A squared smoothing Newton method for semidefinite programming

Ling Liang[1*], Defeng Sun[3] and Kim-Chuan Toh[1,2]

[1*]Department of Mathematics, National University of Singapore, Singapore.
[2]Institute of Operations Research and Analytics, National University of Singapore, Singapore.
[3]Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong.

*Corresponding author(s). E-mail(s): liang.ling@u.nus.edu;
Contributing authors: defeng.sun@polyu.edu.hk;
mattohkc@nus.edu.sg;

Abstract

This paper proposes a squared smoothing Newton method via the Huber smoothing function for solving semidefinite programming problems (SDPs). We first study the fundamental properties of the matrix-valued mapping defined upon the Huber function. Using these results and existing ones in the literature, we then conduct rigorous convergence analysis and establish convergence properties for the proposed algorithm. In particular, we show that the proposed method is well-defined and admits global convergence. Moreover, under suitable regularity conditions, i.e., the primal and dual constraint nondegenerate conditions, the proposed method is shown to have a superlinear convergence rate. To evaluate the practical performance of the algorithm, we conduct extensive numerical experiments for solving various classes of SDPs. Comparison with the state-of-the-art SDP solver `SDPNAL+` demonstrates that our method is also efficient for computing accurate solutions of SDPs.

Keywords: Semidefinite programming, Smoothing Newton method, Huber function, Constraint nondegeneracy

MSC Classification: 90C06 , 90C22 , 90C25

1 Introduction

A standard primal linear semidefinite programming (SDP) problem is the problem of minimizing a linear function in the space $\mathbb{S}^n$ subject to m linear equality constraints and the essential positive semidefinite constraint $X \in \mathbb{S}^n_+$. Mathematically, the primal SDP has its standard form:

$$\min_{X\in\mathbb{S}^n} \ \langle C, X\rangle \quad \text{s.t.} \quad \mathcal{A}X = b,\ X \in \mathbb{S}^n_+, \tag{1}$$

where $C \in \mathbb{S}^n$, $b \in \mathbb{R}^m$ are given data, and $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$ is a linear mapping given by $\mathcal{A}X := \big(\langle A_1, X\rangle \ \dots \ \langle A_m, X\rangle\big)^T$ for all $X \in \mathbb{S}^n$, with given matrices $A_i \in \mathbb{S}^n$, for $i = 1, \dots, m$, and $\langle\cdot,\cdot\rangle$ denoting the standard trace inner product. Note that the adjoint of $\mathcal{A}$, denoted by $\mathcal{A}^*$, is a linear mapping from $\mathbb{R}^m$ to $\mathbb{S}^n$ defined as $\mathcal{A}^*y := \sum_{i=1}^m y_i A_i$ for $y \in \mathbb{R}^m$. Associated with problem (1), the Lagrangian dual problem of (1) is given by

$$\max_{y\in\mathbb{R}^m, Z\in\mathbb{S}^n} \ \langle b, y\rangle \quad \text{s.t.} \quad \mathcal{A}^*y + Z = C,\ Z \in \mathbb{S}^n_+. \tag{2}$$

In this paper, we assume that the primal problem (1) admits at least one optimal solution and satisfies the Slater's condition, i.e., there exists $\tilde{X} \in \mathbb{S}^n_{++}$ such that $\mathcal{A}\tilde{X} = b$. Under these assumptions, the dual problem (2) has an optimal solution and the dual optimal value is equal to the primal optimal value [1]. Thus, the duality gap between (1) and (2) is zero. As a consequence, the following system of KKT optimality conditions, given as

$$\mathcal{A}X = b, \quad \mathcal{A}^*y + Z = C, \quad X \in \mathbb{S}^n_+,\ Z \in \mathbb{S}^n_+,\ \langle X, Z\rangle = 0, \tag{3}$$

admits a solution. We call an arbitrary triple $(\bar{X}, \bar{y}, \bar{Z}) \in \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n$ a KKT point if it satisfies the KKT conditions in (3). Let $\bar{X}$ be an optimal solution to the primal problem (1), for simplicity, we denote the set of associated Lagrangian multipliers $\mathcal{M}(\bar{X})$ as

$$\mathcal{M}(\bar{X}) := \big\{(y, Z) \in \mathbb{R}^m \times \mathbb{S}^n \ : \ (\bar{X}, y, Z) \text{ is a KKT point}\big\}. \tag{4}$$

Then, under the aforementioned conditions, $\mathcal{M}(\bar{X})$ is a nonempty set.

The research on SDPs has been active for decades and still receives strong attention to date. Indeed, SDP has become one of the fundamental modeling and optimization tools which encompasses a wide range of applications in different fields. The increasing interest in SDP has resulted in fruitful and impressive works in the literature. For theoretical developments and many important applications of SDP in engineering, finance, optimal control, statistics, machine learning, combinatorics, and beyond, we refer the reader to [1–6], to mention just a few. In the next few paragraphs, we will briefly review some influential works on developing efficient solution methods, including interior

[1] Note, however, that the strong duality may not hold for (1) – (2) in general (see e.g., [1]).

point methods (IPMs), first-order methods (FOMs), augmented Lagrangian methods (ALMs), and smoothing Newton methods, for solving SDPs.

Perhaps the most notable and reliable algorithms for solving SDPs are based on the IPMs, see [6, 7] for comprehensive surveys. IPMs admit polynomial worst-case complexity and are able to deliver highly accurate solutions. The essential idea of a primal-dual IPM is to solve the KKT conditions in (3) by solving a sequence of perturbed KKT conditions:

$$\mathcal{A}X = b, \quad \mathcal{A}^*y + Z = C, \quad XZ = \mu I, \quad \mu > 0,$$

with μ being driven to zero. At each iteration of the IPM, one usually needs to solve a linear system whose coefficient matrix (i.e., the Schur complement matrix) is of size $m \times m$ and generally fully dense and ill-conditioned. Thus, IPMs may not be suitable for some practical applications since SDPs arising from such applications are typically of large sizes. To alleviate this issue, many modifications and extensions of IPMs have been developed. Two main approaches are: (a) investigating iterative linear system solvers for computing the search directions [8, 9] in order to handle large-scale linear systems; (b) exploiting the underlying sparsity structures (such as the chordal structure of the aggregate sparsity pattern) in the SDP data for more efficiency [10, 11]. For (a), we note that iterative linear system solvers (such as the PCG method) require carefully designed preconditioners to handle the issue of ill-conditioning. However, finding an effective preconditioner is still a challenging task. For (b), we note that not all SDP problems admit appealing sparsity structures, and hence the range of applications for this approach remains limited. In summary, there exist several critical issues that may stop one from applying an IPM and its variants for solving large-scale SDPs in real-world applications.

FOMs are at the forefront on developing scalable algorithms for solving large-scale optimization problems due to the low per-iteration complexity. For SDPs, we mention some relevant FOMs to capture the picture on recent developments along this direction. In the early 21th century, Helmberg et al. [12] applied a spectral bundle method to solve a special regularized problem for the dual SDP problem (2). The authors proved the convergence of the proposed method under the condition that the trace of any primal optimal solution is fixed. FOMs such as the (accelerated) proximal gradient method are also popular; See for instance the works [13, 14] together with some numerical evaluations of their practical performance. Later, Renegar [15] proposed an FOM which transforms the SDP problem into an equivalent convex optimization problem. But no numerical experiment was conducted in this work and hence the practical performance of the proposed algorithm is unclear. Meanwhile, operator splitting methods and the alternating direction method of multipliers (ADMM) and their variants [16–19] have been demonstrated to be well suited for solving large-scale SDPs, although high accuracy solutions may not be achievable in general. More recently, in order to design storage-efficient algorithms, Yurtsever et al. [20] proposed a conditional-gradient-based augmented Lagrangian method equipped with random sketching techniques for solving SDPs with fixed trace constraints. The method has state-of-the-art performance in memory consumption but whether it is also efficient in computational time for obtaining high accuracy solutions of general SDPs (with or without fixed trace constraints) requires further investigations. Another similar storage-optimal framework can be found in [21]. Indeed, those existing works have shown that FOMs are scalable algorithms. However, a commonly accepted fact for FOMs is that they may not have favorable efficiency for computing accurate solutions or they may even fail to deliver moderately accurate solutions. Therefore, when high quality solutions are needed, FOMs may not be attractive.

Being studied for decades, augmented Lagrangian methods (ALMs) have been shown to be very suitable for solving large-scale optimization problems (including SDPs) efficiently and accurately. Many efficient algorithms based on the ALM have been proposed. For example, Jarre and Rendl [22] developed an augmented primal-dual method. Later, Malick et al. [23] proposed a Moreau-Yosida regularization method for the primal SDP (1). Both methods perform reasonably well on some SDPs with a relatively large number of linear constraints. Zhao et al. [18] developed a dual-based ALM (i.e. the ALM is applied to the dual problem (2)) whose design principle is fundamentally different from the one in [23] by relying on the deep connection between the primal proximal point algorithm and the dual-based ALM [24, 25], and the highly efficient semi-smooth Newton method [26]. The solver developed in [18] has shown very promising practical performance for a large collection of SDPs compared with existing ones. Moreover, practical experience shows that when a good initial point is available, the performance of the algorithm can be further improved. This has motivated the same group of researchers to develop a hybrid framework, namely `SDPNAL+` [27], which combines the dual-based ALM [18] and the ADMM method [17] with some majorization techniques. `SDPNAL+` has demonstrated excellent numerical performance and it can handle additional polyhedral constraints. Lastly, we mention another popular ALM-based approach for solving low-rank SDPs, namely, the Burer-Monteiro (BM) method [28]. The key ingredient of the BM method is to replace the essential constraint $X \in \mathbb{S}^n_+$ with the low-rank factorization $X = RR^T$ which reformulates (1) as a nonconvex optimization problem in $\mathbb{R}^{n\times r}$, where r is a prior bound on the rank of an optimal solution. Many important works on the connection between the factorized problem and the primal SDP (1) have also been published; see for instance [29–31]. However, choosing a suitable r to balance a conservative (large) and an aggressive (small) choice is a delicate task as the former will lead to higher computational cost per iteration while the latter may lead to convergence failure. As a consequence, the practical performance of the BM method is sensitive to the choice of r.

