{x}}^* \leq \eta\mu(z)$ and $G(y; \bar{x}) - (0; \bar{s}) = G(y^+; \bar{x}^+) - (0; \bar{s}^+)$. We therefore apply Newton's method to the equations

$$
\begin{aligned}
G(y; \bar{x}) - (0; \bar{s}) &= G(y^+; \bar{x}^+) - (0; \bar{s}^+) \\
\psi(\bar{x}, \bar{s}, \mu(z)) &= 0
\end{aligned}
\tag{12}
$$

The Newton step for these of equations is the solution $\delta_z := (\delta_{\bar{x}}, \delta_y, \delta_{\bar{s}})$ to the following linear system of equations: s

$$
G(\delta_y; \delta_{\bar{x}}) - (0; \delta_{\bar{s}}) = 0 \tag{13}
$$
$$
\delta_{\bar{s}} + \mu(z) H_{\bar{x}} \delta_{\bar{x}} = -\psi(\bar{x}, \bar{s}, \mu(z)) \tag{14}
$$

We then solve (13)–(14) and starting from $z = z^+$, we apply

$$
z := z + \hat{\alpha}\delta_z \tag{15}
$$

repeatedly until $\|\psi(\bar{x}, \bar{s}, \mu(z))\| \leq \eta\mu(z)$.

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 4.

4.5 Convergence and complexity of algorithm

It is evident that our algorithm is a nonsymmetric conic generalization of the simplified LP homogeneous and self-dual model [26]. Similarly to [26], let us write $\theta^{k+1} = (1 - \alpha^k)\theta^k$ and $\theta^0 = 1$, where α^k is the step length taken in the prediction step in the k'th iteration. From (13), we see that the linear residuals do not change during the correction phase. Thus, a useful result from [26] applies also to our algorithm:

Lemma 7 *Algorithm* 1 *generates iterates $z^k = (x^k, \tau^k, y^k, s^k, \kappa^k), k = 0, 1, \ldots$ that satisfy*

$$
\bar{\nu}\left(\mu^k/\theta^k + \theta^k\mu^0\right) = (s^k)^T x^0 + (x^k)^T s^0 + \kappa^k\tau^0 + \tau^k\kappa^0 \tag{16}
$$

Proof See Lemma 4, page 57 in [26].

This lemma implies that if μ^k and θ^k decrease at the same rate, then (16) functions as a normalizing constraint—i.e. all the iterates remain bounded. This is readily seen: The left-hand side of (16) remains bounded and since each term on the right-hand

---

Algorithm 1 Nonsymmetric Predictor–Corrector Algorithm

Require: Barrier function F, $\eta \leq 1/6$, and initial point $z \in \mathcal{F} \cap \mathcal{N}(\eta)$.
  $\hat{\alpha} := 1/84$
  Repeat
    Set $\mu := \mu(z)$
    Stopping
      If stopping criteria satisfied: **terminate**.
    Prediction
      Solve (10)–(11) for d_z
      Choose largest α so that $z + \alpha d_z \in \mathcal{F} \cap \mathcal{N}(2\eta)$
      Set $z := z + \alpha d_z$ and $\mu = \mu(z)$.
    Correction
      Solve (13)–(14) for δ_z
      Set $z := z + \hat{\alpha}\delta_z$
      Solve (13)–(14) for δ_z
      Set $z := z + \hat{\alpha}\delta_z$
  End

---

side is non-negative, each term must remain individually bounded. In particular as μ^k decreases to zero, at least one of τ^k and κ^k will go to zero while the other will remain non-negative and bounded above.

The following theorem will now finish our analysis.

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

$$\mu(z) \leq \epsilon\mu(z^0) \quad and \quad \|G(y; \bar{x}) - (0; \bar{s})\| \leq \epsilon\|G(y^0; \bar{x}^0) - (0; \bar{s}^0)\|$$

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

Proof See Appendix 5.

It is important to note that this theorem alone does not guarantee that we have recovered a sufficiently accurate solution (or infeasibility certificate) to the original problem (PD), only to (HSD). From the proofs of Theorem 1 and Lemma 3 it follows, however, that μ^k and θ^k decrease at the same rate. Therefore, Lemma 3 guarantees that τ^k and κ^k both remain bounded and if the final point z has one of τ or κ large enough a sufficiently accurate solution or infeasibility certificate for (PD) has been determined. For further details regarding the exact accuracy obtained and an explanation of all different types of feasibility cases, the reader is again referred to [11,19]. For us, importance is placed on the ability to practically distinguish these cases and what we mean by "sufficiently close" is precisely stated in Sect. 5.4.

In this section, we have emphasized only the asymptotic behavior of our algorithm. 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.

5 Implementation

In order for an interior-point method to be practical and competitive, the implementation must deviate somewhat from the pure theoretical algorithm. In this section, we describe how such an efficient algorithm can be implemented.

