A semi-smooth Newton method for the nonlinear conic problem with generalized simplicial cones

Nicolas F. Armijo * Yunier Bello-Cruz† Gabriel Haeser ‡

April 17, 2026

Abstract

In this work we develop and analyze a semi-smooth Newton method for the general nonlinear conic programming problem. In particular, we study the problem with a generalized simplicial cone, i.e., the image of a symmetric cone under a linear mapping. We generalize Robinson's normal equations to a conic setting, yielding what we call the conic projection equations. The resulting system is equivalent to the KKT conditions associated with the nonlinear conic programming problem. A semi-smooth Newton iteration is proposed for solving it, and local quadratic convergence is established. We study properties of generalized simplicial cones and prove strong semi-smoothness of the projection operator onto them. Numerical experiments compare the method against a recent smoothing Newton approach on the circular cone programming problem, and we also apply it to the low-rank matrix completion problem.

Keywords: Conic programming, nonlinear programming, second-order cone programming, circular cone programming, Robinson's normal equations, semi-smooth Newton method.

2010 AMS Subject Classification: 90C33, 15A48.

1 Introduction

In this paper we study the general nonlinear conic programming (NCP) problem:

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g(x) \in \mathcal{K}, \end{array} \tag{1.1}$$

where $\mathbb{X}$ and $\mathbb{Y}$ are finite-dimensional inner product spaces, $\mathcal{K} \subset \mathbb{Y}$ is a closed convex cone, and $f : \mathbb{X} \to \mathbb{R}$, $g : \mathbb{X} \to \mathbb{Y}$ are twice continuously differentiable functions. The nonlinear conic programming problem has been widely studied over the last three decades, with significant contributions in the areas of optimality conditions, constraint qualifications, and algorithms for particular, yet important, instances. See [11, 30].

Our approach is based on what we call *conic projection equations*, which extend Robinson's normal equations [36]. Robinson formulated these equations in the context of complementarity conditions for minimizing a function f over a closed convex set C; they served as the foundation

*Department of Applied Mathematics, State University of Campinas, Brazil (e-mail: nfarmijo@ime.unicamp.br). The author was supported by Fapesp grant 2019/13096-2.

†Northern Illinois University, USA (e-mail: yunierbello@niu.edu). The author was partially supported by the NSF Grant DMS-2307328 and by an internal grant from NIU.

‡Department of Applied Mathematics, University of São Paulo, Brazil (e-mail: ghaeser@ime.usp.br). The author was supported by CNPq and Fapesp grant 2018/24293-0.

for algorithms targeting variational inequalities, see, for example, [34] for box-constrained problems and [45] for polyhedral convex set constrained problems.

In the nonlinear conic programming setting, analogous projection-based systems were studied in [7], where the authors addressed the quadratic conic programming problem in the particular case of the nonnegative cone. They employed a piecewise linear system to locate KKT points and a semi-smooth Newton method to solve it; a related approach was developed in [5] for a special piecewise linear system arising from positively constrained convex quadratic programming. This formulation is also closely related to the absolute value equation (AVE); see, for example, [7]. A more recent contribution appears in [3], where Gauss–Seidel-type and Jacobi-type methods were proposed for solving such piecewise linear systems. In the same vein, [8] extended the nonlinear system to the second-order cone, addressing the quadratic second-order cone programming problem via a semi-smooth Newton method with global convergence guarantees under assumptions on the Hessian of the objective function. For the convex quadratic semidefinite programming problem, the authors of [27] proposed a two-phase augmented Lagrangian method, resulting in the MATLAB package QSDPNAL. The earlier work [35] studied the same problem and, based on an equivalent formulation, proposed a Newton method. More recently, [4] addressed the non-convex quadratic conic programming problem, focusing on the nearest correlation matrix problem and proposing a locally quadratically convergent semi-smooth Newton method using a different, easy-to-compute subdifferential.

Over time, many algorithms have been developed for the various structures arising in conic programming. Interior-point methods, such as [22] for semidefinite programming, are classical in the area. Their efficiency, particularly for linear programming, has made them a sustained area of research; see, for example, [20] for recent developments. Augmented Lagrangian (AL) methods offer an alternative approach: they approximately minimize a penalized version of the classical Lagrangian by solving a subproblem involving the constraint function, followed by updates of the multiplier and penalty parameters. Notable AL methods related to our work include SDPNAL+ [42] and QSDPNAL [27], which have demonstrated efficiency for linear and quadratic semidefinite programming. Augmented Lagrangian methods remain an active area of research, receiving significant attention in nonlinear programming, conic programming [18, 31, 21], and Riemannian optimization [2].

The methods most directly related to this work are Newton-type methods [25], which solve the primal-dual complementarity optimality conditions by approximating the zeros of a nonlinear system of equations. A well-known variant employs smoothing functions to approximate the complementarity conditions, with the properties of these functions ensuring that solutions of the resulting nonlinear system correspond to KKT points of the original problem. Smoothing methods have been proposed recently for second-order cone programming [40, 41, 14] and semidefinite programming [28]. Additionally, [24] studied a semi-smooth Newton method for the nearest correlation matrix problem, and [43] proposed a Lagrange–Newton algorithm for sparse nonlinear programming.

In this paper, we extended Robinson's normal equations to the setting of nonlinear conic programming, yielding a system of *conic projection equations* that is equivalent to the KKT conditions of problem (1.1) (Theorem 3.1). In contrast to smoothing-based approaches, our formulation does not employ a smoothing function; instead, it encodes complementarity through a direct nonlinear system involving the conic projection.

We study the generalized simplicial cone $M\mathcal{K}$, defined as the image of a symmetric cone $\mathcal{K}$ under a linear mapping M, and establish the strong semi-smoothness of the projection onto $M\mathcal{K}$ (Theorem 2.11), including the case where M is rank-deficient. We also derive an explicit characterization of the dual cone $(M\mathcal{K})^*$ (Proposition 2.9) and provide sufficient conditions for the closedness of $M\mathcal{K}$ (Propositions 2.6 and 2.7). Building on these results, we propose a semi-smooth Newton method (Algorithm 1) for solving the conic projection equations, incorporating a regularization strategy for singular Jacobians and an escape mechanism for non-strongly stationary points. We prove local quadratic convergence (Theorem 4.6).

We evaluate the proposed method on two classes of problems: circular cone programming and low-rank matrix completion. Second-order cone programming (SOCP) has numerous applications in areas such as control, signal processing, and finance, and its structure, variants, and solution methods have been extensively studied; see, for example, [12, 13, 19, 29, 38, 44]. Our focus is on circular cone programming, a natural generalization of SOCP, and specifically on Newton-type methods, in particular the smoothing method of [40]. The semi-smooth Newton method developed here achieves comparable or better precision than [40], while being up to two orders of magnitude faster.

The low-rank matrix completion problem is intrinsically challenging due to the combinatorial nature of the rank constraint, which significantly complicates the design of high-precision algorithms. Our interest in this problem was motivated by the reformulation of [9], which replaces the rank constraint with continuous matricial constraints, enabling a purely continuous optimization approach. Leveraging this reformulation, we apply the proposed semi-smooth Newton method and compute high-precision KKT points for more than 94% of the tested instances.

The paper is organized as follows. In Section 2, we present preliminary results, including properties of generalized simplicial cones. Section 3 reviews the nonlinear conic programming problem and introduces the generalized system of projection equations, establishing its equivalence with the KKT conditions. In Section 4, we develop the semi-smooth Newton method and analyze its convergence properties. Finally, in Section 5, we report numerical experiments on circular cone programming and low-rank matrix completion.

2 Preliminaries

Let $\mathbb{X}$ and $\mathbb{Y}$ be finite-dimensional normed vector spaces equipped with an inner product, which we indistinctly denote by $\langle \cdot, \cdot \rangle$. Let Id denote the identity operator. Given a linear operator $M : \mathbb{X} \to \mathbb{Y}$, its operator norm is defined by $\|M\| := \sup\{\|Mx\| : x \in \mathbb{X},\ \|x\| = 1\}$, and its adjoint $M^* : \mathbb{Y} \to \mathbb{X}$ is the unique linear operator satisfying $\langle Mx, y \rangle = \langle x, M^*y \rangle$ for all $x \in \mathbb{X}$ and $y \in \mathbb{Y}$. We say that M is positive semidefinite (resp. positive definite) if $\langle Mx, x \rangle \geq 0$ (resp. > 0) for all $x \in \mathbb{X}$. We denote by $\mathbb{S}^n$ the set of $n \times n$ real symmetric matrices, by $\mathbb{S}^n_+$ the subset of positive semidefinite matrices, and by $\mathbb{L}^n$ the second-order cone. For a fixed cone $\mathcal{K} \subset \mathbb{X}$, the dual and polar cones are defined by

$$\mathcal{K}^* := \{y \in \mathbb{X} : \langle y, x \rangle \geq 0,\ \forall\, x \in \mathcal{K}\} \quad \text{and} \quad \mathcal{K}^\circ := \{y \in \mathbb{X} : \langle y, x \rangle \leq 0,\ \forall\, x \in \mathcal{K}\} = -\mathcal{K}^*,$$

respectively.

Lemma 2.1 (Spectral properties of symmetric matrices; see e.g. [23, Ch. 4])**.** *Let $E \in \mathbb{S}^n$ and let $\lambda_{\min}(E)$, $\lambda_{\max}(E)$ denote the smallest and largest eigenvalues of E, respectively. Then:*

1. *$x^T E x \leq \lambda_{\max}(E)\, x^T x$ for all $x \in \mathbb{R}^n$.*

2. *If E is positive definite, then $\lambda_{\max}(E^{-1}) = 1/\lambda_{\min}(E) > 0$.*

The projection of a point x onto a closed convex set $\mathcal{K}$ is defined by $\Pi_{\mathcal{K}}(x) := \operatorname{argmin}\{\|y-x\| : y \in \mathcal{K}\}$. For a function f, we denote by D_f the set of points where f is differentiable, by $Df(x)$ its differential at x, and by $f'(x)$ its Jacobian. The Clarke generalized Jacobian is denoted by $\partial_C f(x)$ and defined as

$$\partial_C f(x) := \operatorname{conv}\left\{\lim_{k\to\infty} f'(x_k) : x_k \to x,\ x_k \in D_f\right\}.$$

Throughout this work, we write $V_{\mathcal{K}}(x)$ for a generic element of $\partial_C \Pi_{\mathcal{K}}(x)$ and $V_f(x)$ for a generic element of $\partial_C f(x)$.

A key notion for our analysis is the strong semi-smoothness of a function. A Lipschitz function $f : \mathbb{X} \to \mathbb{Y}$ is said to be *strongly semi-smooth* at x if

$$f(x+h) - f(x) - V_f(x+h)\, h = O(\|h\|^2)$$

for all $h \in \mathbb{X}$ and all $V_f(x+h) \in \partial_C f(x+h)$. We say that f is strongly semi-smooth if it is strongly semi-smooth at every $x \in \mathbb{X}$.

The following two classical results are used in our analysis.

Theorem 2.2 (Contraction mapping principle [33, Thm. 8.2.2, p. 153])**.** *Let $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ be a function. Suppose there exists $\lambda \in [0,1)$ such that $\|\Phi(y) - \Phi(x)\| \le \lambda\,\|y-x\|$ for all $x, y \in \mathbb{R}^n$. Then there exists a unique $\bar{x} \in \mathbb{R}^n$ such that $\Phi(\bar{x}) = \bar{x}$.*

Theorem 2.3 (Mean value theorem [15, Prop. 2.6.5, p. 72])**.** *Let $\mathbb{X}, \mathbb{Y}$ be finite-dimensional normed spaces and let $f : \mathbb{X} \to \mathbb{Y}$ be a Lipschitz function. Then,*

$$f(y) - f(x) \in \operatorname{conv}(\partial_C f([x,y]) \cdot (y-x)),$$

that is, $f(y) - f(x) = h$ for some h in the convex hull of the set $\{V(y-x) : V \in \partial_C f(z),\ z = tx + (1-t)y,\ t \in [0,1]\}$.

We conclude this section by presenting key properties of the projection onto a closed convex cone.