Different from IPMs, one could also consider solving the nonsmooth reformulation of the KKT conditions in (3):

$$\mathcal{F}(X,y,Z) := \begin{pmatrix} \mathcal{A}X - b \\ -\mathcal{A}^*y - Z + C \\ X - \Pi_{\mathbb{S}^n_+}(X - Z) \end{pmatrix} = 0, \quad (X,y,Z) \in \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n, \quad (5)$$

where the conditions $X \in \mathbb{S}^n_+,\ Z \in \mathbb{S}^n_+,\ \langle X, Z\rangle = 0$ are replaced by a single nonsmooth equation $X - \Pi_{\mathbb{S}^n_+}(X - Z) = 0$ and $\Pi_{\mathbb{S}^n_+}(\cdot)$ is the projection operator onto $\mathbb{S}^n_+$. Obviously, the function $\mathcal{F}(\cdot)$ is nonsmooth due to the nonsmoothness of $\Pi_{\mathbb{S}^n_+}(\cdot)$. Thus, the classical Jacobian of $\mathcal{F}(\cdot)$ is not well-defined and the classical Newton method is not applicable for solving (3). Since the B-subdifferential and the Clarke's generalized Jacobian [32] of $\Pi_{\mathbb{S}^n_+}(\cdot)$, namely, $\partial_B \Pi_{\mathbb{S}^n_+}$ and $\partial \Pi_{\mathbb{S}^n_+}$, are well-defined, and $\Pi_{\mathbb{S}^n_+}(\cdot)$ is strongly semi-smooth [33], $\partial_B \mathcal{F}$ and $\partial \mathcal{F}$ are both well-defined, and $\mathcal{F}(\cdot)$ is also strongly semismooth. Then, it is natural to apply the semi-smooth Newton (SSN) method [26] that has a fast local convergence rate under suitable regularity conditions. However, the global convergence of the SSN method needs a valid merit function for which the line search procedure for computing a step size is well-defined. Typically, such a valid merit function should be continuously differentiable or satisfy other stronger conditions. Yet, defining such a merit function from $\mathcal{F}(\cdot)$ directly turns out to be difficult. This motivates us to develop a smoothing Newton method since a suitable merit function can readily be obtained, and its global convergence can be established. Moreover, the smoothing Newton method inherits the strong local convergence properties from the classical Newton method and the semismooth Newton method, under similar regularity conditions.

Existing smoothing Newton methods have been developed and studied extensively in different areas. To mention just a few of these works, the reader is referred to [3, 34–40]. These smoothing methods can be roughly divided into two groups, Jacobian smoothing Newton methods [34, 35] and squared smoothing Newton methods [3, 37, 38]. The convergence analysis of Jacobian smoothing Newton methods strongly depends on the so-called Jacobian consistency property and other strong conditions for the underlying smoothing functions. However, many smoothing functions, such as those functions defined via normal maps [41], do not satisfy these conditions. Furthermore, to get stronger convergence results, it is often useful to add a small perturbation term to the smoothing function. In this case, conditions ensuring those stronger results for the Jacobian smoothing Newton methods are generally not satisfied.

Smoothing Newton methods rely on an appropriate smoothing function for the plus function $\rho(t) := \max\{0, t\}$, $t \in \mathbb{R}$, in order to deal with the nonsmooth projector $\Pi_{\mathbb{S}^n_+}(\cdot)$. To the best of our knowledge, the most notable and commonly used smoothing function is the so-called Chen-Harker-Kanzow-Smale (CHKS) function [42–44], $\xi(\epsilon, t) = (\sqrt{t^2 + 4\epsilon^2} + t)/2,\ (\epsilon, t) \in \mathbb{R} \times \mathbb{R}$, which has been extensively studied and used for solving SDPs and semidefinite linear and nonlinear complementarity problems. For other smoothing functions whose properties have also been well-studied, the reader is referred to [38, 45] for more details. However, it is easy to see that $\xi(\epsilon, t)$ maps any negative number t to a positive one when $\epsilon \neq 0$. Thus, it destroys the possible sparsity structure when evaluating the Jacobian of the merit function. Hence, smoothing Newton methods based on the CHKS function would require more computational effort. To resolve this issue, we propose to use the Huber smoothing function [46] which maps any negative number to zero so that the underlying sparsity structure from the plus function is inherited. To use the Huber smoothing function, we then need to study some fundamental properties, including continuity, differentiability and (strong) semismoothness of the matrix-valued mapping associated with the Huber function.

Although much progress has been made in developing efficient algorithms for solving SDPs in the literature, the efficiency and cost of these methods are still far from satisfactory. This motivates us to study and analyze the squared smoothing Newton method via the Huber smoothing function. To evaluate the efficiency, we implement the algorithm and conduct extensive numerical experiments by solving several classes of SDPs. Our theoretical analysis and numerical results show that the proposed method admits elegant convergence properties and it is efficient for computing accurate solutions. Compared to the ALM based method in [18] for solving an SDP problem, our smoothing Newton method has the advantage that it is applied to a single nonlinear system of equations, whereas the ALM method needs to solve a sequence of subproblems for which the total number of Newton directions required is generally more than the former. Moreover, since the practical performance of the smoothing Newton method for solving SDPs has not been systematically evaluated in the literature, our theoretical and numerical studies can certainly help one to gain better understanding on this class of algorithms.

The rest of the paper is organized as follows. In Section 2, we summary some existing results that will be used in later analysis. Then, we study the continuity, differentiability and semismoothness of the matrix-valued mapping defined upon the Huber function in Section 3. Using these results, we are able to analyze the correctness, global convergence and local fast convergence rate of the proposed squared smoothing Newton method in Section 4. In Section 5, we conduct numerical experiments for solving various classes of SDPs and evaluate the practical performance of the proposed method. Finally, we conclude the paper in Section 6.

2 Preliminary

Let $\mathbb{Y}$ and $\mathbb{Z}$ be any two finite dimensional real vector spaces, each equipped with an inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\|\cdot\|$. Assume that $\mathcal{O} \subseteq \mathbb{Y}$ is an open set and $\Theta : \mathcal{O} \to \mathbb{Z}$ is a locally Lipschitz continuous function on $\mathcal{O}$. Then, Θ is Fréchet-differentiable almost everywhere on $\mathcal{O}$ [47]. Hence, the

B-subdifferential [48] of Θ at $y \in \mathcal{O}$, denoted by $\partial_B \Theta(y)$, is well-defined and given by

$$\partial_B \Theta(y) := \left\{ V : V = \lim_{k \to \infty} \Theta'(y^k), y^k \to y, \Theta'(y^k) \text{ is the F-differential at } y^k \right\}.$$

Moreover, the Clarke's generalized Jacobian [32] of Θ is also well-defined at any $y \in \mathcal{O}$ and is denoted as $\partial \Theta(y) := \operatorname{conv}\{\partial_B \Theta(y)\}$, where "conv" denotes the convex hull of the underlying set.

2.1 Properties of $\Pi_{\mathbb{S}^n_+}(\cdot)$

Denote the matrix-valued mapping $\Omega_0(d) : \mathbb{R}^N \to \mathbb{S}^N$ for any integer $N > 0$ as

$$[\Omega_0(d)]_{ij} = \frac{\rho(d_i) + \rho(d_j)}{|d_i| + |d_j|}, \quad i, j = 1, \ldots, N, \quad \forall d \in \mathbb{R}^N, \tag{6}$$

with the convention that $0/0 := 1$. Let the spectral decomposition of $W \in \mathbb{S}^n$ be

$$W = PDP^T, \quad D = \operatorname{diag}(d), \quad d = (d_1, \ldots, d_n)^T \in \mathbb{R}^n, \quad d_1 \geq \cdots \geq d_n, \tag{7}$$

where $P \in \mathbb{R}^{n \times n}$ is orthogonal. For later usage, we also define the following three index sets: $\alpha := \{i : d_i > 0\}$, $\beta := \{i : d_i = 0\}$, and $\gamma := \{i : d_i < 0\}$, corresponding to the positive, zero and negative eigenvalues of W, respectively.

It is clear that both $\partial_B \Pi_{\mathbb{S}^n_+}(W)$ and $\partial \Pi_{\mathbb{S}^n_+}(W)$ are well-defined for any $W \in \mathbb{S}^n$ (since $\Pi_{\mathbb{S}^n_+}(\cdot)$ is globally Lipschitz continuous with modulus 1 on $\mathbb{S}^n$). We then have the following useful lemma on $\partial_B \Pi_{\mathbb{S}^n_+}(W)$ and $\partial \Pi_{\mathbb{S}^n_+}(W)$ whose proof can be found in [5, Proposition 2.2].

Lemma 1 *Suppose that $W \in \mathbb{S}^n$ has the spectral decomposition (7). Then $V \in \partial_B \Pi_{\mathbb{S}^n_+}(W)$ (respectively, $\partial \Pi_{\mathbb{S}^n_+}(W)$) if and only if there exists $V_{|\beta|} \in \partial_B \Pi_{\mathbb{S}^{|\beta|}_+}(0)$ (respectively, $\partial \Pi_{\mathbb{S}^{|\beta|}_+}(0)$) such that*

$$V(H) = P \begin{pmatrix} \tilde{H}_{\alpha\alpha} & \tilde{H}_{\alpha\beta} & [\Omega_0(d)]_{\alpha\gamma} \circ \tilde{H}_{\alpha\gamma} \\ \tilde{H}_{\alpha\beta}^T & V_{|\beta|}(\tilde{H}_{\beta\beta}) & 0 \\ H_{\alpha\gamma}^T \circ [\Omega_0(d)]_{\alpha\gamma}^T & 0 & 0 \end{pmatrix} P^T,$$

where $\tilde{H} := P^T H P$ for any $H \in \mathbb{S}^n$. In particular, $V_{|\beta|} \in \partial_B \Pi_{\mathbb{S}^{|\beta|}_+}(0)$ if and only if there exist an orthogonal matrix $U \in \mathbb{R}^{|\beta| \times |\beta|}$ and

$$\Omega_{|\beta|} \in \left\{ \Omega \in \mathbb{S}^{|\beta|} : \Omega = \lim_{k \to \infty} \Omega_0(z^k),\ \mathbb{R}^{|\beta|} \ni z^k \to 0,\ z_1^k \geq \cdots \geq z_{|\beta|}^k,\ z^k \neq 0 \right\}$$

such that

$$V_{|\beta|}(Y) = U \left[\Omega_{|\beta|} \circ \left(U^T Y U \right) \right] U^T, \quad \forall Y \in \mathbb{S}^{|\beta|}.$$

Another critical and fundamental concept of $\Pi_{\mathbb{S}^n_+}(\cdot)$ is its semismoothness, which plays an essential role in analyzing the convergence of Newton-type algorithms. Recall that a mapping $\Theta : \mathbb{Y} \to \mathbb{Z}$ is said to be directionally differentiable at the point $y \in \mathbb{Y}$ if the limit, defined by

$$\Theta(y;\ h) := \lim_{t \downarrow 0} \frac{\Theta(y + th) - \Theta(y)}{t},$$

exists for any $h \in \mathbb{Y}$. Then the definition of the semismoothness is given as follows.