Algorithm 2 Aggressive step implementation

Require: Barrier function F, $0 < \eta \le \beta < 1$, 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 d_z
      Choose largest α so that $z + \alpha d_z \in \mathcal{F} \cap \mathcal{N}(\beta)$
      Set $z := z + \alpha d_z$ and $\mu = \mu(z)$.
    Correction
      Repeat
        Solve (13)–(14) for δ_z
        Choose $\hat{\alpha}$ to approximately minimize $\|\psi\|_{\bar{x}}^*$ along δ_z
        Set $z := z + \hat{\alpha} \delta_z$
      Until $z \in \mathcal{F} \cap \mathcal{N}(\eta)$.
  End

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 $\beta \ge 0.80$. This leads to faster convergence once we get close to the optimal point. Indeed we do observe what appears to be super-linear 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 $\|\psi\|_{\bar{x}}^* \le \eta \mu$ 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 $H_{\bar{x}}$ and of the possibly sparse $m \times m$ matrix $Q = A H_{\bar{x}}^{-1} A^T$. 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. [20]), a rank-2 updating scheme: $B^+ = B + k^{(v)} v v^T + k^{(w)} w w^T$. 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 $q \le J$, let $B^{(q)}$ be the q'th update of H^{-1}, i.e.

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

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

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

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

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

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 back-substitutions 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 $\|\psi\|_{\bar{x}}^* \leq \eta\mu$. The resulting BFGS search direction is a descent direction for the function $\|\psi\|_{\bar{x}}^*$, so by using a backtracking line search along these directions, we can not make the objective worse by proceeding 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 $\|v\|_{\bar{x}}^*$ is computed as $(v^T H_{\bar{x}}^{-1} v)^{1/2}$. Computing this number requires the evaluation and factorization of $H_{\bar{x}}$. But since $H_{\bar{x}}$ is block-diagonal, this operation is cheap.

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 $AH_{\bar{x}}^{-1}A^T$ relative to the cost of subsequent back-substitutions, 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 $AH_{\bar{x}}^{-1}A^T$ do not vary with $\bar{x}$, 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 solving self-scaled problems. Mehrotra's second order correction generalizes nicely to self-scaled conic problem by use of the Jordan product that can be defined on such cones, see e.g. [2]. For non-symmetric cones, this generalization seems to no longer be possible. Hoping to achieve a similar improvement in performance, we suggest instead to compute 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 $z = (\bar{x}, y, \bar{s})$, it is clear that the equations (10)–(11) defining d_z can be written

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

The central path is thus the solution of the ordinary differential equation defined by $d_z(\mu) = f(z)$. A step in the predictor direction, i.e. the direction d_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 $Kd_z = u$. Such a direction is defined by

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

$$\zeta = (\zeta_{\bar{x}}, \zeta_y, \zeta_{\bar{s}}) = z(\mu) + \theta h f(z)$$

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 $H_{\zeta_{\bar{x}}}$ the BFGS update of $H_{\bar{x}}$. In Sect. 5.1, we showed how to implement such a procedure efficiently.

5.3 Initial point

The initial point $z^0 = (\bar{x}^0, y^0, \bar{s}^0)$ is required to satisfy $z^0 \in \mathcal{F} \cap \mathcal{N}(\eta)$. We therefore choose some $\bar{x}^0 \in \bar{\mathcal{K}}^\circ$ and set $\bar{s}^0 = -g_{\bar{x}^0}$. We then get $\bar{\nu}\mu(z^0) = (\bar{x}^0)^T\bar{s}^0 = -(\bar{x}^0)^T g_{\bar{x}^0} \overset{(20)}{=} \bar{\nu}$ 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 $(\bar{x}, y, \bar{s}) = (x, \tau, y, s, \kappa)$ that satisfies the bounds in Theorem 1 solves to ϵ-accuracy the homogeneous model (HSD). However, we are interested 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. Consider the following inequalities:

$$\|Ax - \tau b\|_\infty \leq \epsilon \cdot \max\{1, \|[A, b]\|_\infty\} \qquad \text{(P)}$$

$$\|A^T y + s - c\tau\|_\infty \leq \epsilon \cdot \max\left\{1, \left\|\left[A^T, I, -c\right]\right\|_\infty\right\} \qquad \text{(D)}$$

$$\left|-c^T x + b^T y - \kappa\right| \leq \epsilon \cdot \max\left\{1, \left\|\left[-c^T, b^T, 1\right]\right\|_\infty\right\} \qquad \text{(G)}$$

$$\left|c^T x/\tau - b^T y/\tau\right| \leq \epsilon \cdot \left(1 + \left|b^T y/\tau\right|\right) \qquad \text{(A)}$$

$$\tau \leq \epsilon \cdot 10^{-2} \cdot \max\{1, \kappa\} \qquad \text{(T)}$$

$$\tau \leq \epsilon \cdot 10^{-2} \cdot \min\{1, \kappa\} \qquad \text{(K)}$$

$$\mu \leq \epsilon \cdot 10^{-2} \cdot \mu^0 \qquad \text{(M)}$$

We then terminate and conclude as follows:

(OPT)	(P) ∧ (D) ∧ (A) ⇒	Feas. and approx. optimal solution found
(INFEAS)	(P) ∧ (D) ∧ (G) ∧ T ⇒	Problem nearly primal or dual infeasible
(ILLP)	(K) ∧ (M) ⇒	Problem deemed ill-posed

In case (OPT), the approximately optimal solution $(x, y, s)/\tau$ is returned. If we find (INFEAS), the problem is deemed 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.