Theorem 2.4 (Properties of the conic projection and its generalized Jacobian; see e.g. [4, Thm. 2.1])**.** *Let $\mathbb{X}$ be a finite-dimensional normed vector space and let $\mathcal{K} \subset \mathbb{X}$ be a closed convex cone. Then the projection $\Pi_{\mathcal{K}}$ is differentiable almost everywhere, and both the Jacobian $\Pi'_{\mathcal{K}}(x)$ and any element $V_{\mathcal{K}}(x) \in \partial_C \Pi_{\mathcal{K}}(x)$ is a self-adjoint positive semidefinite linear operator satisfying the following properties:*

1. *$\|V_{\mathcal{K}}(x)\| \le 1$ for all $V_{\mathcal{K}}(x) \in \partial_C \Pi_{\mathcal{K}}(x)$ and all $x \in \mathbb{X}$.*
2. *$\Pi'_{\mathcal{K}}(x)\, x = \Pi_{\mathcal{K}}(x)$ for all $x \in D_{\Pi_{\mathcal{K}}}$.*
3. *$V_{\mathcal{K}}(x)\, x = \Pi_{\mathcal{K}}(x)$ for all $V_{\mathcal{K}}(x) \in \partial_C \Pi_{\mathcal{K}}(x)$.*
4. *The eigenvalues of $\Pi'_{\mathcal{K}}(x)$ satisfy*

   $$0 \le \lambda_{\min}(\Pi'_{\mathcal{K}}(x)) \le \lambda_{\max}(\Pi'_{\mathcal{K}}(x)) \le 1.$$

   Since eigenvalues are continuous functions of the matrix entries, the same bounds hold for any $V_{\mathcal{K}}(x) \in \partial_C \Pi_{\mathcal{K}}(x)$:

   $$0 \le \lambda_{\min}(V_{\mathcal{K}}(x)) \le \lambda_{\max}(V_{\mathcal{K}}(x)) \le 1.$$

Lemma 2.5 (Linearization error bound for the conic projection; [4, Lem. 2.1])**.** *Let $x, y \in \mathbb{X}$ and $V_{\mathcal{K}}(x) \in \partial_C \Pi_{\mathcal{K}}(x)$. Then*

$$\|\Pi_{\mathcal{K}}(y) - \Pi_{\mathcal{K}}(x) - V_{\mathcal{K}}(x)(y-x)\| \le \|y-x\|.$$

2.1 Generalized simplicial cones

In this section we study properties of the generalized simplicial cone, that is, cones in the form $M\mathcal{K}$, where $M\colon \mathbb{X} \to \mathbb{Y}$ is a linear mapping and $\mathcal{K} \subset \mathbb{X}$ is a symmetric cone. A simplicial cone is defined as a cone in which every generator consists solely of extreme rays. However, while finitely generated simplicial cones are typically represented as $M\mathbb{R}^n_+$, we use the term "generalized" here in the sense that we study a general cone $\mathcal{K}$ instead of $\mathbb{R}^n_+$. One motivation for this study is the circular cone programming problem, where the cone constraint is defined by a linear transformation of the second-order cone. To develop methods that utilize projection onto these types of sets, it is essential first to understand the topological properties of the resulting generalized simplicial cone and, most importantly, how to project onto it.

2.1.1 On the closedness of $M\mathcal{K}$

The linearity of M preserves the algebraic properties of $M\mathcal{K}$; however, the topological structure may change. For the projection onto a set to be well defined and unique, a sufficient condition is that the set be closed and convex. While convexity is maintained, the generalized simplicial cone $M\mathcal{K}$ may not always be closed; see, for example, [1, 16], where it is shown that both the second-order and the semidefinite cone can lose closedness under linear transformations. Characterizing when the image of a cone under a linear map remains closed is a subtle question that has received considerable attention; see, for example, [37]. The following two results provide sufficient conditions, each involving the relationship between $\mathrm{Ker}(M)$ and $\mathcal{K}$.

Proposition 2.6 (Closedness of the image of a cone under a linear map; [37, Thm. 9.1])**.** *Let $M : \mathbb{X} \to \mathbb{Y}$ be a linear mapping, let $\mathcal{K} \subset \mathbb{X}$ be a closed convex cone, and let $\hat{\mathcal{K}} := M\mathcal{K}$. If $\mathrm{Ker}(M) \cap \mathcal{K} = \{0\}$, then $\hat{\mathcal{K}}$ is closed.*

The previous result requires that the kernel and the cone intersect trivially. The next proposition shows that, for symmetric cones, a complementary condition, the kernel meeting the interior of the cone, also guarantees closedness, albeit with a stronger conclusion.

Proposition 2.7 (Closedness via interior intersection with the kernel)**.** *Let $\mathcal{K} \subset \mathbb{X}$ be a symmetric cone and let $M\colon \mathbb{X} \to \mathbb{Y}$ be a linear mapping with $\mathrm{rank}(M) = r$. If $\mathrm{Ker}(M) \cap \mathrm{int}(\mathcal{K}) \neq \emptyset$, then $\hat{\mathcal{K}} = \mathrm{Im}(M) \simeq \mathbb{R}^r$. In particular, $\hat{\mathcal{K}}$ is closed.*

Proof. The inclusion $\hat{\mathcal{K}} \subset \mathrm{Im}(M)$ holds by definition. We prove the reverse inclusion. Let $w \in \mathrm{Ker}(M) \cap \mathrm{int}(\mathcal{K})$. Since w lies in the interior of $\mathcal{K}$, there exists $\delta > 0$ such that $B(w, \delta) \subset \mathcal{K}$. Let $v \in \mathbb{X}$ be arbitrary with $v \neq 0$, and set $\varepsilon := \delta/(2\|v\|)$. Then

$$w + \varepsilon v \in B(w, \delta) \subset \mathcal{K} \quad \text{and} \quad w - \varepsilon v \in B(w, \delta) \subset \mathcal{K}.$$

Applying M and using $Mw = 0$, we obtain $\varepsilon Mv \in \hat{\mathcal{K}}$ and $-\varepsilon Mv \in \hat{\mathcal{K}}$. Since $\hat{\mathcal{K}}$ is a cone, it follows that $Mv, -Mv \in \hat{\mathcal{K}}$. As $v \in \mathbb{X}$ was arbitrary, $\mathrm{Im}(M) \subset \hat{\mathcal{K}}$, and therefore $\hat{\mathcal{K}} = \mathrm{Im}(M)$. □

In light of the discussion above, we make the following assumption throughout the remainder of this work.

Assumption 2.8. *The cone $M\mathcal{K}$ is closed.*

2.1.2 Projection onto generalized simplicial cones

In general, projecting onto a set is a difficult task, and many algorithms have been developed for specific cases. Even when existence and uniqueness of the projection are guaranteed, a closed-form expression is often unavailable. Generalized simplicial cones are an important instance of this situation: given a vector x, the goal is to compute or approximate $\Pi_{M\mathcal{K}}(x)$. In previous work, [17] proposed a semi-smooth Newton method for the case $\mathcal{K} = \mathbb{R}^n_+$, and [5] studied a related piecewise linear system arising from positively constrained convex quadratic programming. Here, we extend this approach to a general closed convex cone satisfying Assumption 2.8.

The projection of x onto $M\mathcal{K}$ can be obtained by solving

$$\begin{aligned} \min \quad & \frac{1}{2}\|Mz - x\|^2 \\ \text{s.t.} \quad & z \in \mathcal{K}. \end{aligned} \tag{2.1}$$

If z^* is a solution of (2.1), then $\Pi_{M\mathcal{K}}(x) = Mz^*$. Since the problem is convex and satisfies Slater's condition, its KKT conditions reduce to the nonlinear equation

$$(M^*M - \mathrm{Id})\,\Pi_{\mathcal{K}}(z) + z = M^*x. \tag{2.2}$$

A semi-smooth Newton method for solving (2.2) was proposed in [4]; an alternative Picard iteration was developed in [6].

In the semi-smooth Newton method studied in Section 4, we need to project onto a generalized simplicial cone $M\mathcal{K}$ or onto its dual $(M\mathcal{K})^*$. In general, the dual of a non-symmetric cone may not admit an explicit description; however, for generalized simplicial cones, $(M\mathcal{K})^*$ can be characterized as follows.

Proposition 2.9 (Dual of a generalized simplicial cone). *Let $\mathbb{Z}$ and $\mathbb{Y}$ be finite-dimensional spaces of dimensions m and n, respectively. Let $M : \mathbb{Z} \to \mathbb{Y}$ be a full-rank linear mapping, let $\mathcal{K} \subset \mathbb{Z}$ be a closed convex cone, and let $M\mathcal{K}$ be a generalized simplicial cone. Then,*

$$(M\mathcal{K})^* = M(M^*M)^{-1}\mathcal{K}^* + \mathrm{Ker}(M^*), \tag{2.3}$$

where $\mathrm{Ker}(M^*)$ is the kernel of the adjoint M^*. In the case $n = m$, this reduces to $(M\mathcal{K})^* = (M^*)^{-1}\mathcal{K}^*$.

Proof. We first show the inclusion $M(M^*M)^{-1}\mathcal{K}^* + \mathrm{Ker}(M^*) \subset (M\mathcal{K})^*$. Let $z = M(M^*M)^{-1}w + v$ with $w \in \mathcal{K}^*$ and $v \in \mathrm{Ker}(M^*)$, and let $y = Mx \in M\mathcal{K}$ with $x \in \mathcal{K}$. Then,

$$\begin{aligned} \langle z, y\rangle &= \left\langle M(M^*M)^{-1}w + v,\, Mx\right\rangle \\ &= \left\langle M(M^*M)^{-1}w,\, Mx\right\rangle + \langle v,\, Mx\rangle \\ &= \left\langle M^*M(M^*M)^{-1}w,\, x\right\rangle + \langle M^*v,\, x\rangle \\ &= \langle w,\, x\rangle \geq 0, \end{aligned} \tag{2.4}$$

where we used $M^*v = 0$ and $w \in \mathcal{K}^*$. Since this holds for all $y \in M\mathcal{K}$, we conclude $z \in (M\mathcal{K})^*$.

For the reverse inclusion, let $z \in (M\mathcal{K})^*$. Then,

$$\langle z,\, Mx\rangle \geq 0 \quad \forall\, x \in \mathcal{K} \quad \Longleftrightarrow \quad \langle M^*z,\, x\rangle \geq 0 \quad \forall\, x \in \mathcal{K} \quad \Longleftrightarrow \quad M^*z \in \mathcal{K}^*. \tag{2.5}$$

Since M has full rank, every $z \in \mathbb{Y}$ can be decomposed as $z = M(M^*M)^{-1}y + v$, where $y \in \mathbb{Z}$ and $v \in \mathrm{Ker}(M^*)$. Applying M^* yields $M^*z = y$, so that

$$M^*z \in \mathcal{K}^* \quad \Longleftrightarrow \quad y \in \mathcal{K}^*.$$

Therefore, $z \in M(M^*M)^{-1}\mathcal{K}^* + \mathrm{Ker}(M^*)$. □

Even when the dual cone is not known explicitly or is difficult to characterize, for instance when M is not square, Moreau's decomposition provides a practical alternative. For any $x \in \mathbb{Y}$,

$$\begin{aligned} -x &= \Pi_{M\mathcal{K}}(-x) + \Pi_{(M\mathcal{K})^\circ}(-x) \\ &= \Pi_{M\mathcal{K}}(-x) - \Pi_{(M\mathcal{K})^*}(x), \end{aligned} \tag{2.6}$$

and therefore

$$\Pi_{(M\mathcal{K})^*}(x) = \Pi_{M\mathcal{K}}(-x) + x. \tag{2.7}$$

Thus, projecting onto the dual cone $(M\mathcal{K})^*$ reduces to projecting onto $M\mathcal{K}$, which is itself a particular instance of the nonlinear conic programming problem (1.1). The semi-smooth Newton method developed in Section 4 handles this as a special case, yielding a unified framework for nonlinear problems constrained by generalized simplicial cones.

We conclude this section with two results on the semi-smoothness of the projection operator, a property on which the local quadratic convergence of the method developed in subsequent sections relies.

Theorem 2.10 (Strong semi-smoothness of the projection onto symmetric cones; [39, Prop. 3.3])**.** *The metric projection $\Pi_{\mathcal{K}}$ onto a symmetric cone $\mathcal{K}$ is strongly semi-smooth.*

The previous result covers important classes such as the nonnegative orthant $\mathbb{R}^n_+$, the second-order cone $\mathbb{L}^n$, and the positive semidefinite cone $\mathbb{S}^n_+$. The following theorem shows that strong semi-smoothness is preserved under linear transformations, extending to generalized simplicial cones $M\mathcal{K}$ even when M is rank-deficient.