Definition 1 Let $\Theta : \mathcal{O} \subseteq \mathbb{Y} \to \mathbb{Z}$ be a locally Lipschitz continuous function. Θ is said to be semismooth at $y \in \mathcal{O}$ if Θ is directionally differentiable at y and for any $V \in \partial\Theta(y + h)$,

$$\|\Theta(y + h) - \Theta(y) - Vh\| = o(\|h\|), \quad h \to 0.$$

Θ is said to be strongly semismooth at $y \in \mathbb{Y}$ if Θ is semismooth at y and for any $V \in \partial\Theta(y + h)$

$$\|\Theta(y + h) - \Theta(y) - Vh\| = O(\|h\|^2), \quad h \to 0.$$

We say that Θ is semismooth (respectively, strongly semismooth) if Θ is semismooth (respectively, strongly semismooth) at every $y \in \mathbb{Y}$.

Semismoothness was originally introduced by Mifflin [49] for functionals. Qi and Sun [26] extended it to vector-valued functions. It is also well-known that the projector $\Pi_{\mathbb{S}^n_+}(\cdot)$ is strongly semismooth on $\mathbb{S}^n$ [33]. Later, we will also show that the smoothed counterpart of $\Pi_{\mathbb{S}^n_+}(\cdot)$ is also strongly semismooth on $\mathbb{S}^n$ (see Proposition 4).

2.2 Equivalent conditions

For the rest of this section, we describe several important conditions related to SDPs and present some connections between these conditions. The presented materials are borrowed directly from the literature. See for instance [3] and the references therein.

First, we recall the general concept of constraint nondegeneracy. Let $g : \mathbb{Y} \to \mathbb{Z}$ be a continuously differentiable function and $\mathcal{C}$ be a nonempty closed convex set in $\mathbb{Z}$. Consider the following feasibility problem:

$$g(\xi) \in \mathcal{C}, \quad \xi \in \mathbb{Y}. \tag{8}$$

Let $\bar{\xi} \in \mathbb{Y}$ be a feasible solution to the above feasibility problem. The tangent cone of $\mathcal{C}$ at the point $g(\bar{\xi})$ is denoted by $\mathcal{T}_{\mathcal{C}}(g(\bar{\xi}))$, and the largest linear subspace of $\mathbb{Y}$ contained in $\mathcal{T}_{\mathcal{C}}(g(\bar{\xi}))$, (i.e., the linearity space of $\mathcal{T}_{\mathcal{C}}(g(\bar{\xi}))$) is denoted by $\mathrm{lin}(\mathcal{T}_{\mathcal{C}}(g(\bar{\xi})))$.

Definition 2 A feasible solution $\bar{\xi}$ for (8) is constraint nondegenerate if

$$g'(\bar{\xi})\mathbb{Y} + \mathrm{lin}(\mathcal{T}_{\mathcal{C}}(g(\bar{\xi}))) = \mathbb{Z}.$$

The concept of degeneracy was introduced by Robinson [50–52] and the term "constraint nondegeneracy" was also coined by Robinson [53]. For nonlinear programming (i.e., $\mathbb{Y}$ is the Euclidean space $\mathbb{R}^m$ and $\mathcal{C} = \{0\}^{m_1} \times \mathbb{R}^{m_2}_+$ with $m = m_1 + m_2$), the constraint nondegeneracy condition is equivalent to the well known linear independence constraint qualification (LICQ) [51].

Let $\mathcal{I}$ denote the identity map on $\mathbb{S}^n$. Applying Definition 2, we see that the primal constraint nondegeneracy holds at a feasible solution $\bar{X} \in \mathbb{S}^n_+$ of the primal problem (1) if

$$\begin{pmatrix} \mathcal{A} \\ \mathcal{I} \end{pmatrix} \mathbb{S}^n_+ + \begin{pmatrix} \{0\} \\ \mathrm{lin}(\mathcal{T}_{\mathbb{S}^n_+}(\bar{X})) \end{pmatrix} = \begin{pmatrix} \mathbb{R}^m \\ \mathbb{S}^n \end{pmatrix}. \tag{9}$$

Similarly, the dual constraint nondegeneracy holds at a feasible solution $(\bar{y}, \bar{Z}) \in \mathbb{R}^n \times \mathbb{S}^n_+$ of the dual problem (2) if

$$\begin{pmatrix} \mathcal{A}^* & \mathcal{I} \\ 0 & \mathcal{I} \end{pmatrix} \begin{pmatrix} \mathbb{R}^m \\ \mathbb{S}^n \end{pmatrix} + \begin{pmatrix} \{0\} \\ \mathrm{lin}(\mathcal{T}_{\mathbb{S}^n_+}(\bar{Z})) \end{pmatrix} = \begin{pmatrix} \mathbb{S}^n \\ \mathbb{S}^n \end{pmatrix}. \tag{10}$$

Without much difficulty, one can show that the primal constraint nondegeneracy (9) is equivalent to $\mathcal{A}\mathrm{lin}(\mathcal{T}_{\mathbb{S}^n_+}(\bar{X})) = \mathbb{R}^m$, and the dual constraint nondegeneracy (10) is equivalent to $\mathcal{A}^*\mathbb{R}^m + \mathrm{lin}(\mathcal{T}_{\mathbb{S}^n_+}(\bar{Z})) = \mathbb{S}^n$. Moreover, if $\bar{X}$ is an optimal solution for (1), then under the primal constraint nondegeneracy condition, we can show that $\mathcal{M}(\bar{X})$ is a singleton [54, Theorem 2].

It is shown in [3, Theorem 18] that the primal and dual constraint nondegeneracy conditions are both necessary and sufficient conditions for the nonsingularity of elements in both $\partial_B \mathcal{F}(\bar{X}, \bar{y}, \bar{Z})$ and $\partial \mathcal{F}(\bar{X}, \bar{y}, \bar{Z})$, where $(\bar{X}, \bar{y}, \bar{Z})$ is a KKT point. Specifically, we have the following theorem.

Theorem 2 ([3]) *Let $(\bar{X}, \bar{y}, \bar{Z}) \in \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n$ be a KKT point. Then the following statements are equivalent:*

1. *The primal constraint nondegeneracy condition holds at $\bar{X}$ and the dual constraint nondegeneracy condition holds at $(\bar{y}, \bar{Z})$, respectively.*
2. *Every element in $\partial \mathcal{F}(\bar{X}, \bar{y}, \bar{Z})$ is nonsingular.*
3. *Every element in $\partial_B \mathcal{F}(\bar{X}, \bar{y}, \bar{Z})$ is nonsingular.*

We next present the concept of strong second-order sufficient condition for linear SDPs. We note that the concept was introduced in [5] for general nonlinear SDPs. To this end, let $(\bar{X}, \bar{y}, \bar{Z})$ be any KKT point. Write $W = \bar{X} - \bar{Z}$, and assume that W has the spectral decomposition (7). Partition P accordingly with respect to the index set α, β and γ as $P = \begin{pmatrix} P_\alpha & P_\beta & P_\gamma \end{pmatrix}$, where

$P_\alpha \in \mathbb{R}^{n\times|\alpha|}$, $P_\beta \in \mathbb{R}^{n\times|\beta|}$, and $P_\gamma \in \mathbb{R}^{n\times|\gamma|}$. Then, by the fact that $\bar{X}$ and $\bar{Z}$ commute, it holds that

$$\bar{X} = P_\alpha \mathrm{diag}(d_1,\ldots,d_{|\alpha|})P_\alpha^T, \quad \bar{Z} = -P_\gamma \mathrm{diag}(d_{1+|\alpha|+|\beta|},\ldots,d_n)P_\gamma^T.$$

For any given matrix $B \in \mathbb{S}^n$, let $B^\dagger$ be the Moore-Penrose pseudoinverse of B. We define the linear-quadratic function $\Gamma_B : \mathbb{S}^n \times \mathbb{S}^n \to \mathbb{R}$ by

$$\Gamma_B(S,H) := 2\left\langle S, HB^\dagger H\right\rangle, \quad \forall (S,H) \in \mathbb{S}^n \times \mathbb{S}^n. \tag{11}$$

Using these notation, the definition of the strong second-order sufficient condition is given as follows.

Definition 3 Let $\bar{X}$ be an optimal solution to the primal problem (1). We say that the strong second-order sufficient condition holds at $\bar{X}$ if $\sup_{(y,Z)\in\mathcal{M}(\bar{X})}\{-\Gamma_{\bar{X}}(-Z,H)\} > 0$ for any $0 \neq H \in \bigcap_{(y,Z)\in\mathcal{M}(\bar{X})} \mathrm{app}(y,Z)$, where $\mathrm{app}(y,Z) := \left\{B \in \mathbb{S}^n \,:\, \mathcal{A}B = 0,\ P_\beta^T B P_\gamma = 0,\ P_\gamma^T B P_\gamma = 0\right\}$.

Finally, we present a result that links the strong second-order sufficient condition and the dual constraint nondegeneracy condition; See [3, Proposition 15] for a proof.