For all test 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 $\mathbb{R}_+$. This limitation to cones of such low dimension implies simple expressions for the barrier function and its gradient and Hessian. As we shall see, it does impose any restrictions on which problems can be formulated. See also [15] for further discussion on this topic.

The notation used in this section is independent of previous sections.

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

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

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

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

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 [15] 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 [14], it is shown that this is equivalent to

$$\min_{x,y,t} \quad t, \quad \text{s.t.} \quad Ax = b, \quad e^T y = t$$

$$(x_j; y_j; t) \in K_{(1/p)}, \quad j = 1, \ldots, M.$$

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

$$\min_{z} \sum_{j=1}^{M} a_j \|z - C^{(j)}\|_{p_j} \tag{17}$$

where $a_j \geq 0$ are the weights. We can then formulate (17) in conic form:

$$\min_{z^+, z^-, v, w, u} \sum_{j=1}^{M} a_j u_1^{(j)}$$

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

6.2.3 Geometric programming

This is a problem of the type

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

where $g^{(j)}$ are monomials and $f^{(j)}$ are posynomials. Using the notation $x^v := \prod_{i=1}^{n} x_i^{v_i}$ where each $x_i > 0$, they can be writting

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

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

- the matrix $A^{(j)} := \left(a_1^{(j)}, a_2^{(j)}, \ldots, a_{N_j}^{(j)}\right)^T \in \mathbb{R}^{N_j \times N}$,
- the vector $d^{(j)} = (d_1^{(j)}, \ldots, d_{N_j}^{(j)})^T \in \mathbb{R}^{N_j \times 1}$ and
- the vector $c^{(j)} = \log(d^{(j)}) = (\log(d_1), \ldots, \log(d_{N_j}))^T \in \mathbb{R}^{N_j \times 1}$

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

- the vector $b^{(j)}$, the scalar $k^{(j)}$, the scalar $h^{(j)} = \log(k^{(j)})$.

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

$$
\min_{\boldsymbol{u}_+, \boldsymbol{u}_-, \boldsymbol{w}, \boldsymbol{v}, \boldsymbol{y}, t^{(0)}} \quad t^{(0)}
$$

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

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

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

$$
\begin{aligned}
\min_x \quad & \sum_{j=1}^{N} d_j x_j \log x_j \\
\text{s.t.} \quad & Ax = b \\
& x_j \geq 0, \quad j = 1, \ldots, N
\end{aligned}
$$

which can be formulated as

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

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

Table 1 Parameters used in computational experiments

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

our algorithm to `SeDuMi` (see [22]) when called through `CVX` (see [8]). We intention-
ally compare only the number of Cholesky factorizations performed by each algorithm.
This is to eliminate from the comparisons the CPU-time consumed by software over-
head. Therefore, it is reasonable to measure only the dominating operations, i.e. the
Cholesky 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 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

Table 2 Computational results for *p*-cone problems

Problem		NPC						CVX/SeDuMi			
Name and size	*p*	*m*	*n*	it	ch	ce	st	*m*	*n*	ch	st
bandm	1.13	777	1,416	9	19	1.1	opt	6,913	1,4632	21	opt
M = 305	1.57	777	1,416	11	23	1.1	opt	8,801	18,408	26	opt
N = 472	2.09	777	1,416	14	29	1.1	opt	9,745	20,296	27	opt
sp(*A*) = 1.73 %	4.71	777	1,416	23	37	0.6	opt	10,689	22,184	26	opt
	7.39	777	1,416	24	43	0.8	opt	11,633	24,072	26	opt
blend	1.13	188	342	9	19	1.1	opt	1,670	3,534	21	opt
M = 74	1.57	188	342	9	20	1.2	opt	2,126	4,446	22	opt
N = 114	2.09	188	342	9	16	0.8	opt	2,354	4,902	20	opt
sp(*A*) = 6.19 %	4.71	188	342	11	19	0.7	opt	2,582	5,358	20	opt
	7.39	188	342	13	21	0.6	opt	2,810	5,814	21	opt
⋮				*More results online*							
stocfor1	1.13	282	495	9	16	0.8	opt	2,427	5,115	19	opt
M = 117	1.57	282	495	8	17	1.1	opt	3,087	6,435	20	opt
N = 165	2.09	282	495	9	19	1.1	opt	3,417	7,095	22	opt
sp(*A*) = 2.60 %	4.71	282	495	18	30	0.7	opt	3,747	7,755	25	opt
	7.39	282	495	22	29	0.3	opt	4,077	8,415	26	opt

Data $A \in \mathbb{R}^{M \times N}$ and *b* from NETLIB. sp(*A*) denotes the sparsity of *A*. This table contains only a part of
the instances tested. Full results can be found in the electronic supplements

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 [15] is used to solve the same kind of problem. Overall, our algorithm performs better, both in terms of iterations and factorizations.