Theorem 2.11 (Strong semi-smoothness of the projection onto generalized simplicial cones)**.** *Let $M : \mathbb{Z} \to \mathbb{Y}$ be a linear mapping, let $\mathcal{K} \subset \mathbb{Z}$ be a symmetric cone, and let $\hat{\mathcal{K}} := M\mathcal{K}$. Then the metric projection $\Pi_{\hat{\mathcal{K}}}$ is strongly semi-smooth.*

Proof. Let $x, h \in \mathbb{Y}$. Let w denote the solution of (2.2) corresponding to x, and let z denote the solution corresponding to $x + h$. Setting $k := z - w$, equations (2.2) for $x + h$ and x read

$$(M^*M - \mathrm{Id})\,\Pi_{\mathcal{K}}(w + k) + (w + k) = M^*(x + h), \tag{2.8}$$

$$(M^*M - \mathrm{Id})\,\Pi_{\mathcal{K}}(w) + w = M^*x. \tag{2.9}$$

Subtracting (2.9) from (2.8) and applying the mean value theorem (Theorem 2.3), there exists $V_{\mathcal{K}}(u) \in \partial_C \Pi_{\mathcal{K}}(u)$ with $u \in [w, w + k]$ such that

$$Tk = M^*h, \quad \text{where} \quad T := (M^*M - \mathrm{Id})\,V_{\mathcal{K}}(u) + \mathrm{Id}\,. \tag{2.10}$$

We now distinguish three cases.

Case 1: T is invertible. In this case, $k = T^{-1}M^*h$, and hence

$$\begin{aligned}\Pi_{\hat{\mathcal{K}}}(x+h) - \Pi_{\hat{\mathcal{K}}}(x) - V_{\hat{\mathcal{K}}}(x+h)\,h &= M[\Pi_{\mathcal{K}}(w+k) - \Pi_{\mathcal{K}}(w) - V_{\mathcal{K}}(w+k)\,k] \\ &= M \cdot O\big(\|k\|^2\big) \\ &= M \cdot O\big(\|T^{-1}M^*\|^2\,\|h\|^2\big) \\ &= O\big(\|h\|^2\big)\,,\end{aligned}$$

where the second equality uses the strong semi-smoothness of $\Pi_{\mathcal{K}}$ (Theorem 2.10). This case applies, in particular, when M has full rank; see [4, Lem. 4.1].

Case 2: T is singular and $h \notin \mathrm{Ker}(M^*)$. Setting $\tilde{k} := M^*h \neq 0$, we estimate

$$\begin{aligned}\big\|\Pi_{\hat{\mathcal{K}}}(x+h) - \Pi_{\hat{\mathcal{K}}}(x) - V_{\hat{\mathcal{K}}}(x+h)\,h\big\| &\le \|M\|\,\Big\|\Pi_{\mathcal{K}}(w+\tilde{k}) - \Pi_{\mathcal{K}}(w) - V_{\mathcal{K}}(w+\tilde{k})\,\tilde{k}\Big\| \\ &= \|M\|\,O\Big(\|\tilde{k}\|^2\Big) \\ &= \|M\|\,O\big(\|M^*\|^2\,\|h\|^2\big) \\ &= O\big(\|h\|^2\big)\,.\end{aligned}$$

Case 3: $h \in \mathrm{Ker}(M^*)$. In this case, $M^*(x+h) = M^*x$, so equations (2.8) and (2.9) have the same right-hand side. Consequently, $\Pi_{\hat{\mathcal{K}}}(x+h) = M\Pi_{\mathcal{K}}(w+k) = M\Pi_{\mathcal{K}}(w) = \Pi_{\hat{\mathcal{K}}}(x)$, and the error reduces to

$$\big\|\Pi_{\hat{\mathcal{K}}}(x+h) - \Pi_{\hat{\mathcal{K}}}(x) - V_{\hat{\mathcal{K}}}(x+h)\,h\big\| = \big\|V_{\hat{\mathcal{K}}}(x+h)\,h\big\| = \|MV_{\mathcal{K}}(w+k)\,k\|\,. \tag{2.11}$$

Since $\|V_{\mathcal{K}}(w+k)\| \le 1$ (Theorem 2.41), we have

$$\|MV_{\mathcal{K}}(w+k)\,k\| \le \|M\|\,\|k\|.$$

It remains to show that $\|k\| = O(\|h\|^2)$. By the boundedness of $V_{\mathcal{K}}$ and M, we can choose sequences $h_i \in \mathrm{Ker}(M^*)$ and corresponding k_i such that $\|h_i\|^2 = \alpha\,\|k_i\|$ for some $\alpha > 0$. Then,

$$\frac{\|MV_{\mathcal{K}}(w+k_i)\,k_i\|}{\|h_i\|^2} = \frac{\|MV_{\mathcal{K}}(w+k_i)\,k_i\|}{\alpha\,\|k_i\|} \le \frac{\|M\|\,\|V_{\mathcal{K}}(w+k_i)\|\,\|k_i\|}{\alpha\,\|k_i\|} \le \frac{\|M\|}{\alpha} =: \tilde{C}.$$

Hence, $\big\|V_{\hat{\mathcal{K}}}(x+h)\,h\big\| = O(\|h\|^2)$.

In all three cases, the $O(\|h\|^2)$ estimate holds, so $\Pi_{\hat{\mathcal{K}}}$ is strongly semi-smooth. □

3 Nonlinear Conic Programming Problem

In this section we establish the connection between the conic projection equations and the nonlinear conic programming problem (1.1). Let $\mathbb{X}$ and $\mathbb{Y}$ be finite-dimensional inner product spaces and let $\mathcal{K} \subset \mathbb{Y}$ be a closed convex cone. The Lagrangian associated with problem (1.1) is

$$L(x,\lambda) := f(x) - \langle \lambda,\, g(x)\rangle, \tag{3.1}$$

where $\lambda \in \mathbb{Y}$. The corresponding KKT conditions are

$$\nabla_x L(x, \lambda) = \nabla f(x) - (Dg(x))^* \lambda = 0, \tag{3.2}$$
$$\langle \lambda, g(x) \rangle = 0, \tag{3.3}$$
$$g(x) \in \mathcal{K}, \tag{3.4}$$
$$\lambda \in \mathcal{K}^*. \tag{3.5}$$

These conditions can be written compactly. Define $\mathbf{K} := \mathbb{X} \times \mathcal{K}^*$, so that $\mathbf{K}^* = \{0\} \times \mathcal{K}$. Then (3.2)–(3.5) are equivalent to

$$\left\langle \begin{pmatrix} \nabla f(x) - (Dg(x))^* \lambda \\ g(x) \end{pmatrix}, \begin{pmatrix} x \\ \lambda \end{pmatrix} \right\rangle = 0, \qquad \begin{pmatrix} \nabla f(x) - (Dg(x))^* \lambda \\ g(x) \end{pmatrix} \in \mathbf{K}^*, \qquad (\bar{x}, \bar{\lambda}) \in \mathbf{K}. \tag{3.6}$$

The corresponding conic projection equations are

$$\nabla f(x) - (Dg(x))^* \Pi_{\mathcal{K}^*}(\lambda) = 0, \tag{3.7}$$
$$g(x) - \Pi_{\mathcal{K}^*}(\lambda) + \lambda = 0. \tag{3.8}$$

The system (3.7)–(3.8) has the same dimension as the number of constraints in the original problem. The projection acts on the Lagrange multiplier λ onto the dual cone $\mathcal{K}^*$, encoding the conic constraint; it can be handled efficiently via Moreau's decomposition (2.7).

The following theorem is the main result of this section. It establishes the equivalence between the conic projection equations (3.7)–(3.8) and the KKT conditions (3.6), thereby enabling the use of equation-solving methods to locate first-order optimality points of problem (1.1).

Theorem 3.1 (Equivalence between projection equations and KKT conditions)**.** *If (x, λ) solves* (3.7)*–*(3.8)*, then $(x, \Pi_{\mathcal{K}^*}(\lambda))$ satisfies the KKT conditions* (3.6)*. Conversely, if (x, σ) satisfies* (3.6)*, then (x, λ) solves* (3.7)*–*(3.8)*, where $\lambda := \sigma - g(x)$.*

Proof. **(i)** Suppose (x, λ) solves (3.7)–(3.8). From (3.7), the stationarity condition (3.2) holds with multiplier $\Pi_{\mathcal{K}^*}(\lambda)$. By definition, $\Pi_{\mathcal{K}^*}(\lambda) \in \mathcal{K}^*$, which gives (3.5). From (3.8) and Moreau's decomposition,

$$g(x) = \Pi_{\mathcal{K}^*}(\lambda) - \lambda = -\Pi_{\mathcal{K}^\circ}(-\lambda) - \lambda = \Pi_{\mathcal{K}}(-\lambda) \in \mathcal{K},$$

which yields (3.4). It remains to verify the complementarity condition (3.3). Using the identity $g(x) = \Pi_{\mathcal{K}}(-\lambda)$ and $\Pi_{\mathcal{K}^*}(\lambda) = -\Pi_{\mathcal{K}^\circ}(-\lambda)$, together with the orthogonality property of Moreau's decomposition, we obtain

$$\langle g(x), \Pi_{\mathcal{K}^*}(\lambda) \rangle = \langle \Pi_{\mathcal{K}}(-\lambda), -\Pi_{\mathcal{K}^\circ}(-\lambda) \rangle = -\langle \Pi_{\mathcal{K}}(-\lambda), \Pi_{\mathcal{K}^\circ}(-\lambda) \rangle = 0.$$

(ii) Suppose (x, σ) satisfies (3.6), so that $\sigma \in \mathcal{K}^*$, $g(x) \in \mathcal{K}$, $\langle g(x), \sigma \rangle = 0$, and $\nabla f(x) - (Dg(x))^* \sigma = 0$. Set $\lambda := \sigma - g(x)$. Since $\sigma \in \mathcal{K}^*$, $g(x) \in \mathcal{K}$, and $\langle \sigma, g(x) \rangle = 0$, the uniqueness of Moreau's decomposition gives

$$\Pi_{\mathcal{K}^*}(\lambda) = \sigma \quad \text{and} \quad \Pi_{\mathcal{K}^\circ}(\lambda) = -g(x).$$

Substituting into (3.7)–(3.8):

$$\nabla f(x) - (Dg(x))^* \Pi_{\mathcal{K}^*}(\lambda) = \nabla f(x) - (Dg(x))^* \sigma = 0,$$

and

$$g(x) - \Pi_{\mathcal{K}^*}(\lambda) + \lambda = g(x) - \sigma + (\sigma - g(x)) = 0.$$

Hence, (x, λ) solves (3.7)–(3.8). □

Establishing general sufficient conditions for the existence and uniqueness of solutions to (3.7)–(3.8) may be overly restrictive. Since we assume that problem (1.1) admits at least one solution, we do not pursue such conditions here. In the particular case where $g = \mathrm{Id}$, however, a sufficient condition can be obtained. The system then reduces to

$$\nabla f(x) - \Pi_{\mathcal{K}^*}(\lambda) = 0, \tag{3.9}$$

$$x - \Pi_{\mathcal{K}^*}(\lambda) + \lambda = 0. \tag{3.10}$$

By Moreau's decomposition, (3.10) gives $x = \Pi_{\mathcal{K}}(-\lambda)$. Substituting into (3.9) and setting $y := -\lambda$, the system reduces to the single equation

$$\nabla f(\Pi_{\mathcal{K}}(y)) - \Pi_{\mathcal{K}}(y) + y = 0. \tag{3.11}$$

Proposition 3.2 (Existence and uniqueness for the identity-constrained case)**.** *Let $f : \mathbb{X} \to \mathbb{R}$ be twice continuously differentiable with Lipschitz continuous gradient. If $g = \mathrm{Id}$ in problem* (1.1) *and $\|\mathrm{Id} - \nabla^2 f(z)\| < 1$ for all $z \in \mathbb{X}$, then equation* (3.11) *has a unique solution.*

Proof. The equivalence between (3.7)–(3.8) and (3.11) when $g = \mathrm{Id}$ was established above. It remains to show uniqueness under the hypothesis $\|\mathrm{Id} - \nabla^2 f(z)\| < 1$ for all $z \in \mathbb{X}$.