Lemma 3 *Let $(\bar{X},\bar{y},\bar{Z}) \in \mathbb{S}^n\times\mathbb{R}^m\times\mathbb{S}^n$ be a KKT point such that $\mathcal{M}(\bar{X}) = \{(\bar{y},\bar{Z})\}$. Then, the following two statements are equivalent:*

1. *The strong second-order sufficient condition holds at $\bar{X}$.*
2. *The dual constraint nondegeneracy condition holds at $(\bar{y},\bar{Z})$.*

3 The Huber smoothing function

Since the plus function $\rho(t) = \max\{0,t\}$, $t \in \mathbb{R}$ is not differentiable at $t = 0$, we consider its Huber smoothing (or approximation) function that is defined as follows:

$$h(\epsilon,t) = \begin{cases} t - \frac{|\epsilon|}{2} & t > |\epsilon| \\ \frac{t^2}{2|\epsilon|} & 0 \le t \le |\epsilon| \\ 0 & t < 0 \end{cases}, \quad \forall (\epsilon,t) \in \mathbb{R}\backslash\{0\}\times\mathbb{R}, \quad h(0,t) = \rho(t), \quad \forall t \in \mathbb{R}. \tag{12}$$

Clearly, $h(\epsilon,t)$ is continuously differentiable for $\epsilon \neq 0$ and $t \in \mathbb{R}$. Moreover, one can easily check that h is directionally differentiable at $(0,t)$ for any $t \in \mathbb{R}$. Since $\Pi_{\mathbb{S}^n_+}(W) = P\mathrm{diag}(\rho(d_1),\ldots,\rho(d_n))P^T$, we can compute $\Pi_{\mathbb{S}^n_+}(W)$ approximately by evaluating the matrix-valued mapping $\Phi(\epsilon,W)$ that is defined as

$$\Phi(\epsilon,W) := P\mathrm{diag}(h(\epsilon,d_1),\ldots,h(\epsilon,d_n))P^T, \quad \forall (\epsilon,W) \in \mathbb{R}\times\mathbb{S}^n.$$

In this section, we shall study some fundamental properties of Φ. We note that the techniques used in our analysis are not new but mainly borrowed from those in the literature. However, existing techniques were mainly used for analyzing the CHKS-smoothing function. We will show in this section that the same analysis framework is applicable to the Huber smoothing function (12).

Before presenting our results, we need to introduce some useful notation. For any $(\epsilon, d) \in \mathbb{R}\backslash\{0\} \times \mathbb{R}^N$, where $N > 0$ is any dimension, we define the matrix-valued mappings $\Omega : \mathbb{R} \times \mathbb{R}^N \to \mathbb{S}^N$ and $\mathcal{D} : \mathbb{R} \times \mathbb{R}^N \to \mathbb{S}^N$ as follows:

$$\begin{aligned} [\Omega(\epsilon, d)]_{ij} &:= \begin{cases} \frac{h(\epsilon,d_i)-h(\epsilon,d_j)}{d_i-d_j} & d_i \neq d_j \\ h_2'(\epsilon, d_i) & d_i = d_j \end{cases}, \quad 1 \leq i,j \leq N, \\ \mathcal{D}(\epsilon, d) &:= \operatorname{diag}(h_1'(\epsilon, d_1), \ldots, h_1'(\epsilon, d_N)). \end{aligned} \tag{13}$$

Here h_1' and h_2' denote the partial derivatives with respect to the first and the second arguments of h, respectively. Note that $0 \leq [\Omega(\epsilon, d)]_{ij} \leq 1$, $1 \leq i,j \leq N$, for any $(\epsilon, d) \in \mathbb{R} \times \mathbb{R}^N$, and that $0 \leq |h_1'(\epsilon, d_i)| \leq \frac{1}{2}$, $1 \leq i \leq N$, for any $\epsilon \neq 0$.

Let W have the spectral decomposition (7) and $\lambda_1 > \cdots > \lambda_r$ be the distinct eigenvalues of W with multiplicities $m_1, \ldots, m_r$. Define $s_1 := 0$, $s_j := \sum_{i=1}^{j-1} m_j$ for $j = 2, \ldots, r$, and $s_{r+1} := n$. For $j = 1, \ldots, r$, denote the matrices $P_j := \begin{pmatrix} p_{s_j+1} & \ldots & p_{s_{j+1}} \end{pmatrix} \in \mathbb{R}^{n \times m_j}$ and $Q_j := P_j P_j^T \in \mathbb{S}^n$. Then, it is clear that

$$W = \sum_{j=1}^{r} \lambda_j Q_j, \quad \Phi(\epsilon, W) = \sum_{j=1}^{r} h(\epsilon, \lambda_j) Q_j.$$

Now, consider $W + tH$, which admits a decomposition $W + tH = \sum_{i=1}^{n} d_i(t) p_i(t) p_i(t)^T$ with $d_1(t) \geq \cdots \geq d_n(t)$ and $\{p_1(t), \ldots, p_n(t)\}$ forming an orthonormal basis for $\mathbb{R}^n$. Similarly, for $j = 1, \ldots, r$, we can define the matrices $P_j(t) := \begin{pmatrix} p_{s_j+1}(t) & \ldots & p_{s_{j+1}}(t) \end{pmatrix} \in \mathbb{R}^{n \times m_j}$ and $Q_j(t) := P_j(t) P_j(t)^T \in \mathbb{S}^n$. By the definition of directional differential, we may denote

$$d_i'(W;\ H) := \lim_{t \downarrow 0} \frac{d_i(t) - d_i}{t}, \quad \forall H \in \mathbb{S}^n,$$

if the limit exists, for any $i = 1, \ldots, n$.

Proposition 4 *Given any $W \in \mathbb{S}^n$ having the spectral decomposition (7). The following hold:*

1. *For any $\epsilon \neq 0$, Φ is continuously differentiable at (ϵ, W) and it holds that*

$$\Phi'(\epsilon, W)(\tau, H) = P \left[\Omega(\epsilon, d) \circ (P^T H P) + \tau \mathcal{D}(\epsilon, d)\right] P^T, \quad \forall (\tau, H) \in \mathbb{R} \times \mathbb{S}^n, \tag{14}$$

   where $\Omega(\epsilon, d)$, $\mathcal{D}(\epsilon, d)$ are defined in (13).
2. *Φ is locally Lipschitz continuous on $\mathbb{R} \times \mathbb{S}^n$.*

3. Φ *is directionally differentiable at* $(0, W)$*, and for any* $(\tau, H) \in \mathbb{R} \times \mathbb{S}^n$*, it holds that*

$$\Phi'((0,W);\ (\tau,H)) = \frac{1}{2}\sum_{1\le k\neq j\le r}\frac{h(0,\lambda_k)-h(0,\lambda_j)}{\lambda_k-\lambda_j}\,(Q_jHQ_k+Q_kHQ_j) + \sum_{i\in\alpha}\left(d_i'(W;\ H)-\frac{|\tau|}{2}\right)p_ip_i^T + \sum_{i\in\beta}h(\tau,d_i'(W;\ H))p_ip_i^T, \tag{15}$$

where for any $1 \le i \le n$ *with* $d_i = \lambda_j$ *for some* $1 \le j \le r$*, the elements of* $\{d_i'(W;\ H) : i = s_j + 1, \ldots, s_{j+1}\}$ *are the eigenvalues of* $P_j^T H P_j$*, arranged in decreasing order.*

4. Φ *is strongly semismooth on* $\mathbb{R} \times \mathbb{S}^n$*.*

Proof See Appendix A. Interested readers are referred to [37, 55] for similar results for other popular smoothing functions. □

Notice that the matrix P_j is defined up to an orthogonal transformation, i.e., it can be replaced with P_jU, where $U \in \mathbb{R}^{m_j\times m_j}$ is an arbitrary orthogonal matrix. For any given $H \in \mathbb{S}^n$, consider the matrix $P_j^T H P_j$ which admits the following spectral decomposition

$$P_j^T H P_j = U_j \mathrm{diag}(\mu_1, \ldots, \mu_{m_j}) U_j^T, \quad \mu_1 \ge \cdots \ge \mu_{m_j},$$

where $U_j \in \mathbb{R}^{m_j\times m_j}$ is orthogonal. Then, we may replace P_j by P_jU_j. In this way, $P_j^T H P_j$ is always a diagonal matrix whose diagonal entries are arranged in decreasing order. As a consequence, we can easily verify that

$$\Phi'((0,W);\ (\tau,H)) = P\begin{pmatrix} \tilde H_{\alpha\alpha} & \tilde H_{\alpha\beta} & [\Omega_0(d)]_{\alpha\gamma}\circ\tilde H_{\alpha\gamma} \\ \tilde H_{\alpha\beta}^T & \Phi_{|\beta|}(\tau,\tilde H_{\beta\beta}) & 0 \\ [\Omega_0(d)]_{\alpha\gamma}^T\circ H_{\alpha\gamma}^T & 0 & 0\end{pmatrix}P^T - \frac{|\tau|}{2}\sum_{i\in\alpha}p_ip_i^T, \quad \forall(\tau,H)\in\mathbb{R}\times\mathbb{S}^n, \tag{16}$$

where $\tilde H := P^T H P$ and the mapping $\Phi_{|\beta|} : \mathbb{R} \times \mathbb{S}^{|\beta|} \to \mathbb{S}^{|\beta|}$ is defined by replacing the dimension n in the definition of $\Phi : \mathbb{R} \times \mathbb{S}^n \to \mathbb{S}^n$ with $|\beta|$.

Define the mapping $\mathcal{L} : \mathbb{R} \times \mathbb{S}^n \to \mathbb{S}^n$ as

$$\mathcal{L}(\tau, H) := \Phi'((0, W);\ (\tau, H)), \quad (\tau, H) \in \mathbb{R} \times \mathbb{S}^n.$$

Using (16), we see that the mapping $\mathcal{L}(\cdot,\cdot)$ is F-differentiable at (τ, H) if and only if $\tau \ne 0$. Obviously, $\mathcal{L}$ is locally Lipschitz continuous everywhere. Hence, $\partial_B\mathcal{L}(0, 0)$ is well-defined. Then, we have the following useful result that builds an insightful connection between $\partial_B\Phi$ and $\partial_B\mathcal{L}$. To prove the result, we will basically follow the proof of [3, Lemma 4]. However, due to the second term on the right hand side of (16), we need to modify the proof to make the paper self-contained; See Appendix B for more details.