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

Table 3 Results for facility location problems

Problem				NPC		
N	M	ν	$\bar{p}$	it	ch	ce
3	4	44	2.03	11.1	18.2	0.65
10	4	128	2.07	13.2	20.1	0.54
3	20	220	2.09	17.1	27.5	0.64
19	4	236	2.00	13.8	21.0	0.54
		⋮ *More results online*				
10	12	384	2.06	16.0	25.1	0.58
32	4	392	2.03	13.4	20.9	0.56
10	20	640	1.99	18.7	30.5	0.66
19	20	1,180	2.01	19.7	30.5	0.60
32	20	1,960	1.98	17.7	31.5	0.79

The algorithm always terminated after reaching optimality as all problem instances were feasible by construction. This table contains only a part of the instances tested. Full results can be found in the electronic supplements

Table 4 Results for geometric programs

Problem			NPC				mskgpopt	
Name	n	dod	it	ch	ce	st	ch	st
beck751	7	10	16	30	0.9	opt	18	opt
beck753	7	10	13	27	1.1	opt	10	opt
car	37	104	15	28	0.9	opt	46	opt
demb761	11	19	12	22	0.8	ipr	10	opt
				$\vdots$ *More results online*				
demb781	2	1	7	10	0.4	opt	7	opt
fang88	11	16	9	18	1.0	opt	11	opt
jha88	30	274	17	34	1.0	opt	13	opt
mra01	61	844	16	30	0.9	opt	58	opt
mra02	126	3,494	30	57	0.9	opt	53	opt
rijc786	8	3	9	16	0.8	opt	6	opt
rijc787	7	40	12	23	0.9	opt	36	opt

This table contains only a part of the instances tested. Full results can be found in the electronic supplements

of iterations and Cholesky factorization respectively used by algorithm $\mathcal{A}$ to solve instance j and let us define the ratio of sums $\mathcal{S} = (\sum_j C_j^{\text{NPC}})/(\sum_j C_j^{\text{NPC}})$. Further let $\mathcal{R}_j^{\text{it}} = I_j^{\text{NPC}}/I_j^{\text{MOSEK}}$ and $\mathcal{R}_j^{\text{ch}} = C_j^{\text{NPC}}/C_j^{\text{MOSEK}}$. If we let an overbar denote arithmetic mean and a tilde denote geometric mean over all j, we then find

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

For these problems we therefore conclude that our algorithm performs somewhat inferiorly to MOSEK, using less iterations but cumulatively 30 % more Cholesky 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 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 unfavourable to MOSEK. With the notation defined in Sect. 6.3.3, we find

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

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

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
ento22	794	28	28	60	1.1	ipr	14	ipr
ento21	931	28	55	112	1.0	ipr	18	ipr
a_tb	1,127	25	38	87	1.3	opt	97	opt
ento23	1,563	28	34	73	1.1	ipr	14	ipr
			$\vdots$	*More results online*				
a_35	4,333	37	43	90	1.1	ipr	18	ipr
a_24	5,162	37	36	90	1.5	ipr	23	ipr
ento3	5,172	28	49	126	1.6	opt	146	opt
ento50	5,172	28	49	126	1.6	opt	146	opt
a_46	9,455	37	40	102	1.6	ipr	20	ipr
a_56	9,702	37	65	158	1.4	opt	123	opt
ento25	10,142	28	116	250	1.2	opt	149	opt
entodif	12,691	40	50	130	1.6	opt	155	opt
ento48	15,364	31	16	52	2.2	opt	47	opt

This table contains only a part of the instances tested. Full results can be found in the electronic supplements

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 and not modelled as conic optimization problems. Our algorithm, on the other hand, uses a purely conic formulation and thus is a much more general purpose algorithm. We use no particular tuning of parameters to particular problems. 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 across various problem types, 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. Inspired by techniques known to significantly improve efficiency of algorithms for self-scaled cones, we have developed

techniques similar in purpose but for the non-symmetric case. These include quasi-Newton updating to reduce computational load and a Runge–Kutta type second order search direction, which is new in this context. We demonstrated 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 Sect. 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 [15].
– For geometric programs, our algorithm compares somewhat unfavourable to MOSEK.
– For entropy maximization problems, our algorithm again compares somewhat unfavourable to MOSEK.

The computational results comparing our algorithm to MOSEK should, however, be seen in the light of the comments in Sect. 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 separate correction phase. Nesterov remarks in [15] that this process can be seen as the process of finding a scaling point, i.e. a point w such that $x = \nabla^2 F(w)s$. 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 the experiments with our algorithm that rarely more a couple of correction steps are needed, some or all of which may be comparably inexpensive quasi-Newton steps.