Define $\phi(y) := y - \nabla f(y)$ and $\Phi := \phi \circ \Pi_{\mathcal{K}}$. Observe that equation (3.11) is equivalent to the fixed-point problem $\Phi(y) = y$. We show that Φ is a contraction. By the mean value theorem, for any $x, y \in \mathbb{X}$ there exists $z \in [x, y]$ such that

$$\phi(y) - \phi(x) = (y - x) - \big(\nabla f(y) - \nabla f(x)\big) = \big(\mathrm{Id} - \nabla^2 f(z)\big)(y - x),$$

and therefore $\|\phi(y) - \phi(x)\| \le \|\mathrm{Id} - \nabla^2 f(z)\| \, \|y - x\| < \|y - x\|$. Since the projection $\Pi_{\mathcal{K}}$ is non-expansive, the composition $\Phi = \phi \circ \Pi_{\mathcal{K}}$ is also a contraction. By the contraction mapping principle (Theorem 2.2), Φ has a unique fixed point, which is the unique solution of (3.11). □

Remark 3.3. *In the particular case $f(x) = \frac{1}{2} x^T Q x + q^T x$, taking $\mathcal{K} = \mathbb{R}^n_+$ and $\mathcal{K} = \mathbb{L}^n$ recovers [7, Prop. 1] and [8, Prop. 5], respectively.*

A special case of problem (1.1) arises when an additional conic constraint is imposed on x. For a closed convex cone $\mathcal{C} \subset \mathbb{X}$, the problem becomes

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g(x) \in \mathcal{K} \\ & x \in \mathcal{C}. \end{array} \tag{3.12}$$

Setting $\hat{g}(x) := (x,\, g(x))$ and $\hat{\mathcal{C}} := \mathcal{C} \times \mathcal{K}$, problem (3.12) reduces to (1.1) with constraint $\hat{g}(x) \in \hat{\mathcal{C}}$. The dual cone decomposes as $\hat{\mathcal{C}}^* = \mathcal{C}^* \times \mathcal{K}^*$, so the multiplier takes the form $\lambda = (\mu, \sigma) \in \mathcal{C}^* \times \mathcal{K}^*$. Substituting into the conic projection equations, (3.7) becomes

$$\begin{aligned} \nabla f(x) - (D\hat{g}(x))^* \Pi_{\hat{\mathcal{C}}^*}(\lambda) &= \nabla f(x) - (\mathrm{Id},\, Dg(x))^* \big(\Pi_{\mathcal{C}^*}(\mu),\, \Pi_{\mathcal{K}^*}(\sigma)\big) \\ &= \nabla f(x) - (Dg(x))^* \Pi_{\mathcal{K}^*}(\sigma) - \Pi_{\mathcal{C}^*}(\mu) = 0, \end{aligned}$$

and (3.8) becomes

$$\begin{aligned} \hat{g}(x) - \Pi_{\hat{\mathcal{C}}^*}(\lambda) + \lambda &= \big(x,\, g(x)\big) - \big(\Pi_{\mathcal{C}^*}(\mu),\, \Pi_{\mathcal{K}^*}(\sigma)\big) + (\mu,\, \sigma) \\ &= \big(x - \Pi_{\mathcal{C}^*}(\mu) + \mu,\; g(x) - \Pi_{\mathcal{K}^*}(\sigma) + \sigma\big) = (0,\, 0). \end{aligned}$$

By Moreau's decomposition, the first component gives $x = \Pi_{\mathcal{C}}(-\mu)$. Setting $y := -\mu$, the system reduces to

$$\nabla f(\Pi_{\mathcal{C}}(y)) - (Dg(\Pi_{\mathcal{C}}(y)))^* \Pi_{\mathcal{K}^*}(\sigma) - \Pi_{\mathcal{C}}(y) + y = 0, \tag{3.13}$$

$$g(\Pi_{\mathcal{C}}(y)) - \Pi_{\mathcal{K}^*}(\sigma) + \sigma = 0. \tag{3.14}$$

By Theorem 3.1, the KKT points of (3.12) are recovered as $(\bar{x}, \bar{\lambda}) = \big(\Pi_{\mathcal{C}}(\bar{y}), \Pi_{\mathcal{K}^*}(\bar{\sigma})\big)$, where $(\bar{y}, \bar{\sigma})$ solves (3.13)–(3.14).

4 Semi-smooth Newton Method for Nonlinear Conic Programming

Having characterized the first-order KKT points of (1.1) via the conic projection equations (3.7)–(3.8), we now turn to solving this system. We define the residual map

$$H(x, \lambda) := \begin{pmatrix} \nabla f(x) - (Dg(x))^* \Pi_{\mathcal{K}^*}(\lambda) \\ g(x) - \Pi_{\mathcal{K}^*}(\lambda) + \lambda \end{pmatrix}, \tag{4.1}$$

and apply a semi-smooth Newton method to find its zeros. Since f and g are twice continuously differentiable and $\Pi_{\mathcal{K}^*}$ is strongly semi-smooth, the Clarke generalized Jacobian of H takes the form

$$J_H(x, \lambda) = \begin{pmatrix} \nabla^2 f(x) - (D^2 g(x))^* \Pi_{\mathcal{K}^*}(\lambda) & -(Dg(x))^* V_{\mathcal{K}^*}(\lambda) \\ Dg(x) & \mathrm{Id} - V_{\mathcal{K}^*}(\lambda) \end{pmatrix}, \tag{4.2}$$

where $V_{\mathcal{K}^*}(\lambda) \in \partial_C \Pi_{\mathcal{K}^*}(\lambda)$. The standard semi-smooth Newton iteration reads as

$$J_H(x^k, \lambda^k) \begin{pmatrix} x^{k+1} - x^k \\ \lambda^{k+1} - \lambda^k \end{pmatrix} = -H(x^k, \lambda^k). \tag{4.3}$$

Remark 4.1. *When g is linear and $\mathcal{K} = \{0\}$, iteration (4.3) reduces to the method studied in [4]. If the constraint $g(x) \in \mathcal{K}$ is absent entirely, we recover the first method in [3].*

It is well known that the Newton direction is highly effective near a solution but may encounter difficulties far from one. To improve robustness, we decouple the search direction from the stepsize by writing the iteration as

$$J_H(x^k, \lambda^k) \begin{pmatrix} d_x^k \\ d_\lambda^k \end{pmatrix} = -H(x^k, \lambda^k), \tag{4.4}$$

and updating

$$(x^{k+1}, \lambda^{k+1}) = (x^k, \lambda^k) + \alpha_k (d_x^k, d_\lambda^k), \tag{4.5}$$

where $\alpha_k > 0$ is chosen to ensure sufficient decrease in the merit function

$$\theta(x, \lambda) := \tfrac{1}{2} \|H(x, \lambda)\|^2. \tag{4.6}$$

4.1 Convergence of the semi-smooth Newton method

We begin with a key lemma establishing the strong semi-smoothness of the merit function θ, which is central to the convergence analysis.

Lemma 4.2 (Strong semi-smoothness of the merit function)**.** *Let f and g be twice continuously differentiable and let $\mathcal{K}$ be a symmetric cone. Then $\theta(x,\lambda) = \frac{1}{2}\|H(x,\lambda)\|^2$ is strongly semi-smooth, that is,*

$$\theta(x+w_x,\,\lambda+w_\lambda) = \theta(x,\lambda) + \nabla\theta(x+w_x,\,\lambda+w_\lambda)^T(w_x,\,w_\lambda) + O\big(\|(w_x,\,w_\lambda)\|^2\big)\,,$$

where $\nabla\theta(x,\lambda) \in \partial_C\theta(x,\lambda)$. Moreover,

$$\nabla\theta(x,\lambda)^T(w_x,\,w_\lambda) = H(x,\lambda)^T J_H(x,\lambda)\,(w_x,\,w_\lambda). \tag{4.7}$$

Proof. Since f and g are smooth, the only nonsmooth component of H is $\Pi_{\mathcal{K}^*}(\lambda)$. By Moreau's decomposition,

$$\Pi_{\mathcal{K}^*}(\lambda) = \lambda + \Pi_{\mathcal{K}}(-\lambda),$$

which expresses $\Pi_{\mathcal{K}^*}$ as an affine function of $\Pi_{\mathcal{K}}$. Since affine maps preserve strong semi-smoothness and Theorem 2.11 guarantees that $\Pi_{\mathcal{K}}$ is strongly semi-smooth, it follows that $\Pi_{\mathcal{K}^*}$ is strongly semi-smooth. Therefore H, being a composition of smooth functions with $\Pi_{\mathcal{K}^*}$, is strongly semi-smooth. Since $\phi(t) := \frac{1}{2}\|t\|^2$ is smooth and its composition with a strongly semi-smooth function is again strongly semi-smooth, we conclude that $\theta = \phi \circ H$ is strongly semi-smooth.

It remains to establish (4.7). At any differentiability point $(x,\lambda) \in D_\theta$, the chain rule gives

$$\nabla\theta(x,\lambda)^T(w_x,\,w_\lambda) = H(x,\lambda)^T J_H(x,\lambda)\,(w_x,\,w_\lambda),$$

where $J_H(x,\lambda)$ is given by (4.2) with $\Pi'_{\mathcal{K}^*}(\lambda)$ in place of $V_{\mathcal{K}^*}(\lambda)$. Since D_θ is dense and both sides are continuous in the Clarke sense, the identity extends to all (x,λ) by replacing $\Pi'_{\mathcal{K}^*}(\lambda)$ with any $V_{\mathcal{K}^*}(\lambda) \in \partial_C\Pi_{\mathcal{K}^*}(\lambda)$, yielding (4.7). ☐

A useful consequence of Lemma 4.2 is that the Newton direction is a descent direction for θ. When $J_H(x^k,\lambda^k)$ is nonsingular and (x^k,λ^k) is not a solution of (3.7)–(3.8), it follows from (4.4) and (4.7) that

$$\begin{aligned}
\nabla\theta(x^k,\lambda^k)^T(d_x^k,\,d_\lambda^k) &= H(x^k,\lambda^k)^T J_H(x^k,\lambda^k)\,(d_x^k,\,d_\lambda^k)\\
&= -H(x^k,\lambda^k)^T H(x^k,\lambda^k)\\
&= -\|H(x^k,\lambda^k)\|^2\\
&< 0.
\end{aligned}$$

When $J_H(x^k,\lambda^k)$ is singular, the Newton system (4.4) cannot be solved directly. In this case, we replace it with the regularized normal equation

$$\left({J_H^k}^T J_H^k + \sqrt{\theta(x^k,\lambda^k)}\ \mathrm{Id}\right)(d_x^k,\,d_\lambda^k) = -{J_H^k}^T H(x^k,\lambda^k), \tag{4.8}$$

where we write $J_H^k := J_H(x^k,\lambda^k)$ for brevity. The coefficient matrix in (4.8) is symmetric positive definite, so the system always has a unique solution. Moreover, the resulting direction is still a descent direction for θ:

$$\begin{aligned}
\nabla\theta(x^k,\lambda^k)^T(d_x^k,\,d_\lambda^k) &= H(x^k,\lambda^k)^T J_H^k\,(d_x^k,\,d_\lambda^k)\\
&= -(d_x^k,\,d_\lambda^k)^T\left({J_H^k}^T J_H^k + \sqrt{\theta(x^k,\lambda^k)}\ \mathrm{Id}\right)(d_x^k,\,d_\lambda^k)\\
&< 0,
\end{aligned}$$

where the last inequality holds because the matrix is positive definite and $(d_x^k, d_\lambda^k) \neq 0$.

The following lemma provides a sufficient condition for the stepsize $\alpha_k = 1$ to satisfy the Armijo condition, ensuring that full Newton steps are eventually accepted.

Lemma 4.3 (Sufficient condition for a unit stepsize; [35, Lem. 5.2])**.** *Suppose there exists a constant* $\rho > 0$ *such that*

$$\nabla\theta(x^k, \lambda^k)^T (d_x^k,\, d_\lambda^k) \leq -\rho\, \|(d_x^k,\, d_\lambda^k)\|^2$$

for all k*. Then, for every* $c \in (0, \frac{1}{2})$*, there exists* $\hat{k} \geq 0$ *such that*

$$\theta(x^k + d_x^k,\, \lambda^k + d_\lambda^k) \leq \theta(x^k, \lambda^k) + c\, \nabla\theta(x^k, \lambda^k)^T (d_x^k,\, d_\lambda^k)$$

for all $k \geq \hat{k}$*, i.e., the Armijo linesearch accepts the unit stepsize* $\alpha_k = 1$*.*

We now verify that both search directions satisfy the hypothesis of Lemma 4.3.