Lemma 5 *Suppose that $W \in \mathbb{S}^n$ has the eigenvalue decomposition (7). Then, it holds that*

$$\partial_B \Phi(0, W) = \partial_B \mathcal{L}(0, 0).$$

Recall that $\Phi_{|\beta|}$ is also locally Lipschitz continuous. Hence, $\partial_B \Phi_{|\beta|}$ and $\partial \Phi_{|\beta|}$ are both well-defined. The following lemma provides an effective way to calculate $\partial_B \Phi(0, W)$ and $\partial \Phi(0, W)$.

Lemma 6 *For any $W \in \mathbb{S}^n$ with the spectral decomposition (7). $V \in \partial_B \Phi(0, W)$ (respectively, $\Phi(0, W)$) if and only if there exist $V_{|\beta|} \in \partial_B \Phi_{|\beta|}(0, 0)$ (respectively, $\partial \Phi_{|\beta|}(0, 0)$) and a scalar $v \in \{-1, 1\}$ (respectively, $[-1, 1]$) such that for all $(\tau, H) \in \mathbb{R} \times \mathbb{S}^n$,*

$$V(\tau, H) = \begin{pmatrix} \tilde{H}_{\alpha\alpha} & \tilde{H}_{\alpha\beta} & [\Omega_0(d)]_{\alpha\gamma} \circ \tilde{H}_{\alpha\gamma} \\ \tilde{H}_{\alpha\beta}^T & V_{|\beta|}(\tau, \tilde{H}_{\beta\beta}) & 0 \\ \tilde{H}_{\alpha\gamma}^T \circ [\Omega_0(d)]_{\alpha\gamma}^T & 0 & 0 \end{pmatrix} P^T + \frac{v\tau}{2} \sum_{i \in \alpha} p_i p_i^T,$$

where $\tilde{H} := P^T H P$ and p_i is the i-th column of P for $i = 1, \ldots, n$. Moreover, $V_{|\beta|} \in \partial_B \Phi_{|\beta|}(0, 0)$ if and only if there exist an orthogonal matrix $U \in \mathbb{R}^{|\beta| \times |\beta|}$ and a symmetric matrix

$$\Omega_{|\beta|} \in \left\{ \Omega \in \mathbb{S}^{|\beta|} : \Omega_{|\beta|} = \lim_{k \to \infty} \Omega(\epsilon^k, z^k), (\epsilon^k, z^k) \to (0, 0), \epsilon^k \neq 0, z_1^k \geq \cdots \geq z_{|\beta|}^k \right\}$$

such that

$$V_{|\beta|}(0, Y) = U \left[\Omega_{|\beta|} \circ (U^T Y U) \right] U^T, \quad \forall Y \in \mathbb{S}^{|\beta|}.$$

Proof We only prove the case for the B-subdifferential as the case for Clarke's generalized Jacobian can be proved in a similar manner. Let the smooth mapping $\Psi : \mathbb{R} \times \mathbb{S}^n \to \mathbb{R} \times \mathbb{S}^n$ be defined as $\Psi(\tau, H) := (\tau, P^T H P)$ for any $(\tau, H) \in \mathbb{R} \times \mathbb{S}^n$. It is clear that $\Psi'(\tau, H) : \mathbb{R} \times \mathbb{S}^n \to \mathbb{R} \times \mathbb{S}^n$ is onto. Define the mapping $\Upsilon : \mathbb{R} \times \mathbb{S}^n \to \mathbb{S}^n$ as follows:

$$\Upsilon(\nu, Y) := P \begin{pmatrix} Y_{\alpha\alpha} & Y_{\alpha\beta} & [\Omega_0(d)]_{\alpha\gamma} \circ Y_{\alpha\gamma} \\ Y_{\alpha\beta}^T & \Phi_{|\beta|}(\nu, Y_{\beta\beta}) & 0 \\ Y_{\alpha\gamma}^T \circ [\Omega_0(d)]_{\alpha\gamma}^T & 0 & 0 \end{pmatrix} P^T \\ - \frac{|\nu|}{2} \sum_{i \in \alpha} p_i p_i^T, \quad \forall (\nu, Y) \in \mathbb{R} \times \mathbb{S}^n.$$

Then by (16), it holds that $\mathcal{L}(\tau, H) = \Upsilon(\Psi(\tau, H))$. Now by [3, Lemma 1] and Lemma 5, we have

$$\partial_B \Phi(0, W) = \partial_B \mathcal{L}(0, 0) = \partial_B \Upsilon(0, 0) \Psi'(0, 0),$$

which proves the first part of the lemma. The expression of $V_{|\beta|} \in \partial_B \Phi(0, 0)$ follows directly from its definition and Proposition 4. Thus, the proof is completed. □

As a corollary of Lemma 1 and Lemma 6, we can verify that for any $V_0 \in \partial_B \Pi_{\mathbb{S}^n_+}(W)$, there exists $V \in \partial_B \Phi(0, W)$ such that

$$V_0(H) = V(0, H), \quad \forall H \in \mathbb{S}^n. \tag{17}$$

We end this section by presenting a useful inequality for elements in $\partial \Phi(0, W)$.

Lemma 7 *Let $W \in \mathbb{S}^n$ have the spectral decomposition (7). Then for any $V \in \partial \Phi(0, W)$,*

$$\langle H - V(0, H), V(0, H)\rangle \geq 0, \quad \forall H \in \mathbb{S}^n.$$

Proof We first consider $V \in \partial_B \Phi(0, W)$. Using Lemma 6, there exist an orthogonal matrix $U \in \mathbb{R}^{|\beta| \times |\beta|}$ and a symmetric matrix $\Omega_{|\beta|} \in \mathbb{S}^{|\beta|}$ such that

$$\begin{aligned}
\langle H - V\left(0, H\right), V\left(0, H\right)\rangle &= \left\langle P^T \left(H - V(0, H)\right) P, P^T V(0, H) P \right\rangle \\
= 2\left\langle (E_{\alpha\gamma} - [\Omega_0(d)]_{\alpha\gamma}) \circ \tilde{H}_{\alpha\gamma}, [\Omega_0(d)]_{\alpha\gamma} \circ \tilde{H}_{\alpha\gamma} \right\rangle & \\
+ \left\langle \tilde{H}_{\beta\beta} - U\left[\Omega_{|\beta|} \circ \left(U^T \tilde{H}_{\beta\beta} U\right)\right] U^T, U\left[\Omega_{|\beta|} \circ \left(U^T \tilde{H}_{\beta\beta} U\right)\right] U^T \right\rangle & \\
= 2\left\langle (E_{\alpha\gamma} - [\Omega_0(d)]_{\alpha\gamma}) \circ \tilde{H}_{\alpha\gamma}, [\Omega_0(d)]_{\alpha\gamma} \circ \tilde{H}_{\alpha\gamma} \right\rangle & \\
+ \left\langle \left(E_{\beta\beta} - \Omega_{|\beta|}\right) \circ \left(U^T \tilde{H}_{\beta\beta} U\right), \Omega_{|\beta|} \circ \left(U^T \tilde{H}_{\beta\beta} U\right) \right\rangle, &
\end{aligned}$$

where $E_{\alpha\gamma}$ and $E_{\beta\beta}$ denote the matrices of all ones in $\mathbb{R}^{|\alpha| \times |\gamma|}$ and $\mathbb{R}^{|\beta| \times |\beta|}$, respectively. Notice that the elements in both the matrices $\Omega_0(d)$ and $\Omega_{|\beta|}$ are all inside the interval $[0, 1]$. Hence, we conclude that $\langle H - V(0, H), V(0, H)\rangle \geq 0$, for any $V \in \partial_B \Phi(0, W)$ and $H \in \mathbb{S}^n$.

Next, we let $V \in \Phi(0, W)$. By Carathéodory's theorem, there exist a positive integer q and $V^i \in \partial_B \Phi(0, W)$, $i = 1, \ldots, q$, such that V is the convex combination of $V^1, \ldots, V^q$. Therefore, there exist nonnegative scalars $t_1, \ldots, t_q$ such that $V = \sum_{i=1}^q t_i V^i$ with $\sum_{i=1}^q t_i = 1$. Define the convex function $\theta(X) = \langle X, X\rangle, X \in \mathbb{S}^n$. By the convexity of θ, we have

$$\begin{aligned}
\langle V(0, H), V(0, H)\rangle = \theta(V(0, H)) &= \theta\left(\sum_{i=1}^q t_i V^i(0, H)\right) \\
&\leq \sum_{i=1}^q t_i \theta(V^i(0, H)) = \langle H, V(0, H)\rangle .
\end{aligned}$$

This completes the proof. □

4 A squared smoothing Newton method

In this section, we shall present our main algorithm, a squared smoothing Newton method via the Huber smoothing function. We then focus on analyzing its correctness, global convergence, and the fast local convergence rate under suitable regularity conditions.

By the results presented in the previous section, we can define the smoothing function for $\mathcal{F}$ based on the smoothing function Φ for $\Pi_{\mathbb{S}^n_+}$. In particular, let $\mathcal{E} : \mathbb{R} \times \mathbb{X} \to \mathbb{R}^m \times \mathbb{S}^n \times \mathbb{S}^n$ be defined as

$$\mathcal{E}(\epsilon, X, y, Z) = \begin{pmatrix} \mathcal{A}X + \kappa_p \, |\epsilon| \, y - b \\ -\mathcal{A}^* y - Z + C \\ (1 + \kappa_c \, |\epsilon|)X - \Phi(\epsilon, X - Z) \end{pmatrix}, \quad \forall (\epsilon, X, y, Z) \in \mathbb{R} \times \mathbb{X}, \quad (18)$$

where $\kappa_p > 0$, $\kappa_c > 0$ are two given constants and $\mathbb{X} := \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n$. Then, $\mathcal{E}$ is a continuously differentiable function around any (ϵ, X, y, Z) for any $\epsilon \neq 0$. Also, it satisfies

$$\mathcal{E}(\epsilon', X', y', Z') \to \mathcal{F}(X, y, Z), \quad \text{as } (\epsilon', X', y', Z') \to (0, X, y, Z).$$

Note that adding the perturbation terms $\kappa_p \, |\epsilon| \, y$ and $\kappa_c \, |\epsilon| \, X$ for constructing the smoothing function of $\mathcal{F}$ is crucial in our algorithmic design since it ensures the correctness of our proposed algorithm (see Lemma 8).