Appendix 1: 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 [17,18].

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 [17,18], 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

$$\|s\|_{-g_x}^* = \|(H_{-g_x}^*)^{-1/2}s\| = \|H_x^{1/2}s\| = \|s\|_x^*. \tag{18}$$

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

$$H_x x = -g_x \tag{19}$$
$$x^T g_x = -\nu \tag{20}$$
$$\|x\|_x^2 = \nu. \tag{21}$$

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

$$x \in \mathcal{K}^\circ \Rightarrow W(x) = \{u, \|u - x\|_x \le 1\} \subseteq \mathcal{K} \tag{22}$$
$$s \in (\mathcal{K}^*)^\circ \Rightarrow W^*(s) = \{h, \|h - s\|_s^* \le 1\} \subseteq K^*. \tag{23}$$

Appendix 2: The homogeneous and self-dual model

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 $0 > -b^T y = -\bar{x}^T A^T y = \bar{x}^T s \ge 0$, a contradiction. We can argue similarly if $c^T x < 0$,

and this completes the proof of Lemma 1.

Self-duality

The dual of (HSD) problem is

$$\max_{\hat{y}_1,\hat{y}_1,\hat{y}_1,\hat{y}_1,\hat{y}_1,\hat{y}_1,\hat{s}} \quad 0$$

$$\text{s.t.} \quad \begin{pmatrix} A^T & 0 & -c \\ 0 & c^T & -b^T \\ 0 & -I & 0 \\ -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \end{pmatrix} + \begin{pmatrix} \hat{s}_1 \\ \hat{s}_2 \\ \hat{s}_3 \\ \hat{s}_4 \end{pmatrix} = 0 \tag{24}$$

$$- A\hat{y}_2 + b\hat{y}_3 = 0 \tag{25}$$

$$\hat{s} \in (\mathcal{K} \times \mathbb{R}_+ \times \mathcal{K}^* \times \mathbb{R}_+)^*, \quad \hat{y} \text{ free.} \tag{26}$$

After a few eliminations, we see that (24)–(26) are equivalent to

$$\begin{array}{rrrr} A\hat{s}_3 & -b\hat{s}_4 & & = 0 \\ -A^T\hat{y}_1 & & +c\hat{s}_4 & -\hat{s}_1 = 0 \\ b^T\hat{y}_1 & -c^T\hat{s}_3 & & -\hat{s}_2 = 0 \end{array} \tag{27}$$

$$(\hat{s}_3, \hat{s}_4) \in \mathcal{K} \times \mathbb{R}_+, \quad (\hat{s}_1, \hat{s}_2) \in \mathcal{K}^* \times \mathbb{R}_+, \quad \hat{y}_1 \in \mathbb{R}^m.$$

Through the following identification of variables

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

it is clear that the constraints (27) are equivalent to those of the problem (HSD). Since the objective function in both problems is constant zero, the two problems are identical and this proves Lemma 2.

Appendix 3: Prediction

The direction d_z is defined by

$$G(d_y; d_{\bar{x}}) - (0; d_{\bar{s}}) = -(G(y; \bar{x}) - (0; \bar{s})) \tag{28}$$

$$d_{\bar{s}} + \mu H_{\bar{x}} d_{\bar{x}} = -\bar{s} \tag{29}$$

Reduction of residuals

We first show:

$$1. \ \bar{s}^T d_{\bar{x}} + \bar{x}^T d_{\bar{s}} + \bar{x}^T \bar{s} = \psi(z)^T d_{\bar{x}} \tag{30}$$

$$2. \ (\bar{x} + d_{\bar{x}})^T (\bar{s} + d_{\bar{s}}) = 0 \tag{31}$$

$$3. \ d_{\bar{x}}^T d_{\bar{s}} = -\psi(z)^T d_{\bar{x}}. \tag{32}$$

1. We get $\bar{s}^T d_{\bar{x}} + \bar{x}^T d_{\bar{s}} + \bar{x}^T \bar{s} \overset{(29)}{=} \bar{s}^T d_{\bar{x}} + \bar{x}^T (-\bar{s} - \mu H_{\bar{x}} d_{\bar{x}}) + \bar{x}^T \bar{s}$, which, after reduction, gives $d_{\bar{x}}^T (\bar{s} - \mu H_{\bar{x}} \bar{x}) = \psi(z)^T d_{\bar{x}}$.
2. Equation (28) is equivalent to $G(y + d_y; \bar{x} + d_{\bar{x}}) - (0; \bar{s} + d_{\bar{s}}) = 0$. Pre-multiplying this equation by $(y + d_y, \bar{x} + d_{\bar{x}})$ 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:

$$
\begin{aligned}
\bar{\nu}\mu(z^+) &= (\bar{x} + \alpha d_{\bar{x}})^T (\bar{s} + \alpha d_{\bar{s}}) \\
&= \bar{x}^T \bar{s} + \alpha(\bar{s}^T d_{\bar{x}} + \bar{x}^T d_{\bar{s}}) + \alpha^2 d_{\bar{x}}^T d_{\bar{s}} \\
&\overset{(30) - (32)}{=} \bar{x}^T \bar{s} + \alpha(-\bar{x}^T \bar{s} + \psi(z)^T d_{\bar{x}}) + \alpha^2(-\psi(z)^T d_{\bar{x}}) \\
&= (1 - \alpha)\bar{x}^T \bar{s} + \alpha(1 - \alpha)\psi(z)^T d_{\bar{x}}
\end{aligned}
$$

which after division by $\bar{\nu}$ proves Lemma 3.