Proposition 4.4 (Sufficient descent property)**.** *The Newton direction given by* (4.4) *and the regularized direction given by* (4.8) *both satisfy*

$$\nabla\theta(x^k, \lambda^k)^T (d_x^k,\, d_\lambda^k) \leq -\rho_k\, \|(d_x^k,\, d_\lambda^k)\|^2$$

for some constant $\rho_k > 0$ *at each iteration. In particular, if* $\rho := \inf_k \rho_k > 0$*, the hypothesis of Lemma 4.3 is satisfied.*

Proof. We write $J_H^k := J_H(x^k, \lambda^k)$ and $H^k := H(x^k, \lambda^k)$ for brevity.

(i) Nonsingular case. Since $(d_x^k, d_\lambda^k) = -(J_H^k)^{-1} H^k$, we have

$$\begin{aligned}
\|(d_x^k,\, d_\lambda^k)\|^2 &= (H^k)^T ({J_H^k}^T)^{-1} (J_H^k)^{-1} H^k \\
&= (H^k)^T ({J_H^k}^T J_H^k)^{-1} H^k \\
&\leq \lambda_{\max}\Big(({J_H^k}^T J_H^k)^{-1} \Big)\, \|H^k\|^2 \\
&= \frac{1}{\rho_k}\, \|H^k\|^2,
\end{aligned}$$

where $\rho_k := \lambda_{\min}({J_H^k}^T J_H^k) > 0$, and the last equality uses Lemma 2.1. Since $\nabla\theta(x^k, \lambda^k)^T (d_x^k, d_\lambda^k) = -\|H^k\|^2$ by (4.7) and (4.4), we conclude

$$\nabla\theta(x^k, \lambda^k)^T (d_x^k,\, d_\lambda^k) = -\|H^k\|^2 \leq -\rho_k\, \|(d_x^k,\, d_\lambda^k)\|^2.$$

(ii) Singular case. Define $A^k := {J_H^k}^T J_H^k + \sqrt{\theta(x^k, \lambda^k)}\, \mathrm{Id}$, which is symmetric positive definite. From (4.8), $(d_x^k, d_\lambda^k) = -(A^k)^{-1} {J_H^k}^T H^k$, and by (4.7),

$$\nabla\theta(x^k, \lambda^k)^T (d_x^k,\, d_\lambda^k) = (H^k)^T J_H^k\, (d_x^k,\, d_\lambda^k) = -({J_H^k}^T H^k)^T (A^k)^{-1} ({J_H^k}^T H^k).$$

Similarly, $\|(d_x^k,\, d_\lambda^k)\|^2 = ({J_H^k}^T H^k)^T (A^k)^{-2} ({J_H^k}^T H^k)$. Setting $u := {J_H^k}^T H^k$ and using that $\lambda_{\max}\big((A^k)^{-1}\big)\, (A^k)^{-1} - (A^k)^{-2}$ is positive semidefinite (Lemma 2.1), we obtain

$$\|(d_x^k,\, d_\lambda^k)\|^2 = u^T (A^k)^{-2} u \leq \frac{1}{\rho_k} \Big(-\nabla\theta(x^k, \lambda^k)^T (d_x^k,\, d_\lambda^k) \Big),$$

where $\rho_k := \lambda_{\min}(A^k) \geq \sqrt{\theta(x^k, \lambda^k)} > 0$ whenever (x^k, λ^k) is not a solution. Rearranging gives $\nabla\theta(x^k, \lambda^k)^T (d_x^k,\, d_\lambda^k) \leq -\rho_k\, \|(d_x^k,\, d_\lambda^k)\|^2$. □

Remark 4.5. *The condition $\rho := \inf_k \rho_k > 0$ in Proposition 4.4 is satisfied, in particular, when the Jacobian $J_H(\bar{x}, \bar{\lambda})$ is nonsingular at the solution, which is a standard assumption for Newton-type methods to achieve local quadratic convergence. When the Jacobian is singular at the solution, ρ_k may tend to zero, and the unit stepsize is no longer guaranteed to be eventually accepted; nevertheless, the linesearch still produces a sufficient decrease at each iteration, so convergence of the iterates is not lost.*

With a well-defined iteration producing descent directions for the merit function θ, it remains to select an appropriate stepsize α_k. Regardless of whether $J_H(x^k, \lambda^k)$ is singular or not, we employ a classical Armijo backtracking linesearch. For a fixed parameter $c \in (0, \frac{1}{2})$, the Armijo condition reads

$$\theta(x^k + \alpha_k\, d_x^k,\, \lambda^k + \alpha_k\, d_\lambda^k) \leq \theta(x^k, \lambda^k) + \alpha_k\, c\, \nabla\theta(x^k, \lambda^k)^T (d_x^k,\, d_\lambda^k). \tag{4.9}$$

When $J_H(x^k, \lambda^k)$ is nonsingular, recalling that $\nabla\theta(x^k, \lambda^k)^T (d_x^k, d_\lambda^k) = -\|H(x^k, \lambda^k)\|^2$, condition (4.9) reduces to

$$\tfrac{1}{2} \left\| H(x^k + \alpha_k\, d_x^k,\, \lambda^k + \alpha_k\, d_\lambda^k) \right\|^2 \leq \left(\tfrac{1}{2} - \alpha_k\, c\right) \|H(x^k, \lambda^k)\|^2. \tag{4.10}$$

When $J_H(x^k, \lambda^k)$ is singular, the right-hand side of (4.9) takes the form

$$\tfrac{1}{2}\|H(x^k, \lambda^k)\|^2 - \alpha_k\, c\, (d_x^k,\, d_\lambda^k)^T \left(J_H(x^k, \lambda^k)^T J_H(x^k, \lambda^k) + \sqrt{\theta(x^k, \lambda^k)}\ \mathrm{Id} \right) (d_x^k,\, d_\lambda^k). \tag{4.11}$$

We conclude this subsection with an estimate relating the merit function to the distance to a solution $(\bar{x}, \bar{\lambda})$. Suppose that $\|J_H(x^k, \lambda^k)\| \leq C_1$ for some constant $C_1 > 0$. Since $H(\bar{x}, \bar{\lambda}) = 0$ and $\theta(\bar{x}, \bar{\lambda}) = 0$, applying Lemma 4.2 yields

$$\begin{aligned}
\theta(x^k, \lambda^k) &= \theta(\bar{x}, \bar{\lambda}) + \nabla\theta(x^k, \lambda^k)^T \left((x^k, \lambda^k) - (\bar{x}, \bar{\lambda})\right) + O\Big(\|(x^k, \lambda^k) - (\bar{x}, \bar{\lambda})\|^2\Big) \\
&= H(x^k, \lambda^k)^T J_H(x^k, \lambda^k) \left((x^k, \lambda^k) - (\bar{x}, \bar{\lambda})\right) + O\Big(\|(x^k, \lambda^k) - (\bar{x}, \bar{\lambda})\|^2\Big) \\
&\leq \|J_H(x^k, \lambda^k)\|\, \|H(x^k, \lambda^k)\|\, \|(x^k, \lambda^k) - (\bar{x}, \bar{\lambda})\| + O\Big(\|(x^k, \lambda^k) - (\bar{x}, \bar{\lambda})\|^2\Big) \\
&\leq C_1\, \|H(x^k, \lambda^k)\|\, \|(x^k, \lambda^k) - (\bar{x}, \bar{\lambda})\| + O\Big(\|(x^k, \lambda^k) - (\bar{x}, \bar{\lambda})\|^2\Big).
\end{aligned}$$

Since $\|H(x^k, \lambda^k)\| = \sqrt{2\,\theta(x^k, \lambda^k)}$, the left-hand side appears on both sides of the inequality, giving

$$\theta(x^k, \lambda^k) = O\Big(\|(x^k, \lambda^k) - (\bar{x}, \bar{\lambda})\|^2\Big), \tag{4.12}$$

and consequently

$$\sqrt{\theta(x^k, \lambda^k)} = O\Big(\|(x^k, \lambda^k) - (\bar{x}, \bar{\lambda})\|\Big). \tag{4.13}$$

Theorem 4.6 (Local quadratic convergence of the semi-smooth Newton method)**.** *Suppose $(\bar{x}, \bar{\lambda})$ is a solution of (3.7)–(3.8), that is, $H(\bar{x}, \bar{\lambda}) = 0$. Assume that there exist constants $C_1, C_2, C_3 > 0$ such that, for all k:*

1. *$\|J_H(x^k, \lambda^k)\| \leq C_1$,*
2. *$\|J_H(x^k, \lambda^k)^{-1}\| \leq C_2$ whenever $J_H(x^k, \lambda^k)$ is nonsingular,*

3. $\left\| \left(J_H(x^k,\lambda^k)^T J_H(x^k,\lambda^k) + \sqrt{\theta(x^k,\lambda^k)}\ \mathrm{Id} \right)^{-1} \right\| \leq C_3$ *whenever* $J_H(x^k,\lambda^k)$ *is singular.*

If the semi-smooth Newton method generates a sequence (x^k,λ^k) *converging to* $(\bar{x},\bar{\lambda})$*, then the convergence is quadratic.*

Proof. We write $J_H^k := J_H(x^k,\lambda^k)$, $H^k := H(x^k,\lambda^k)$ and $\theta^k := \theta(x^k,\lambda^k)$ for brevity.

By Lemma 4.3 and Proposition 4.4, there exists an index $\hat{k}$ such that the unit stepsize $\alpha_k = 1$ satisfies the Armijo condition (4.9) for all $k \geq \hat{k}$. We consider the two cases separately.

(i) Nonsingular case. When J_H^k is nonsingular, the update with $\alpha_k = 1$ gives $(x^{k+1},\lambda^{k+1}) = (x^k,\lambda^k) + (d_x^k, d_\lambda^k)$. Using (4.4) and $H(\bar{x},\bar{\lambda}) = 0$:

$$\begin{aligned}
\|(x^{k+1},\lambda^{k+1}) - (\bar{x},\bar{\lambda})\| &= \|(x^k,\lambda^k) - (\bar{x},\bar{\lambda}) + (d_x^k, d_\lambda^k)\| \\
&= \|(x^k,\lambda^k) - (\bar{x},\bar{\lambda}) - (J_H^k)^{-1} H^k\| \\
&\leq \|(J_H^k)^{-1}\|\, \|H^k - H(\bar{x},\bar{\lambda}) - J_H^k\big((x^k,\lambda^k) - (\bar{x},\bar{\lambda})\big)\| \\
&\leq C_2\, O\Big(\|(x^k,\lambda^k) - (\bar{x},\bar{\lambda})\|^2\Big),
\end{aligned}$$

where the last inequality follows from the strong semi-smoothness of H (Lemma 4.2).

(ii) Singular case. When J_H^k is singular, the direction is given by (4.8). Denoting

$$A^k := (J_H^k)^T J_H^k + \sqrt{\theta^k}\ \mathrm{Id},$$

the update with $\alpha_k = 1$ yields

$$\begin{aligned}
\|(x^{k+1},\lambda^{k+1}) - (\bar{x},\bar{\lambda})\| &= \|(x^k,\lambda^k) - (\bar{x},\bar{\lambda}) - (A^k)^{-1}(J_H^k)^T H^k\| \\
&= \|(A^k)^{-1}\| \|A^k\big((x^k,\lambda^k) - (\bar{x},\bar{\lambda})\big) - (J_H^k)^T H^k\| \\
&\leq C_3\, \Big(\|(J_H^k)^T\| \|H^k - H(\bar{x},\bar{\lambda}) - J_H^k\big((x^k,\lambda^k) - (\bar{x},\bar{\lambda})\big)\| \\
&\qquad + \sqrt{\theta^k} \|(x^k,\lambda^k) - (\bar{x},\bar{\lambda})\|\Big) \\
&\leq C_3 C_1\, O\Big(\|(x^k,\lambda^k) - (\bar{x},\bar{\lambda})\|^2\Big) + C_3\, \sqrt{\theta^k} \|(x^k,\lambda^k) - (\bar{x},\bar{\lambda})\|,
\end{aligned}$$

where we used the strong semi-smoothness of H and the bound $\|J_H(x^k,\lambda^k)\| \leq C_1$. By estimate (4.13), $\sqrt{\theta^k} = O(\|(x^k,\lambda^k) - (\bar{x},\bar{\lambda})\|)$, so the second term is also $O(\|(x^k,\lambda^k) - (\bar{x},\bar{\lambda})\|^2)$.