Define the function $\widehat{\mathcal{E}} : \mathbb{R} \times \mathbb{X} \to \mathbb{R} \times \mathbb{R}^m \times \mathbb{S}^n \times \mathbb{S}^n$ by

$$\widehat{\mathcal{E}}(\epsilon, X, y, Z) = \begin{pmatrix} \epsilon \\ \mathcal{E}(\epsilon, X, y, Z) \end{pmatrix}, \quad \forall (\epsilon, X, y, Z) \in \mathbb{R} \times \mathbb{X}. \quad (19)$$

Then solving the nonsmooth equation $\mathcal{F}(X, y, Z) = 0$ is equivalent to solving the following system of nonlinear equations

$$\widehat{\mathcal{E}}(\epsilon, X, y, Z) = 0. \quad (20)$$

Associated with the mapping $\widehat{\mathcal{E}}$, we have the natural merit function $\psi : \mathbb{R} \times \mathbb{X} \to \mathbb{R}_+$ that is defined as

$$\psi(\epsilon, X, y, Z) := \left\| \widehat{\mathcal{E}}(\epsilon, X, y, Z) \right\|^2, \quad \forall (\epsilon, X, y, Z) \in \mathbb{R} \times \mathbb{X}. \quad (21)$$

Given $r \in (0, 1]$, $\hat{r} \in (0, \infty)$ and $\tau \in (0, 1)$, we can define two functions $\zeta : \mathbb{R} \times \mathbb{X} \to \mathbb{R}_+$ and $\eta : \mathbb{R} \times \mathbb{X} \to \mathbb{R}_+$ as follows:

$$\zeta(\epsilon, X, y, Z) = r \min \left\{ 1, \left\| \widehat{\mathcal{E}}(\epsilon, X, y, Z) \right\|^{1+\tau} \right\}, \quad (\epsilon, X, y, Z) \in \mathbb{R} \times \mathbb{X}, \quad (22)$$

and

$$\eta(\epsilon, X, y, Z) = \min \left\{ 1, \hat{r} \left\| \widehat{\mathcal{E}}(\epsilon, X, y, Z) \right\|^{\tau} \right\}, \quad (\epsilon, X, y, Z) \in \mathbb{R} \times \mathbb{X}. \quad (23)$$

Then the inexact smoothing Newton can be described in Algorithm 1.

For the rest of this section, we shall show that Algorithm 1 is well-defined and analyze its convergence properties. Note that the global convergence and

Algorithm 1 A squared smoothing Newton method

Input: $\hat{\epsilon} \in (0,\infty)$, $r \in (0,1)$, $\hat{r} \in (0,\infty)$, $\hat{\eta} \in (0,1)$ be such that $\delta := \sqrt{2}\max\{r\hat{\epsilon}, \hat{\eta}\} < 1$, $\rho \in (0,1)$, $\sigma \in (0,1/2)$, $\tau \in (0,1]$, $\epsilon^0 = \hat{\epsilon}$, $(X^0, y^0, Z^0) \in \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n$.

for $k \geq 0$ **do**

  if $\widehat{\mathcal{E}}(\epsilon^k, X^k, y^k, Z^k) = 0$ **then**

    Output: $(\epsilon^k, X^k, y^k, Z^k)$

  else

    Compute $\eta_k := \eta(\epsilon^k, X^k, y^k, Z^k)$ and $\zeta_k := \zeta(\epsilon^k, X^k, y^k, Z^k)$.

    Solve the following equation

$$\widehat{\mathcal{E}}(\epsilon^k, X^k, y^k, Z^k) + \widehat{\mathcal{E}}'(\epsilon^k, X^k, y^k, Z^k) \begin{pmatrix} \Delta\epsilon^k \\ \Delta X^k \\ \Delta y^k \\ \Delta Z^k \end{pmatrix} = \begin{pmatrix} \zeta_k \hat{\epsilon} \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{24}$$

    approximately such that

$$\begin{aligned} \|\mathcal{R}_k\| \leq & \eta_k \left\| \mathcal{E}(\epsilon^k, X^k, y^k, Z^k) + \mathcal{E}'_\epsilon(\epsilon^k, X^k, y^k, Z^k)\Delta\epsilon^k \right\|, \\ \|\mathcal{R}_k\| \leq & \hat{\eta} \left\| \widehat{\mathcal{E}}(\epsilon^k, X^k, y^k, Z^k) \right\|, \end{aligned} \tag{25}$$

    where $\Delta\epsilon^k := -\epsilon^k + \zeta_k\hat{\epsilon}$ and

$$\mathcal{R}_k := \mathcal{E}(\epsilon^k, X^k, y^k, Z^k) + \mathcal{E}'(\epsilon^k, X^k, y^k, Z^k) \begin{pmatrix} \Delta\epsilon^k \\ \Delta X^k \\ \Delta y^k \\ \Delta Z^k \end{pmatrix}.$$

    Compute ℓ_k as the smallest nonnegative integer ℓ satisfying

$$\begin{aligned} & \psi(\epsilon^k + \rho^\ell \Delta\epsilon^k, X^k + \rho^\ell \Delta X^k, y^k + \rho^\ell \Delta y^k, Z^k + \rho^\ell \Delta Z^k) \\ \leq & [1 - 2\sigma(1-\delta)\rho^\ell]\psi(\epsilon^k, X^k, y^k, Z^k). \end{aligned}$$

    Compute $(\epsilon^{k+1}, X^{k+1}, y^{k+1}, Z^{k+1}) = (\epsilon^k + \rho^{\ell_k}\Delta\epsilon^k, X^k + \rho^{\ell_k}\Delta X^k, y^k + \rho^{\ell_k}\Delta y^k, Z^k + \rho^{\ell_k}\Delta Z^k)$.

  end

end

the fast local convergence rate under the nonsingularity of $\partial_B\widehat{\mathcal{E}}$ or $\partial\widehat{\mathcal{E}}$ at solution points of nonlinear equations (20) are studied extensively in the literature (see e.g., [37]). However, since we are using the Huber function, conditions that ensure the nonsingularity conditions need to be redeveloped. Furthermore, Algorithm 1 allows one to specify the parameter $\tau \in (0,1]$ (which is used to control the rate of convergence) whereas existing results mainly focuse on the case when $\tau = 1$. Consequently, to make the paper self-contained, we also present the details of the analysis for our key results.

4.1 Global convergence

We first show that the linear system (24) is well-defined and solvable for any $\epsilon^k > 0$. Hence the inexactness conditions in (25) are always achievable. This objective can be accomplished by showing that the coefficient matrix of the linear system (24) is nonsingular for any $\epsilon^k > 0$.

Lemma 8 *For any $(\epsilon', X', y', Z') \in \mathbb{R}_{++} \times \mathbb{X}$, there exists an open neighborhood $\mathcal{U}$ of (ϵ', X', y', Z') such that $\widehat{\mathcal{E}}'(\epsilon, X, y, Z)$ is nonsingular for any $(\epsilon, X, y, Z) \in \mathcal{U}$ with $\epsilon \in \mathbb{R}_{++}$.*

Proof Simple calculations give

$$\widehat{\mathcal{E}}'(\epsilon, X, y, Z) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \kappa_p y & \mathcal{A} & \kappa_p \epsilon I_m & 0 \\ 0 & 0 & -\mathcal{A}^* & -\mathcal{I} \\ \kappa_c X - \Phi_1'(\epsilon, X - Z) & (1+\kappa_c\epsilon)\mathcal{I} - \Phi_2'(\epsilon, X - Z) & 0 & \Phi_2'(\epsilon, X - Z) \end{pmatrix},$$

where Φ_1' and Φ_2' denote the partial derivatives of Φ with respect to the first and second arguments, respectively, and $\mathcal{I}$ is the identity map over $\mathbb{S}^n$. To show that $\widehat{\mathcal{E}}'(\epsilon, X, y, Z)$ is nonsingular, it suffices to show that the following system of linear equations

$$\widehat{\mathcal{E}}'(\epsilon, X, y, Z)(\Delta\epsilon, \Delta X, \Delta y, \Delta Z) = 0, \quad (\Delta\epsilon, \Delta X, \Delta y, \Delta Z) \in \mathbb{R} \times \mathbb{X},$$

has only the trivial solution $(\Delta\epsilon, \Delta X, \Delta y, \Delta Z) = (0, 0, 0, 0)$. It is obvious that $\Delta\epsilon = 0$. Since $(1 + \kappa_c\epsilon)\Delta X - \Phi_2'(\epsilon, X - Z)\Delta X + \Phi_2'(\epsilon, X - Z)\Delta Z = 0$, it follows that

$$\Delta X = -\left((1+\kappa_c\epsilon)\mathcal{I} - \Phi_2'(\epsilon, X - Z)\right)^{-1} \Phi_2'(\epsilon, X - Z)\Delta Z. \tag{26}$$

By the equality $-\mathcal{A}^*\Delta y - \Delta Z = 0$, it follows that $\Delta Z = -\mathcal{A}^*\Delta y$ which together with (26) implies that

$$\Delta X = \left((1+\kappa_c\epsilon)\mathcal{I} - \Phi_2'(\epsilon, X - Z)\right)^{-1} \Phi_2'(\epsilon, X - Z)\mathcal{A}^*\Delta y. \tag{27}$$

Using the fact that $\mathcal{A}\Delta X + \kappa_p\epsilon\Delta y = 0$ and (27), we get

$$\left(\kappa_p\epsilon I_m + \mathcal{A}\left((1+\kappa_c\epsilon)\mathcal{I} - \Phi_2'(\epsilon, X - Z)\right)^{-1} \Phi_2'(\epsilon, X - Z)\mathcal{A}^*\right)\Delta y = 0.$$

Since $0 \preceq \Phi_2'(\epsilon, X - Z) \preceq \mathcal{I}$, it is easy to show that the coefficient matrix in the above linear system is symmetric positive definite. Thus, $\Delta y = 0$, which further implies that $\Delta X = \Delta Z = 0$. Therefore, the proof is completed. □

The next task is to show that the line search procedure is well-defined, i.e., ℓ_k is finite for all $k \geq 0$. We will see that the inexactness condition $\|\mathcal{R}_k\| \leq$

$\hat{\eta}\left\|\widehat{\mathcal{E}}(\epsilon^k, X^k, y^k, Z^k)\right\|$ plays a fundamental role in our analysis. We also note that the inexactness condition

$$\|\mathcal{R}_k\| \leq \eta_k \left\|\mathcal{E}(\epsilon^k, X^k, y^k, Z^k) + \mathcal{E}'_\epsilon(\epsilon^k, X^k, y^k, Z^k)\Delta\epsilon^k\right\|$$

does not affect the correctness of Algorithm 1 but will be crucial in analyzing the local convergence rate of Algorithm 1.