Bounds on $\bar{s}$, $d_{\bar{s}}$ and $d_{\bar{x}}$

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

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

where we used (21) and $\bar{x}^T\psi = 0$.

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

$$
\begin{aligned}
(\|d_{\bar{s}}\|_{\bar{x}}^*)^2 + \mu^2\|d_{\bar{x}}\|_{\bar{x}}^2 &= (\|\bar{s}\|_{\bar{x}}^*)^2 - 2\mu d_{\bar{x}}^T d_{\bar{s}} \overset{(32)}{=} (\|\bar{s}\|_{\bar{x}}^*)^2 + 2\mu d_{\bar{x}}^T\psi \\
&\leq (\|\bar{s}\|_{\bar{x}}^*)^2 + 2\mu\|d_{\bar{x}}\|_{\bar{x}}\|\psi\|_{\bar{x}}^*
\end{aligned}
$$

by the Cauchy–Schwarz inequality. Therefore: $\mu^2\|d_{\bar{x}}\|_{\bar{x}}^2 \leq (\|\bar{s}\|_{\bar{x}}^*)^2 + 2\mu\|d_{\bar{x}}\|_{\bar{x}}\|\psi\|_{\bar{x}}^*$. Now subtracting $2\mu\|d_{\bar{x}}\|_{\bar{x}}\|\psi\|_{\bar{x}}^*$ and adding $(\|\psi\|_{\bar{x}}^*)^2$ to both sides, we get

$$
\left(\mu\|d_{\bar{x}}\|_{\bar{x}} - \|\psi\|_{\bar{x}}^*\right)^2 \leq (\|\bar{s}\|_{\bar{x}}^*)^2 + (\|\psi\|_{\bar{x}}^*)^2
$$

or

$$\|d_{\bar{x}}\|_{\bar{x}} \leq \mu^{-1} \left(\|\psi\|_{\bar{x}}^* + \sqrt{(\|\bar{s}\|_{\bar{x}}^*)^2 + (\|\psi\|_{\bar{x}}^*)^2} \right)$$

$$\leq \mu^{-1}(\eta\mu + \sqrt{\mu^2(\eta^2 + \bar{v}) + \eta^2\mu^2}) = \eta + \sqrt{\eta^2 + \bar{v}} =: k_{\bar{x}}. \qquad (34)$$

For $d_{\bar{s}}$, we similarly have

$$(\|d_{\bar{s}}\|_{\bar{x}}^*)^2 \leq (\|\bar{s}\|_{\bar{x}}^*)^2 + 2\mu\|d_{\bar{x}}\|_{\bar{x}}\|d_{\bar{s}}\|_{\bar{x}}^*$$

$$(\|d_{\bar{s}}\|_{\bar{x}}^* - \mu\|d_{\bar{x}}\|_{\bar{x}})^2 \leq (\|\bar{s}\|_{\bar{x}}^*)^2 + \mu^2\|d_{\bar{x}}\|_{\bar{x}}^2$$

$$\|d_{\bar{s}}\|_{\bar{x}}^* \leq k_{\bar{x}}\mu + \sqrt{\mu^2(\eta^2 + \bar{v}) + k_{\bar{x}}^2\mu^2} = k_{\bar{s}}\mu \qquad (35)$$

where $k_{\bar{s}} := k_{\bar{x}} + \sqrt{(\eta^2 + \bar{v}) + k_{\bar{x}}^2}$.

Feasibility of z^+

Define $\alpha_1 := k_{\bar{x}}^{-1} = \Omega(1/\sqrt{\bar{v}})$. Then for any $\alpha \leq \alpha_1$, we have

$$\|\bar{x} - (\bar{x} + \alpha d_{\bar{x}})\|_{\bar{x}} = \alpha\|d_{\bar{x}}\|_{\bar{x}} \overset{(34)}{\leq} \alpha k_{\bar{x}} \leq 1$$

and so from (22), we conclude $\bar{x} + \alpha d_{\bar{x}} = \bar{x}^+ \in \bar{\mathcal{K}}$.