In both cases, $\|(x^{k+1},\lambda^{k+1}) - (\bar{x},\bar{\lambda})\| = O(\|(x^k,\lambda^k) - (\bar{x},\bar{\lambda})\|^2)$, establishing quadratic convergence. □

4.2 Choice of the generalized Jacobian $V_{\mathcal{K}}$

The method depends explicitly on the choice of $V_{\mathcal{K}}(x) \in \partial_C \Pi_{\mathcal{K}}(x)$. Recall that $\partial_C \Pi_{\mathcal{K}}(x) = \mathrm{conv}(\partial_B \Pi_{\mathcal{K}}(x))$, which yields different expressions for this linear operator depending on the position of x relative to $\mathcal{K}$, in particular whether $x \in \mathrm{bd}(\mathcal{K})$ or $x \in \mathrm{bd}(\mathcal{K}^\circ)$. For the second-order cone and the positive semidefinite cone, closed-form expressions are available; see [26, Lem. 2.6] and [32, Thm. 3.7], respectively. The choice of $V_{\mathcal{K}}(x)$ can have a noticeable impact on performance: in [4], for instance, certain selections led to faster iterations in higher-dimensional problems. For each experiment in Section 5, we specify the generalized Jacobian used.

4.3 Choice of the starting point

The convergence of the method depends on the initial point (x^0, λ^0). Since any solution of (3.7)–(3.8) satisfies both equations simultaneously, a natural strategy is to choose an initial point that already satisfies one of them: either the feasibility equation $g(x)-\Pi_{\mathcal{K}^*}(\lambda)+\lambda = 0$ or the stationarity condition $\nabla f(x) - (Dg(x))^*\Pi_{\mathcal{K}^*}(\lambda) = 0$. Another common choice is a random starting point. In the numerical experiments of Section 5, we observe that different initial points can affect both computation time and the solution found, and that the origin $(x^0, \lambda^0) = (0, 0)$ tends to perform well in general. A deeper investigation of initialization strategies is left for future work.

4.4 Stationarity of θ and escape strategies

A practical difficulty may arise when the method reaches a point (x^k, λ^k) at which $\nabla\theta(x^k, \lambda^k) = 0$ but $\theta(x^k, \lambda^k) \neq 0$. Since $H(x^k, \lambda^k) \neq 0$, the point is not a solution, yet no descent direction can be extracted from the gradient. This occurs precisely when $J_H(x^k, \lambda^k)$ is singular, as $\nabla\theta(x^k, \lambda^k) = 0$ is equivalent to

$$H(x^k, \lambda^k) \in \operatorname{Ker}\Big(J_H(x^k, \lambda^k)^T\Big) = \operatorname{Im}\Big(J_H(x^k, \lambda^k)\Big)^{\perp}.$$

In the regularized system (4.8), the right-hand side becomes $J_H(x^k, \lambda^k)^T H(x^k, \lambda^k) = \nabla\theta(x^k, \lambda^k) = 0$, so the method produces the null direction and stalls.

To address this, we decompose H into its optimality and feasibility components:

$$H^{\text{opt}}(x, \lambda) := \nabla f(x) - (Dg(x))^*\Pi_{\mathcal{K}^*}(\lambda), \tag{4.14}$$

$$H^{\text{feas}}(x, \lambda) := g(x) - \Pi_{\mathcal{K}^*}(\lambda) + \lambda, \tag{4.15}$$

with corresponding merit functions

$$\theta^{\text{opt}}(x, \lambda) := \tfrac{1}{2}\|H^{\text{opt}}(x, \lambda)\|^2, \qquad \theta^{\text{feas}}(x, \lambda) := \tfrac{1}{2}\|H^{\text{feas}}(x, \lambda)\|^2. \tag{4.16}$$

Since $\theta = \theta^{\text{opt}} + \theta^{\text{feas}}$, the gradient decomposes as

$$\nabla\theta(x, \lambda) = J_{H^{\text{opt}}}(x, \lambda)^T H^{\text{opt}}(x, \lambda) + J_{H^{\text{feas}}}(x, \lambda)^T H^{\text{feas}}(x, \lambda). \tag{4.17}$$

In particular, $\nabla\theta(x, \lambda) = 0$ does not require the individual terms to vanish: it is possible that $\nabla\theta^{\text{opt}}(x, \lambda) \neq 0$ and $\nabla\theta^{\text{feas}}(x, \lambda) \neq 0$, with their sum cancelling. This motivates the following distinction.

Definition 4.7 (Stationarity and strong stationarity)**.** *A point (x, λ) is said to be* stationary *for* (3.7)–(3.8) *if $\nabla\theta(x, \lambda) = 0$, and* strongly stationary *if $\nabla\theta^{\text{opt}}(x, \lambda) = 0$ and $\nabla\theta^{\text{feas}}(x, \lambda) = 0$.*

Strong stationarity implies stationarity, but the converse does not hold in general. There are, however, important cases where the two notions coincide, such as the nearest correlation matrix problem studied in [4].

When the method reaches a stationary point that is not strongly stationary, the decomposition (4.17) provides an escape strategy. Since at least one of the individual gradients is nonzero, it can be used as a right-hand side in place of the vanishing $\nabla\theta$. Specifically, we solve

$$\left(J_H(x^k, \lambda^k)^T J_H(x^k, \lambda^k) + \sqrt{\theta(x^k, \lambda^k)}\ \mathrm{Id}\right)(d_x^k, d_\lambda^k) = -J_{H^{\text{feas}}}(x^k, \lambda^k)^T H^{\text{feas}}(x^k, \lambda^k). \tag{4.18}$$

Proposition 4.8 (Feasibility descent at non-strongly stationary points)**.** *If (x^k, λ^k) is stationary but not strongly stationary and $\nabla\theta^{\text{feas}}(x^k, \lambda^k) \neq 0$, then the direction (d_x^k, d_λ^k) given by (4.18) is a descent direction for θ^{feas}.*

Proof. The coefficient matrix in (4.18) is symmetric positive definite, so the direction is well defined and nonzero. Substituting (4.18) into the directional derivative:

$$\begin{aligned}\nabla\theta^{\text{feas}}(x^k, \lambda^k)^T (d_x^k,\, d_\lambda^k) &= H^{\text{feas}}(x^k, \lambda^k)^T J_{H^{\text{feas}}}(x^k, \lambda^k)\, (d_x^k,\, d_\lambda^k) \\ &= -(d_x^k,\, d_\lambda^k)^T \left(J_H(x^k, \lambda^k)^T J_H(x^k, \lambda^k) + \sqrt{\theta(x^k, \lambda^k)}\; \mathrm{Id} \right) (d_x^k,\, d_\lambda^k) \\ &< 0.\end{aligned}$$

□

Since $\nabla\theta^{\text{opt}}(x^k, \lambda^k)^T (d_x^k, d_\lambda^k) = -\nabla\theta^{\text{feas}}(x^k, \lambda^k)^T (d_x^k, d_\lambda^k) > 0$, this direction is an ascent direction for θ^{opt}. We therefore prioritize reducing the feasibility residual, since θ^{opt} can reach non-optimal strongly stationary points, as observed in the numerical experiments of Section 5.

Remark 4.9. *Handling strongly stationary points that are not solutions remains an open question. While the decomposition above provides an escape mechanism for non-strongly stationary points, developing strategies to detect and escape strongly stationary points could significantly improve the robustness of the method.*

We summarize the complete method in Algorithm 1. When the problem involves a generalized simplicial cone constraint, the projection subproblem (2.1) can itself be solved as a particular instance of Algorithm 1.

Algorithm 1 Semi-smooth Newton method for NCP

Require: $c \in (0, \frac{1}{2})$, $(x^0, \lambda^0) \in \mathbb{X} \times \mathbb{Y}$, $\texttt{tol} > 0$, $\texttt{dtol} > 0$, $\texttt{maxiter} \in \mathbb{N}$, $\texttt{maxiter\_ls} \in \mathbb{N}$
  $k \leftarrow 0$
  Compute $\Pi_{\mathcal{K}^*}(\lambda^0)$, $H(x^0, \lambda^0)$, and $\theta(x^0, \lambda^0) = \frac{1}{2}\|H(x^0, \lambda^0)\|^2$
  while $\|H(x^k, \lambda^k)\| \geq \texttt{tol}$ **and** $k < \texttt{maxiter}$ **do**
    Compute $J_H(x^k, \lambda^k)$
    Solve $J_H(x^k, \lambda^k)\,(d_x^k, d_\lambda^k) = -H(x^k, \lambda^k)$
    if $\nabla\theta(x^k, \lambda^k)^T(d_x^k, d_\lambda^k) \geq 0$ **or** $\|(d_x^k, d_\lambda^k)\| < \texttt{dtol}$ **then**
      if $\nabla\theta(x^k, \lambda^k) \neq 0$ **then**
        Solve $\left(J_H(x^k, \lambda^k)^T J_H(x^k, \lambda^k) + \sqrt{\theta(x^k, \lambda^k)}\ \mathrm{Id}\right)(d_x^k, d_\lambda^k) = -\nabla\theta(x^k, \lambda^k)$
      else if $\nabla\theta^{\mathrm{feas}}(x^k, \lambda^k) \neq 0$ **then**
        Solve $\left(J_H(x^k, \lambda^k)^T J_H(x^k, \lambda^k) + \sqrt{\theta(x^k, \lambda^k)}\ \mathrm{Id}\right)(d_x^k, d_\lambda^k) = -\nabla\theta^{\mathrm{feas}}(x^k, \lambda^k)$
      else
        Strongly stationary point found. **Terminate.**
      end if
    end if
    $\alpha_k \leftarrow 1$, $\ell \leftarrow 0$
    while $\theta(x^k + \alpha_k\, d_x^k,\ \lambda^k + \alpha_k\, d_\lambda^k) > \theta(x^k, \lambda^k) + c\,\alpha_k\,\nabla\theta(x^k, \lambda^k)^T(d_x^k, d_\lambda^k)$ **and** $\ell < \texttt{maxiter\_ls}$ **do**
      $\alpha_k \leftarrow \alpha_k/2$
      $\ell \leftarrow \ell + 1$
    end while
    $x^{k+1} \leftarrow x^k + \alpha_k\, d_x^k$
    $\lambda^{k+1} \leftarrow \lambda^k + \alpha_k\, d_\lambda^k$
    Compute $\Pi_{\mathcal{K}^*}(\lambda^{k+1})$, $H(x^{k+1}, \lambda^{k+1})$, and $\theta(x^{k+1}, \lambda^{k+1})$
    $k \leftarrow k + 1$
  end while

5 Numerical Experiments

We evaluate the performance of the semi-smooth Newton method (SSN) on instances of the circular cone programming and low-rank matrix completion problems. The experiments in Subsection 5.1 were conducted using MATLAB R2022b on an Intel Core i7-8700 CPU @ 3.20 GHz with 16 GB of RAM, and those in Subsection 5.2 on an Intel Core i9-12900K CPU @ 3.20 GHz with 128 GB of RAM. Tolerances and parameters specific to each problem are provided in the corresponding subsections.

5.1 Circular cone programming

Circular cone programming is an active area of research, with various smoothing methods proposed for solving it; see, for example, [14, 40]. This problem is a natural instance of the generalized simplicial conic programming framework developed in this paper: the circular cone $\mathbb{L}_\omega^n$ is the image of the second-order cone $\mathbb{L}^n$ under a linear transformation, so the projection machinery of

Section 2.1.2 applies directly. The problem reads

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \in \mathbb{L}^n_\omega, \end{array} \tag{5.1}$$

where the circular cone of half-aperture $\omega \in (0, \frac{\pi}{2})$ is

$$\mathbb{L}^n_\omega := \left\{ x = (x_1, u) \in \mathbb{R} \times \mathbb{R}^{n-1} : \|u\| \leq x_1 \tan \omega \right\}. \tag{5.2}$$

Setting

$$M = \begin{pmatrix} \cot \omega & 0^T \\ 0 & \mathrm{Id} \end{pmatrix}, \tag{5.3}$$

we have $\mathbb{L}^n_\omega = M\mathbb{L}^n$. When $\omega = \frac{\pi}{4}$, $\mathbb{L}^n_\omega$ reduces to the standard second-order cone $\mathbb{L}^n$. The dual cone is

$$(\mathbb{L}^n_\omega)^* = \left\{ x = (x_1, u) \in \mathbb{R} \times \mathbb{R}^{n-1} : \|u\| \leq x_1 \cot \omega \right\}. \tag{5.4}$$

Following [40], we generate 10 random instances for each configuration and compare SSN with the smoothing Newton method of [40], hereafter denoted TZ. We test four half-aperture angles $\omega \in \left\{ \frac{\pi}{12}, \frac{\pi}{6}, \frac{\pi}{4}, \frac{\pi}{3} \right\}$. The parameters for TZ are $\lambda = 0.5$, $\delta = 0.8$, $\mu_0 = 10^{-2}$, $\gamma = 10^{-3}$, $\eta_k = 0.95^k$; for SSN we set $c = 0.1$. Both methods use `maxiter` $= 100$, `maxiter_ls` $= 20$, and `tol` $= 10^{-8}$. For $n = 1000$ both methods are compared; for $n = 5000$ and $n = 7500$ only SSN is run, as the execution times of TZ become prohibitive.