Lemma 9 *For any $(\epsilon', X', y', Z') \in \mathbb{R}_{++} \times \mathbb{X}$, there exist an open neighborhood $\mathcal{U}$ of (ϵ', X', y', Z') and a positive scalar $\bar{\alpha} \in (0, 1]$ such that for any $(\epsilon, X, y, Z) \in \mathcal{U}$ with $\epsilon \in \mathbb{R}_{++}$ and $\alpha \in (0, \bar{\alpha}]$, it holds that*

$$\psi(\epsilon + \alpha\Delta\epsilon, X + \alpha\Delta X, y + \alpha\Delta y, Z + \alpha\Delta Z) \leq [1 - 2\sigma(1-\delta)\alpha]\, \psi(\epsilon, X, y, Z),$$

where $\boldsymbol{\Delta} := (\Delta\epsilon;\ \Delta X;\ \Delta y;\ \Delta Z) \in \mathbb{R} \times \mathbb{X}$ satisfies

$$\Delta\epsilon = -\epsilon + \zeta(\epsilon, X, y, Z)\hat{\epsilon}, \quad \left\|\mathcal{E}(\epsilon, X, y, Z) + \mathcal{E}'(\epsilon, X, y, Z)\boldsymbol{\Delta}\right\| \leq \hat{\eta}\left\|\widehat{\mathcal{E}}(\epsilon, X, y, Z)\right\|.$$

Proof Since $\epsilon' \in \mathbb{R}_{++}$, by Lemma 8, $\widehat{\mathcal{E}}'(\epsilon', X', y', Z')$ is nonsingular. Since $\widehat{\mathcal{E}}$ is continuously differentiable around (ϵ', X', y', Z'), there exists an open neighborhood $\mathcal{U}$ of (ϵ', X', y', Z') such that for any (ϵ, X, y, Z), $\widehat{\mathcal{E}}'(\epsilon, X, y, Z)$ is nonsingular. Therefore, the existence of $\boldsymbol{\Delta}$ is guaranteed.

Denote $\mathcal{R}(\epsilon, X, y, Z) := \mathcal{E}(\epsilon, X, y, Z) + \mathcal{E}'(\epsilon, X, y, Z)\boldsymbol{\Delta}$. Then one can verify that $(\Delta\epsilon, \Delta X, \Delta y, \Delta Z)$ is the unique solution of the following equation

$$\widehat{\mathcal{E}}(\epsilon, X, y, Z) + \widehat{\mathcal{E}}'(\epsilon, X, y, Z)\boldsymbol{\Delta} = \begin{pmatrix} \zeta(\epsilon, X, y, Z)\hat{\epsilon} \\ \mathcal{R}(\epsilon, X, y, Z) \end{pmatrix}.$$

Hence, it holds that

$$\begin{aligned}
&\langle \nabla\psi(\epsilon, X, y, Z), \boldsymbol{\Delta}\rangle = \left\langle 2\nabla\widehat{\mathcal{E}}(\epsilon, X, y, Z)\widehat{\mathcal{E}}(\epsilon, X, y, Z), \boldsymbol{\Delta}\right\rangle \\
=&\left\langle 2\widehat{\mathcal{E}}(\epsilon, X, y, Z), \begin{pmatrix} \zeta(\epsilon, X, y, Z)\hat{\epsilon} \\ \mathcal{R}(\epsilon, X, y, Z) \end{pmatrix} - \widehat{\mathcal{E}}(\epsilon, X, y, Z)\right\rangle \\
=&-2\psi(\epsilon, X, y, Z) + 2\zeta(\epsilon, X, y, Z)\epsilon\hat{\epsilon} + 2\left\langle \mathcal{R}(\epsilon, X, y, Z), \mathcal{E}(\epsilon, X, y, Z)\right\rangle \\
\leq&-2\psi(\epsilon, X, y, Z) + 2r\min\{1, \psi(\epsilon, X, y, Z)^{(1+\tau)/2}\}\epsilon\hat{\epsilon} \\
&+ 2\hat{\eta}\psi(\epsilon, X, y, Z)^{1/2}\left\|\mathcal{E}(\epsilon, X, y, Z)\right\|.
\end{aligned} \tag{28}$$

We consider two possible cases when $\psi(\epsilon, X, y, Z) > 1$ and $\psi(\epsilon, X, y, Z) \leq 1$.

If $\psi(\epsilon, X, y, Z) > 1$, then (28) implies that

$$\begin{aligned}
&\langle \nabla\psi(\epsilon, X, y, Z), \boldsymbol{\Delta}\rangle \\
\leq&-2\psi(\epsilon, X, y, Z) + 2r\epsilon\hat{\epsilon} + 2\hat{\eta}\psi(\epsilon, X, y, Z)^{1/2}\sqrt{\psi(\epsilon, X, y, Z) - \epsilon^2} \\
\leq&-2\psi(\epsilon, X, y, Z) + 2\max\{r\hat{\epsilon}, \hat{\eta}\}\left(\epsilon + \psi(\epsilon, X, y, Z)^{1/2}\sqrt{\psi(\epsilon, X, y, Z) - \epsilon^2}\right) \\
\leq&-2\psi(\epsilon, X, y, Z) + 2\max\{r\hat{\epsilon}, \hat{\eta}\}\psi(\epsilon, X, y, Z) \\
=&\,2\left(\sqrt{2}\max\{r\hat{\epsilon}, \hat{\eta}\} - 1\right)\psi(\epsilon, X, y, Z).
\end{aligned}$$

If $\psi(\epsilon, X, y, Z) \le 1$, then (28) implies that

$$\begin{aligned}
&\langle \nabla\psi(\epsilon, X, y, Z), \mathbf{\Delta} \rangle \\
\le & -2\psi(\epsilon, X, y, Z) + 2r\psi(\epsilon, X, y, Z)^{(1+\tau)/2}\epsilon\hat{\epsilon} \\
& + 2\hat{\eta}\psi(\epsilon, X, y, Z)^{1/2}\sqrt{\psi(\epsilon, X, y, Z) - \epsilon^2} \\
\le & -2\psi(\epsilon, X, y, Z) + 2r\psi(\epsilon, X, y, Z)\hat{\epsilon} + 2\hat{\eta}\psi(\epsilon, X, y, Z)^{1/2}\sqrt{\psi(\epsilon, X, y, Z) - \epsilon^2} \\
\le & -2\psi(\epsilon, X, y, Z) \\
& + 2\max\{r\hat{\epsilon}, \hat{\eta}\}\psi(\epsilon, X, y, Z)^{1/2}\left(\epsilon\psi(\epsilon, X, y, Z)^{1/2} + \sqrt{\psi(\epsilon, X, y, Z) - \epsilon^2}\right) \\
\le & -2\psi(\epsilon, X, y, Z) + 2\max\{r\hat{\epsilon}, \hat{\eta}\}\psi(\epsilon, X, y, Z) \\
= & \, 2\left(\sqrt{2}\max\{r\hat{\epsilon}, \hat{\eta}\} - 1\right)\psi(\epsilon, X, y, Z).
\end{aligned}$$

For both cases, we always have that

$$\langle \nabla\psi(\epsilon, X, y, Z), \mathbf{\Delta} \rangle \le 2\left(\sqrt{2}\max\{r\hat{\epsilon}, \hat{\eta}\} - 1\right)\psi(\epsilon, X, y, Z). \tag{29}$$

Since $\nabla\psi(\cdot)$ is uniformly continuous on $\mathcal{U}$, for all $(\epsilon, X, y, Z) \in \mathcal{U}$ with $\epsilon > 0$, we have from the Taylor expansion that

$$\psi(\epsilon+\alpha\Delta\epsilon, X+\alpha\Delta X, y+\alpha\Delta y, Z+\alpha\Delta Z) = \psi(\epsilon, X, y, Z)+\alpha\langle \nabla\psi(\epsilon, X, y, Z), \mathbf{\Delta} \rangle+o(\alpha). \tag{30}$$

Combining (29) and (30), it holds that

$$\begin{aligned}
&\psi(\epsilon + \alpha\Delta\epsilon, X + \alpha\Delta X, y + \alpha\Delta y, Z + \alpha\Delta Z) \\
&= \psi(\epsilon, X, y, Z) + 2\alpha(\delta - 1)\psi(\epsilon, X, y, Z) + o(\alpha) \\
&= [1 - 2\sigma(1 - \delta)\alpha]\,\psi(\epsilon, X, y, Z) + o(\alpha),
\end{aligned}$$

and the proof is completed. □

As showed before, the nonsingularity of the coefficient matrix in (24) requires ϵ^k to be positive. The next lemma shows that Algorithm 1 generates ϵ^k that is lower bounded by $\zeta(\epsilon^k, X^k, y^k, Z^k)\hat{\epsilon}$. Thus, as long as the optimal solution is not found, ϵ^k remains positive. To present the lemma, we need to define the following set

$$\mathcal{N} := \{(\epsilon, X, y, Z) \;:\; \epsilon \ge \zeta(\epsilon, X, y, Z)\hat{\epsilon}\}.$$