Now, define $\alpha_2 := (1 - \eta)k_{\bar{s}}^{-1} = \Omega(1/\sqrt{\bar{v}})$. Then for $\alpha \leq \alpha_2$, we have

$$\mu^{-1}\|\bar{s}^+ + \mu g_{\bar{x}}\|_{-g_{\bar{x}}}^* = \mu^{-1}\|\bar{s} + \alpha d_{\bar{s}} + \mu g_{\bar{x}}\|_{-g_{\bar{x}}}^* = \mu^{-1}\|\psi + \alpha d_{\bar{s}}\|_{-g_{\bar{x}}}^*$$

$$\overset{(18)}{\leq} \mu^{-1}\|\psi\|_{\bar{x}}^* + \mu^{-1}\alpha\|d_{\bar{s}}\|_{\bar{x}}^* \overset{(35)}{\leq} \eta + \alpha k_{\bar{s}} \leq 1.$$

Since $-g_{\bar{x}} \in \bar{\mathcal{K}}^*$, we have by (23) that $\mu^{-1}\bar{s}^+ \in \bar{\mathcal{K}}^*$ and therefore $\bar{s}^+ \in \bar{\mathcal{K}}^*$. Therefore, Lemma 4 holds with $\alpha = \min\{\alpha_1, \alpha_2\} = \Omega(1/\sqrt{\bar{v}}) = \Omega(1/\sqrt{v})$.

Bound on ψ^+

First recall the definition (6): $\psi(\bar{x}, \bar{s}, t) = \bar{s} + tg_{\bar{x}}$. Now consider for a fixed v_0 the function

$$\Phi_t(\bar{x}) = \bar{x}^T v_0 + tF(\bar{x})$$

which is self-concordant with respect to $\bar{x}$. Define its Newton step by $n_t(\bar{x}) := -\nabla^2\Phi_t(\bar{x})^{-1}\nabla\Phi_t(\bar{x})$. Define also $q = \|n_{t_2}(\bar{x})\|_{\bar{x}}$. From the general theory of self-

concordant functions, the following inequality holds. If $q \leq 1$, then

$$\|n_{t_2}(\bar{x}_2)\|_{\bar{x}_2} \leq \left(\frac{q}{1-q}\right)^2. \tag{36}$$

For a proof of this relation, see e.g. Theorem 2.2.4 in [21]. With $v_0 = \bar{s}^+$, $t_2 = \mu^+$ and $\bar{x}_2 = \bar{x}^+$, the inequality (36) is

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

where $\mu^+ q = \|H_{\bar{x}}^{-1}(\bar{s}^+ + \mu^+ g_{\bar{x}})\|_{\bar{x}} = \|\bar{s}^+ + \mu^+ g_{\bar{x}}\|_{\bar{x}}^*$. From Lemma 3 and (34):

$$|\mu - \mu^+| = |-\alpha\mu + \alpha(1-\alpha)\bar{v}^{-1}\psi^T d_{\bar{x}}|$$
$$\leq \mu\alpha\left(1 + (1-\alpha)\eta k_{\bar{x}}\bar{v}^{-1}\right). \tag{38}$$

By the assumption $\|\psi\|_{\bar{x}}^* \leq \eta\mu$ combined with (34), we have $\psi^T d_{\bar{x}} \geq -\eta k_{\bar{x}}\mu$. Therefore

$$\mu^+ = (1-\alpha)\mu + \alpha(1-\alpha)\bar{v}^{-1}\psi^T d_{\bar{x}}$$
$$\geq \mu(1-\alpha)(1-\alpha\eta k_{\bar{x}}\bar{v}^{-1})$$
$$\mu/\mu^+ \leq (1-\alpha)^{-1}(1-\alpha\eta k_{\bar{x}}\bar{v}^{-1})^{-1} \tag{39}$$

Let us now obtain a bound on q.

$$\mu^+ q = \|\bar{s}^+ + \mu^+ g_{\bar{x}}\|_{\bar{x}}^* = \|\psi - (\mu - \mu^+)g_{\bar{x}} + \alpha d_{\bar{s}}\|_{\bar{x}}^* \tag{40}$$
$$\leq \|\psi\|_{\bar{x}}^* + |\mu - \mu^+|\|g_{\bar{x}}\|_{\bar{x}}^* + \alpha\|d_{\bar{s}}\|_{\bar{x}}^*$$
$$\leq \eta\mu + \mu\alpha\left(1 + (1-\alpha)\eta k_{\bar{x}}\bar{v}^{-1}\right)\sqrt{\bar{v}} + \alpha k_{\bar{s}}\mu$$
$$\leq \mu\left(\eta + \alpha k_{\bar{s}} + \alpha(1 + (1-\alpha)\bar{v}^{-1}\eta k_{\bar{x}})\sqrt{\bar{v}}\right)$$
$$q \leq (\mu/\mu^+)(\eta + \alpha(\sqrt{\bar{v}} + k_{\bar{s}} + \eta k_{\bar{x}}))$$
$$\leq (1-\alpha)^{-1}(1-\alpha\eta k_{\bar{x}}\bar{v}^{-1})^{-1}(\eta + \alpha(\sqrt{\bar{v}} + k_{\bar{s}} + \eta k_{\bar{x}}))$$

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

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

which also implies $q < 1$. Now by (37), we see that (41) implies $\|\psi^+\|_{\bar{x}^+}^* \leq 2\eta\mu^+$ and hence $z^+ \in \mathcal{N}(2\eta)$ which finishes the proof of Lemma 5.