We test four starting points:

- **SP0:** $x^0 = 0$, $\lambda^0 = 0$;
- **SP1:** $x^0 \in \mathrm{int}(\mathbb{L}^n_\omega)$, $\lambda^0 = 0$;
- **SP2:** $x^0 = e_1$ (first canonical vector), $\lambda^0 = \mathbf{1}$ (vector of ones);
- **SP3:** x^0 and λ^0 drawn uniformly at random.

In the tables that follow, each row corresponds to a problem instance and each column pair to an angle, with the SSN result on the left and the TZ result on the right.

$\omega = \frac{\pi}{12}$		$\omega = \frac{\pi}{6}$		$\omega = \frac{\pi}{4}$		$\omega = \frac{\pi}{3}$	
SSN - TZ		SSN - TZ		SSN - TZ		SSN - TZ	
2.4955e-09	1.4267e-07	1.76e-09	2.521e-12	1.4894e-12	1.4853e-12	1.3743e-10	2.0924e-11
9.1313e-09	2.4828	2.5711e-09	2.3644e-12	1.473e-12	1.4375e-12	1.8208e-12	1.4809e-11
3.1483e-09	1.5277e-07	1.5646e-12	3.9581e-12	1.3539e-12	1.4404e-12	7.5853e-12	5.755e-11
9.0684e-10	1.7432e-12	1.744e-12	6.2934e-12	1.5753e-12	1.4228e-12	3.4163e-11	1.5875e-10
1.976e-09	1.724e-12	1.4995e-12	3.0344e-12	1.4763e-12	1.4707e-12	3.2682e-12	1.7282e-11
9.0142e-10	1.7609e-12	1.5669e-12	3.147e-12	1.4123e-12	1.4878e-12	2.4459e-10	2.4211e-11
5.8438e-09	2.0803e-07	1.5131e-12	3.5226e-11	1.4924e-12	1.4186e-12	8.7299e-11	5.0756e-12
2.8914e-09	1.8417e-12	1.7722e-12	1.0519e-11	1.5216e-12	1.567e-12	1.5005e-12	6.1043e-12
2.8631e-09	1.8216e-12	1.5705e-12	4.0537e-12	1.8327e-12	1.5473e-12	1.0723e-09	2.498e-10
1.5002e-09	0.0010149	1.8707e-12	5.0224e-12	1.5909e-12	1.5174e-12	4.5275e-10	5.0215e-11

Table 1: Final residuals $\|H(x, \lambda)\|$ for SP0, $n = 1000$.

$\omega = \frac{\pi}{12}$		$\omega = \frac{\pi}{6}$		$\omega = \frac{\pi}{4}$		$\omega = \frac{\pi}{3}$	
SSN - TZ		SSN - TZ		SSN - TZ		SSN - TZ	
1.0785	169.84	1.0446	102.34	0.88914	45.077	1.1596	1.6865
1.0415	177.16	1.0296	104.05	0.89731	47.025	1.0635	1.6968
1.0281	172.22	1.0072	104.81	0.86835	48.52	1.0699	1.6405
1.0514	169.39	1.0422	108.36	0.94868	50.912	1.0326	1.7228
1.0542	168.38	1.1418	105.37	0.89336	47.222	1.0645	1.649
1.0847	168.37	1.0181	102.05	0.88969	51.721	1.1554	1.6744
1.0634	173.52	1.036	103.98	1.2417	51.976	1.1675	1.6994
1.0221	169.38	1.0185	102.68	0.91758	49.256	1.1858	1.9668
1.0259	167.09	1.0573	104.66	0.88781	46.547	1.206	1.6999
1.0396	172.48	1.0945	105.72	0.86437	48.189	1.2136	1.7597

Table 2: Computation times (seconds) for SP0, $n = 1000$.

Table 1 reports the final residuals $\|H(x, \lambda)\|$ for starting point SP0. Both methods achieve the desired precision for most angles; however, for $\omega = \frac{\pi}{12}$, SSN consistently produces smaller residuals. The computation times in Table 2 reveal a substantial advantage for SSN: for $\omega = \frac{\pi}{6}$ and $\frac{\pi}{4}$, SSN is approximately two orders of magnitude faster. Even for $\omega = \frac{\pi}{3}$, where both methods converge quickly, SSN remains faster. Regarding iteration counts, SSN required at most 8 iterations for every instance where it converged, whereas TZ required up to 65, 34, and 7 iterations for $\omega = \frac{\pi}{6}$, $\frac{\pi}{4}$, and $\frac{\pi}{3}$, respectively.

We also ran the same instances starting from SP1. For angles $\omega = \frac{\pi}{12}$, $\frac{\pi}{6}$, and $\frac{\pi}{4}$, SSN encountered strongly stationary points with $\nabla\theta(x, \lambda) = 0$ but $\theta(x, \lambda) \neq 0$, and thus failed to find optimal solutions, while TZ converged successfully. For $\omega = \frac{\pi}{3}$, both methods converged, with SSN being faster.

For $n = 5000$ with starting point SP0, the results were satisfactory: the average computation times were approximately 58, 48, 43, and 58 seconds for $\omega = \frac{\pi}{12}$, $\frac{\pi}{6}$, $\frac{\pi}{4}$, and $\frac{\pi}{3}$, respectively. Residuals were of order 10^{-11} for the first three angles (with one exception at 10^{-8}) and 10^{-10} for $\omega = \frac{\pi}{3}$. For $n = 7500$ with SP0, the average residual was 3.7×10^{-10} and computation times were approximately 186, 162, 138, and 174 seconds for each angle. The behavior for other starting points was consistent with the $n = 1000$ case: SP1 led to strongly stationary points, while SP2 and SP3 performed well.

We also run both methods with $\omega = \frac{5\pi}{12}$ and $n = 1000$ with different starting points, however neither SSN nor TZ were able to converge within the iteration limit.

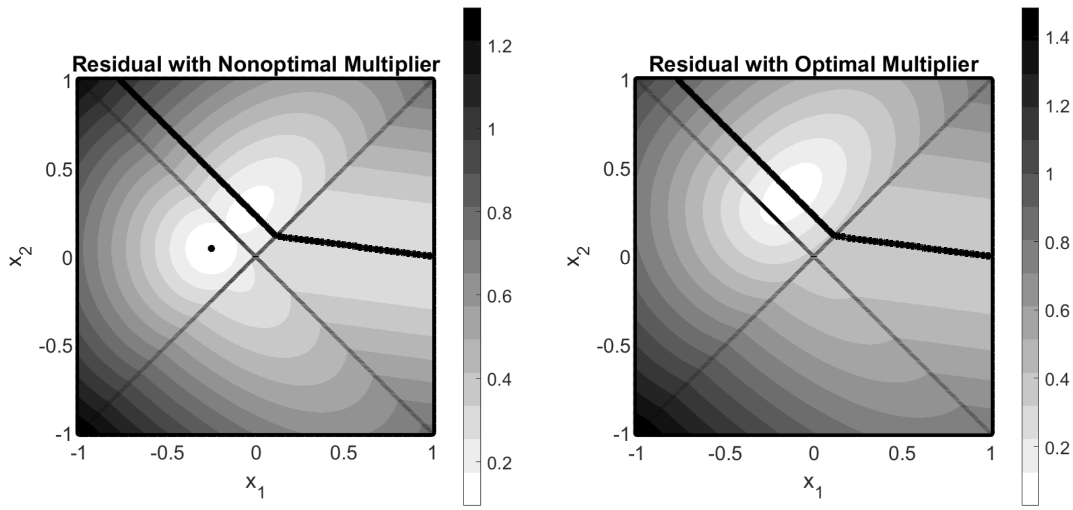

Figure 1: Stationary point for $\theta(x, \lambda)$ with a non optimal and a optimal multiplier.

We conclude this subsection with a remark on the role of strongly stationary points. Figure 1 illustrates how the landscape of stationary points depends on the multiplier. For a second-order conic problem in dimension two, we fix two multipliers: one optimal and one nonoptimal. The boundary of the second-order cone $\mathbb{L}^n$ is shown as a dark gray crossed line passing through the origin; the sublevel sets of the residual $\|H(x, \lambda)\|$ are displayed in shades of gray (decreasing from black to white), and the points satisfying the feasibility condition $A\Pi_{\mathbb{L}^n}(x) - b = 0$ are shown in a continous black line. The right plot illustrates how optimality and feasibility meet at a stationary point. In contrast, the left panel shows that, with a nonoptimal multiplier, the method can converge to a strongly stationary point that is not a solution (isolated point highlighted in black). Developing strategies to detect and escape such points remains an open question and could further improve the method's performance.

5.2 Low-rank matrix completion

We now apply the semi-smooth Newton method to the low-rank matrix completion problem. Finding high-precision solutions is challenging due to the combinatorial nature of the rank constraint; significant efforts have been made in [9, 10]. In [9], a reformulation was introduced that replaces the rank constraint with a matricial quadratic constraint and a trace constraint via the Stiefel manifold, enabling the application of continuous optimization methods. The original problem is

$$\begin{aligned} \min \quad & \frac{1}{2}\|H \circ X - G\|^2 \\ \text{s.t.} \quad & \operatorname{rank}(X) \le \sigma, \end{aligned} \tag{5.5}$$

where σ is a positive integer, $G \in \mathbb{R}^{n\times n}$ is the data matrix, and $H \in \{0,1\}^{n\times n}$ is the mask matrix encoding the observed entries: $H_{ij} = 1$ if G_{ij} is known and $H_{ij} = 0$ otherwise, with $\circ$ denoting the

Hadamard (entrywise) product. By [9, Prop. 1], problem (5.5) is equivalent to

$$\begin{array}{ll} \min & \frac{1}{2}\|H \circ X - G\|^2 \\ \text{s.t.} & X - YX = 0 \\ & Y^2 - Y = 0 \\ & \mathrm{Tr}(Y) \leq \sigma \\ & Y \in \mathbb{S}^n, \quad X \in \mathbb{R}^{n\times n}. \end{array} \tag{5.6}$$

In [9], the authors address a regularized variant of problem (5.6) using two complementary approaches. First, they solve a convex relaxation based on semidefinite programming, which provides lower bounds and high-quality approximate solutions. Second, they reformulate the problem within their mixed-projection conic optimization framework and solve it via a discrete optimization approach, namely an outer-approximation algorithm embedded within a branch-and-bound scheme, yielding global certifiable optimality.

There are two main differences between their approach and ours. First, in [9], the problem is regularized with a Frobenius norm term, which induces strong convexity in the primal variable and ensures strong duality and dual attainment for the resulting convex subproblems arising from the projection-based reformulation. This property is essential for deriving the saddle-point representation exploited in their outer-approximation framework. In contrast, we aim to solve the original problem (5.6) without such regularization. Second, due to the nature of the semismooth Newton SSN method, our approach is designed to compute first-order stationary points, rather than globally optimal solutions. In particular, this difference suggests a possible hybrid strategy: the approximate solutions obtained via the SDP relaxation in [9] can be used as warm starts, which may then be refined to high precision using the SSN method.

For the tests we generate instances following [9]. For each combination of sparsity level $p \in \{10\%, 20\%, 30\%\}$ and rank bound $\sigma \in \{1, 2, 3\}$, we solve 20 random instances of dimension $n = 10$, giving 180 problems in total. SSN is terminated when either the residual or the progress between consecutive iterates falls below 10^{-8}.

Since the sparsity levels are low and the rank constraint is tight, we employ two initialization strategies. The first selects the row of the incomplete matrix G with the largest Frobenius norm and constructs a rank-one initial matrix from it. The second generates perturbations of G with magnitudes depending on the sparsity level, aiming to explore neighborhoods of different stationary points. For each instance, we report the best outcome of the two strategies.

The average $\mathrm{T}_{\max}$ for $\sigma = 2, 3$ was 10000s and 6500s, respectively. However, as shown in Table 3, most of the problems were solved in considerably less time. In constrast in Table 5, from [9], for solving the generated instances their method needed, averaging all problems, times of the order of 2384s, 135226s and 130535s for $\sigma = 1, 2, 3$, respectively. In Table 5 of [9], the authors present a more comprehensive set of experiments on problems of dimensions 20 and 30, which they were able to solve globally with certifiable optimality. In their numerical experiments, they also evaluate the scalability of their SPD-based relaxation on problems with dimensions up to $n = 600$, obtaining good results. Out of the 180 instances tested, SSN found a stationary point in 169, a success rate of approximately 94%. This is notable given the combinatorial complexity of the underlying problem. As expected, the computation time increases with both the sparsity level and the rank bound, but the method solved the majority of instances within a reasonable time. Among the 11 unsolved instances, the method typically stalled due to small progress in the residual, but still achieved residuals on the order of 10^{-4}, a reasonable approximation, particularly since the residuals were not normalized.

p	$\sigma = 1$			$\sigma = 2$		
	0.1	0.2	0.3	0.1	0.2	0.3
	$\leq$ 20s – 100%	$\leq$ 32s – 70%	$\leq$ 200s – 70%	$\leq$ 20s – 100%	$\leq$ 60s – 75%	$\leq$ 500s – 70%
	–	$\leq$ 500s – 85%	$\leq$ 400s – 85%	–	$\leq$ 800s – 90%	$\leq$ 6600s – 85%
	–	$\leq$ 745s – 90%	–	–	$\leq T_{\max}$ – 100%	$\leq T_{\max}$ – 95%

p	$\sigma = 3$		
	0.1	0.2	0.3
	$\leq$ 10s – 90%	$\leq$ 121s – 75%	$\leq$ 431s – 55%
	$\leq T_{\max}$ – 100%	$\leq$ 862s – 90%	$\leq T_{\max}$ – 70%
	–	$\leq T_{\max}$ – 95%	–

Table 3: Cumulative percentages of instances solved within given time thresholds, by rank bound σ and sparsity level p.

6 Concluding Remarks

In this work we extended Robinson's normal equations to the nonlinear conic programming setting, introducing what we call the *conic projection equations*, and established their equivalence with the first-order KKT conditions of the problem. Building on this characterization, we proposed a semi-smooth Newton method and proved its local quadratic convergence, extending previous results from quadratic linearly constrained problems to the general nonlinear conic setting. In the context of generalized simplicial cones, we established the strong semi-smoothness of the projection operator, including the case of rank-deficient linear mappings.

The method was tested on two classes of problems. For circular cone programming, comparisons with the smoothing method of [40] across different half-aperture angles showed that SSN achieves comparable or better precision while being substantially faster, particularly in higher dimensions. For the low-rank matrix completion problem, SSN found high-precision KKT points for the majority of instances, demonstrating that continuous optimization methods based on the reformulation of [9] offer a viable alternative to combinatorial approaches.

An important direction for future work is the development of strategies to detect and escape strongly stationary points that are not solutions, a phenomenon observed in some of our experiments. Addressing this issue could further broaden the applicability and robustness of the method.

Aknowledgments

This work was supported in part by U.S. National Science Foundation NSF, United States of America, under the Grant DMS-2307328, by Conselho Nacional de Desenvolvimento Científico e Tecnológico CNPq, Brazil, and by Fundação de Amparo à Pesquisa do Estado de São Paulo FAPESP, Brazil, under the Grants 2019/13096-2 and 2018/24293-0.

References

[1] Andreani, R., Haeser, G., Mito, L.M., Ramírez, H. and Silveira, T.P.: First- and second-order optimality conditions for second-order cone and semidefinite programming under a constant rank condition. *Math. Program.*, 202(1):473–513, (2023).

[2] Andreani, R., Couto, K.R., Ferreira, O.P. and Haeser, G.: Constraint qualifications and strong global convergence properties of an augmented Lagrangian method on Riemannian manifolds. *SIAM J. Optim.*, 34(2):1799–1825, (2024).

[3] Armijo, N.F., Bello Cruz, Y. and Haeser, G.: On the convergence of iterative schemes for solving a piecewise linear system of equations. *Linear Algebra Appl.*, 665:291–314, (2023).

[4] Armijo, N.F., Bello Cruz, Y. and Haeser, G.: A semi-smooth Newton method for general projection equations applied to the nearest correlation matrix problem. *Optimization*, 1–23, (2025).

[5] Barrios, J.G., Bello Cruz, Y., Ferreira, O.P. and Németh, S.Z.: A semi-smooth Newton method for a special piecewise linear system with application to positively constrained convex programming. *J. Comput. Appl. Math.*, 301:91–100, (2016).

[6] Barrios, J.G., Ferreira, O.P. and Németh, S.Z.: Projection onto simplicial cones by Picard's method. *Linear Algebra Appl.*, 480:27–43, (2015).

[7] Bello Cruz, J., Prudente, L. and Ferreira, O.P.: On the global convergence of the inexact semi-smooth Newton method for absolute value equation. *Comput. Optim. Appl.*, 65(1):93–108, (2016).

[8] Bello Cruz, Y., Ferreira, O.P., Németh, S.Z. and Prudente, L.F.: A semi-smooth Newton method for projection equations and linear complementarity problems with respect to the second-order cone. *Linear Algebra Appl.*, 513:160–181, (2017).

[9] Bertsimas, D., Cory-Wright, R. and Pauphilet, J.: Mixed-projection conic optimization: a new paradigm for modeling rank constraints. *Oper. Res.*, 70(6):3321–3344, (2022).

[10] Bertsimas, D., Cory-Wright, R. and Pauphilet, J.: A new perspective on low-rank optimization. *Math. Program.*, 202:47–92, (2023).

[11] Bonnans, J.F., Shapiro, A.: *Perturbation Analysis of Optimization Problems.* Springer, Bew York, (2000).

[12] Chen, L., Yang, L.-X. and Zhu, J.-Y.: A survey on some recent advances in linear and nonlinear second-order cone programming. *Math. Program. Comput.*, (2025).

[13] Chen, L.: An efficient second-order cone programming approach for dynamic optimal transport. *arXiv preprint arXiv:2505.05424*, (2025).

[14] Chi, X., Wan, Z., Zhu, Z. and Yuan, L.: A nonmonotone smoothing Newton method for circular cone programming. *Optimization*, 65(12):2227–2250, (2016).

[15] Clarke, F.H.: *Optimization and Nonsmooth Analysis.* Society for Industrial and Applied Mathematics, Philadelphia, (1990).

[16] Drusvyatskiy, D., Pataki, G. and Wolkowicz, H.: Coordinate shadows of semidefinite and Euclidean distance matrices. *SIAM J. Optim.*, 25(4):1160–1178, (2015).

[17] Ferreira, O.P. and Németh, S.Z.: Projection onto simplicial cones by a semi-smooth Newton method. *Optim. Lett.*, 9(4):731–741, (2015).

[18] Fukuda, M., Gómez, W., Haeser, G. and Mito, L.M.: Exploiting cone approximations in an augmented Lagrangian method for conic optimization. *Optimization Methods and Software*, 1–18, (2025).

[19] Garrido, M.I.F., Dalzell, A.M. and McArdle, S.: Quantum and classical algorithms for second-order cone programming based on the multiplicative weights update method. *arXiv preprint arXiv:2507.14127*, (2025).

[20] Gill, P.E. and Zhang, M.: A projected-search interior-point method for nonlinearly constrained optimization. *Comput. Optim. Appl.*, 88(1):37–70, (2024).

[21] He, C. and Lu, Z.: A Newton-CG based barrier method for finding a second-order stationary point of nonconvex conic optimization with complexity guarantees. *SIAM J. Optim.*, 33(2):1191–1222, (2023).

[22] Helmberg, C., Rendl, F., Vanderbei, R.J. and Wolkowicz, H.: An interior-point method for semidefinite programming. *SIAM J. Optim.*, 6(2):342–361, (1996).

[23] Horn, R.A. and Johnson, C.R.: *Matrix Analysis*. Cambridge University Press, 2nd edition, (2013).

[24] Im, H., Jung, W.L., Moursi, W.M., Torregrosa-Belin, D. and Wolkowicz, H.: Projection, degeneracy, and singularity degree for spectrahedra. *arXiv preprint arXiv:2407.06408*, (2024).

[25] Izmailov, A.F. and Solodov, M.V.: *Newton-Type Methods for Optimization and Variational Problems*. Springer, (2014).

[26] Kanzow, C., Ferenczi, I. and Fukushima, M.: On the local convergence of semismooth Newton methods for linear and nonlinear second-order cone programs without strict complementarity. *SIAM J. Optim.*, 20(1):297–320, (2009).

[27] Li, X., Sun, D. and Toh, K.-C.: QSDPNAL: a two-phase augmented Lagrangian method for convex quadratic semidefinite programming. *Math. Program. Comput.*, 10:703–743, (2018).

[28] Liang, L., Sun, D. and Toh, K.-C.: A squared smoothing Newton method for semidefinite programming. *Mathematics of Operations Research*, INFORMS, 50(4):2873–2908, (2025).

[29] Lin, Z., Xiong, Z., Ge, D. and Ye, Y.: PDCS: a primal-dual large-scale conic programming solver with GPU enhancements. *arXiv preprint arXiv:2505.00311*, (2025).

[30] Lobo, M.S., Vandenberghe, L., Boyd, S., Lebret, H.: Applications of second-order cone programming. *Linear Algebra Appl.*, 284: 193–228, (1998).

[31] Lu, Y., Ma, H.-M., Xue, D.-Y. and Chen, J.-S.: Augmented Lagrangian method for nonlinear circular conic programs: a local convergence analysis. *Optimization*, 1–33, (2024).

[32] Malick, J. and Sendov, H.S.: Clarke generalized Jacobian of the projection onto the cone of positive semidefinite matrices. *Set-Valued Anal.*, 14(3):273–293, (2006).

[33] Ortega, J.: *Numerical Analysis: A Second Course*. Society for Industrial and Applied Mathematics, Philadelphia, (1987).

[34] Qi, L., Sun, D. and Zhou, G.: A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities. *Math. Program.*, 87(1):1–35, (2000).

[35] Qi, H. and Sun, D.: A quadratically convergent Newton method for computing the nearest correlation matrix. *SIAM J. Matrix Anal. Appl.*, 28(2):360–385, (2006).

[36] Robinson, S.M.: Normal maps induced by linear transformations. *Math. Oper. Res.*, 17(3):691–714, (1992).

[37] Rockafellar, R.T.: *Convex Analysis.* Princeton University Press, Princeton, (1970).

[38] Silveira, T.P. and Haeser, G.: Constant rank-type constraint qualifications and second-order optimality conditions for second-order cone programming. *Proc. Brazilian Soc. Appl. Comput. Math.*, (2025).

[39] Sun, D. and Sun, J.: Löwner's operator and spectral functions in Euclidean Jordan algebras. *Math. Oper. Res.*, 33(2):421–445, (2008).

[40] Tang, J. and Zhou, J.: Improved convergence analysis of a smoothing Newton method for the circular cone programming. *Optimization*, 71(7):2005–2031, (2022).

[41] Tang, J., Zhou, J. and Zhang, H.: An accelerated smoothing Newton method with cubic convergence for weighted complementarity problems. *J. Optim. Theory Appl.*, 196(2):641–665, (2023).

[42] Yang, L., Sun, D. and Toh, K.-C.: SDPNAL+: a majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints. *Math. Program. Comput.*, 7(3):331–366, (2015).

[43] Zhao, C., Xiu, N., Qi, H. and Luo, Z.: A Lagrange–Newton algorithm for sparse nonlinear programming. *Math. Program.*, 195(1):903–928, (2022).

[44] Zheng, J., Varvitsiotis, A., Tan, T.-S. and Lin, W.: A primal-dual framework for symmetric cone programming. *arXiv preprint arXiv:2405.09157*, (2024).

[45] Zhou, Z. and Yu, B.: A smoothing homotopy method for variational inequality problems on polyhedral convex sets. *J. Glob. Optim.*, 58(1):151–168, (2014).