Appendix 4: Correction phase

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

$$G(\delta_y; \delta_{\bar{x}}) - (0; \delta_{\bar{s}}) = 0 \tag{42}$$

$$\delta_{\bar{s}} + \mu H_{\bar{x}} \delta_{\bar{x}} = -\psi(\bar{x}, \bar{s}, \mu) \tag{43}$$

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

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

and therefore

$$\|\delta_{\bar{x}}\|_{\bar{x}} \leq \beta, \quad \|\delta_{\bar{s}}\|_{\bar{x}}^* \leq \beta\mu. \tag{44}$$

From (43), we also have

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

Where we used (44). Now define $q = (\mu^+)^{-1}\|\bar{s}^+ + \mu^+ g_{\bar{x}}\|_{\bar{x}}^*$. Then estimating similarly to (40), we get

$$\begin{aligned}
\mu^+ q &\leq \|\psi(\bar{x}, \bar{s}, \mu) + (\mu^+ - \mu)g_{\bar{x}} + \hat{\alpha}\delta_{\bar{s}}\|_{\bar{x}}^* \\
&\leq \beta\mu(1 + \hat{\alpha}(\beta\bar{v}^{-1/2} + 1))
\end{aligned}$$

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

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

so that altogether

$$q \leq \beta(1 - \hat{\alpha}\bar{v}^{-1}\beta^2)^{-1}(1 + \hat{\alpha}(\beta\bar{v}^{-1/2} + 1)). \tag{46}$$

Now we can apply the theorem (36) with $v_0 = \bar{s}^+$, $t = \mu$ and $\bar{x}_2 = \bar{x}^+$:

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

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

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

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

Appendix 5: Algorithm complexity

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

$$\|G(y; \bar{x}) - (0; \bar{s})\| \le \epsilon \|G(y^0; \bar{x}^0) - (0; \bar{s}^0)\|$$

in $\mathcal{O}(\sqrt{\bar{v}}\log(1/\epsilon)) = \mathcal{O}(\sqrt{v}\log(1/\epsilon))$ 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

$$
\begin{aligned}
\mu(z^+) &\le (1 - \alpha)\mu(z) + \alpha(1 - \alpha)\bar{v}^{-1}\mu\eta k_{\bar{x}} \\
&\le \mu(z)(1 - \alpha)(1 + \alpha\eta k_{\bar{x}}\bar{v}^{-1}) \\
&= \mu(z)(1 - \Omega(1/\sqrt{\bar{v}}))
\end{aligned}
\tag{48}
$$

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

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

Therefore

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

and hence

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

which shows that also $\mu(z)$ is decreased geometrically with a rate of $(1 - \Omega(1/\sqrt{\bar{v}}))$. Therefore $\mu(z) \le \epsilon\mu(z^0)$ in $\mathcal{O}(\sqrt{\bar{v}}\log(1/\epsilon)) = \mathcal{O}(\sqrt{v}\log(1/\epsilon))$ 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. (eds.) High Performance Optimization, pp. 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, 2nd edn. Wiley, New York (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 (2010). http://cvxr.com/cvx
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. www.mosek.com
14. Nesterov, Y.E.: Constructing self-concordant barriers for convex cones. CORE Discussion Paper (2006/30)
15. Nesterov, Y.E.: Towards nonsymmetric conic optimization. Optim. Method Softw. **27**, 893–917 (2012)
16. Nesterov, Y.E., Nemirovski, A.S.: Interior-Point Polynomial Algorithms in Convex Programming. SIAM, Philadelphia (1994)
17. Nesterov, Y.E., Todd, M.J.: Self-scaled barriers and interior-point methods for convex programming. Math. Oper. Res. **22**, 1–42 (1997)
18. Nesterov, Y.E., Todd, M.J.: Primal–dual interior-point methods for self-scaled cones. SIAM J. Optim. **8**, 324–364 (1998)
19. 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)
20. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, Berlin (2006)
21. Renegar, J.: A Mathematical View of Interior-Point Methods in Convex Optimization. SIAM, Philadelphia (1987)
22. Sturm, J.F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optim. Method Softw. **12**, 625–653 (1999)
23. Sturm, J.F.: Implementation of interior point methods for mixed semidefinite and second order cone optimization problems. Optim. Method Softw. **17**, 1105–1154 (2002)
24. Tuncel, L.: Primal–dual symmetry and scale invariance of interior-point algorithms for convex optimization. Math. Oper. Res. **23**, 708–718 (1998)
25. Tuncel, L.: Generalization of primal–dual interior-point methods to convex optimization problems in conic form. Found. Comput. Math. **1**, 229–254 (2001)
26. 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)
27. Xue, G., Ye, Y.: An efficient algorithm for minimizing a sum of p-norms. SIAM J. Optim. **10**, 551–579 (1999)