Lemma 10 *Suppose that for a given $k \ge 0$, $\epsilon^k \in \mathbb{R}_{++}$ and $(\epsilon^k, X^k, y^k, Z^k) \in \mathcal{N}$. Then, for any $\alpha \in [0, 1]$ such that*

$$\psi(\epsilon^k+\alpha\Delta\epsilon^k, X^k+\alpha\Delta X^k, y^k+\alpha\Delta y^k, Z^k+\alpha\Delta Z^k) \le [1-2\sigma(1-\delta)\alpha]\psi(\epsilon^k, X^k, y^k, Z^k),$$

it holds that

$$(\epsilon^k + \alpha\Delta\epsilon^k, X^k + \alpha\Delta X^k, y^k + \alpha\Delta y^k, Z^k + \alpha\Delta Z^k) \in \mathcal{N}.$$

Proof Since $(\epsilon^k, X^k, y^k, Z^k) \in \mathcal{N}$, we have directly from the definition of $\mathcal{N}$ that $\epsilon^k \geq \zeta(\epsilon^k, X^k, y^k, Z^k)\hat{\epsilon}$. Recall that $\Delta\epsilon^k = -\epsilon^k + \zeta(\epsilon^k, X^k, y^k, Z^k)\hat{\epsilon} \leq 0$. It holds that

$$\begin{aligned}
&\epsilon^k + \alpha\Delta\epsilon^k - \zeta(\epsilon^k + \alpha\Delta\epsilon^k, X^k + \alpha\Delta X^k, y^k + \alpha\Delta y^k, Z^k + \alpha\Delta Z^k)\hat{\epsilon} \\
&\geq \epsilon^k + \Delta\epsilon^k - \zeta(\epsilon^k + \alpha\Delta\epsilon^k, X^k + \alpha\Delta X^k, y^k + \alpha\Delta y^k, Z^k + \alpha\Delta Z^k)\hat{\epsilon} \\
&= \zeta(\epsilon^k, X^k, y^k, Z^k)\hat{\epsilon} - \zeta(\epsilon^k + \alpha\Delta\epsilon^k, X^k + \alpha\Delta X^k, y^k + \alpha\Delta y^k, Z^k + \alpha\Delta Z^k)\hat{\epsilon} \\
&\geq 0,
\end{aligned}$$

which indeed implies that $(\epsilon^k + \alpha\Delta\epsilon^k, X^k + \alpha\Delta X^k, y^k + \alpha\Delta y^k, Z^k + \alpha\Delta Z^k) \in \mathcal{N}$. This completes the proof. □

With those previous results, we can now establish the global convergence of Algorithm 1.

Theorem 11 *Algorithm 1 is well-defined and generates an infinite sequence* $\{(\epsilon^k, X^k, y^k, Z^k)\} \subseteq \mathcal{N}$ *with the property that any accumulation point* $(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z})$ *of* $\{(\epsilon^k, X^k, y^k, Z^k)\}$ *is a solution of* $\widehat{\mathcal{E}}(\epsilon, X, y, Z) = 0$*,* $(\epsilon, X, y, Z) \in \mathbb{R} \times \mathbb{X}$*.*

Proof It follows from Lemma 9 and Lemma 10 that Algorithm 1 is well-defined and generates an infinite sequence containing in $\mathcal{N}$. Since the line-search scheme is well-defined, it is obvious that

$$\psi(\epsilon^{k+1}, X^{k+1}, y^{k+1}, Z^{k+1}) < \psi(\epsilon^k, X^k, y^k, Z^k), \quad \forall\, k \geq 0.$$

The monotonically decreasing property of the sequence $\{\psi(\epsilon^k, X^k, y^k, Z^k)\}$ then implies that $\{\zeta_k\}$ is also monotonically decreasing. Hence, there exist $\bar{\psi}$ and $\bar{\zeta}$ such that

$$\psi(\epsilon^k, X^k, y^k, Z^k) \to \bar{\psi}, \quad \zeta_k \to \bar{\zeta}, \quad k \to \infty.$$

Let $(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z})$ be any accumulation point (if it exists) of $\{(\epsilon^k, X^k, y^k, Z^k)\}$. By taking a subsequence if necessary, we may assume that $\{(\epsilon^k, X^k, y^k, Z^k)\}$ converges to $(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z})$. Then, by the continuity of $\psi(\cdot)$, it holds that

$$\bar{\psi} = \psi(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z}), \quad \bar{\zeta} = \zeta(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z}), \quad (\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z}) \in \mathcal{N}.$$

Note that $\bar{\psi} \geq 0$, we now prove that $\bar{\psi} = 0$ by contradiction. To this end, suppose that $\bar{\psi} > 0$. Then $\bar{\zeta} > 0$. By the fact that $(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z}) \in \mathcal{N}$, we see that $\bar{\epsilon} \in \mathbb{R}_{++}$. Therefore, by Lemma 8, we see that there exists a neighborhood of $(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z})$, denoted by $\mathcal{U}$, such that $\widehat{\mathcal{E}}'(\epsilon, X, y, Z)$ is nonsingular for any $(\epsilon, X, y, Z) \in \mathcal{U}$ with $\epsilon > 0$. Note that for k sufficiently large, we have that $(\epsilon^k, X^k, y^k, Z^k)$ belongs to $\mathcal{U}$ with $\epsilon^k > 0$. Furthermore, by Lemma 9, there also exists $\bar{\alpha} \in (0, 1]$ such that for any $\alpha \in (0, \bar{\alpha}]$,

$$\psi(\epsilon^k+\alpha\Delta\epsilon^k, X^k+\alpha\Delta X^k, y^k+\alpha\Delta y^k, Z^k+\alpha\Delta Z^k) \leq [1-2\sigma(1-\delta)\alpha]\psi(\epsilon^k, X^k, y^k, Z^k)$$

for $k \geq 0$ sufficiently large. The existence of the fixed number $\bar{\alpha} \in (0, 1]$ further indicates that there exists a nonnegative integer ℓ such that $\rho^\ell \in (0, \bar{\alpha}]$ and $\rho^{\ell_k} \geq \rho^\ell$ for all k sufficiently large. Therefore, it holds that

$$\begin{aligned}
\psi(\epsilon^{k+1}, X^{k+1}, y^{k+1}, Z^{k+1}) &\leq [1 - 2\sigma(1-\delta)\rho^{\ell_k}]\psi(\epsilon^k, X^k, y^k, Z^k) \\
&\leq [1 - 2\sigma(1-\delta)\rho^{\ell}]\psi(\epsilon^k, X^k, y^k, Z^k),
\end{aligned}$$

for all sufficiently large k. The above inequality implies that $\bar{\psi} \leq 0$, which is a contradiction to the assumption $\bar{\phi} > 0$. Hence, $\bar{\psi} = 0$, which implies that $\widehat{\mathcal{E}}(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z}) = 0$. The proof is completed. □

4.2 Superlinear convergence rate

We next establish the superlinear convergence rate of Algorithm 1 under certain regularity conditions. The following lemma is useful for characterizing the nonsingularity of an element in $\partial\widehat{\mathcal{E}}$ at any accumulation point.

Lemma 12 *Suppose that $(\bar{X}, \bar{y}, \bar{Z}) \in \mathbb{X}$ is a KKT-point. Let $\bar{X} - \bar{Z} := \bar{W} \in \mathbb{S}^n$ and $\bar{V} \in \partial\Phi(0, \bar{W})$. Then, for any ΔX and ΔZ in $\mathbb{S}^n$ such that $\Delta X = \bar{V}(0, \Delta X - \Delta Z)$, it holds that $\langle \Delta X, \Delta Z\rangle \le \Gamma_{\bar{X}}(-\bar{Z}, \Delta X)$, where $\Gamma_{\bar{X}}$ is defined as in (11).*

Proof Suppose that $\bar{W}$ has the spectral decomposition (7). Denote $\Delta\tilde{X} := P^T\Delta X P$ and $\Delta\tilde{Z} = P^T\Delta Z P$. For the index set β, define $\Phi_{|\beta|}$ as before. Then, by Lemma 6, there exists $V_{|\beta|} \in \partial\Phi_{|\beta|}(0,0)$ such that

$$V(0, \Delta X - \Delta Z) = P \begin{pmatrix} \Delta\tilde{H}_{\alpha\alpha} & \Delta\tilde{H}_{\alpha\beta} & [\Omega_0(d)]_{\alpha\gamma} \circ \Delta\tilde{H}_{\alpha\gamma} \\ \Delta\tilde{H}_{\alpha\beta}^T & V_{|\beta|}(0, \Delta\tilde{H}_{\beta\beta}) & 0 \\ \Delta\tilde{H}_{\alpha\gamma}^T \circ [\Omega_0(d)]_{\alpha\gamma}^T & 0 & 0 \end{pmatrix} P^T,$$

where $\Delta\tilde{H} = \Delta\tilde{X} - \Delta\tilde{Z}$. Comparing both sides of the relation $\Delta X = V(0, \Delta X - \Delta Z)$ yields that

$$\Delta\tilde{Z}_{\alpha\alpha} = 0, \quad \Delta\tilde{Z}_{\alpha\beta} = 0, \quad \Delta\tilde{X}_{\beta\gamma} = 0, \quad \Delta\tilde{X}_{\gamma\gamma} = 0,$$

and that

$$\Delta\tilde{X}_{\beta\beta} = V_{|\beta|}(0, \Delta\tilde{X}_{\beta\beta} - \Delta\tilde{Z}_{\beta\beta}),\ \Delta\tilde{X}_{\alpha\gamma} - [\Omega_0(d)]_{\alpha\gamma} \circ \Delta\tilde{X}_{\alpha\gamma} = [\Omega_0(d)]_{\alpha\gamma} \circ \Delta\tilde{Z}_{\alpha\gamma}.$$

By Lemma 7, it holds that

$$\begin{aligned} &\left\langle \Delta\tilde{X}_{\beta\beta}, -\Delta\tilde{Z}_{\beta\beta} \right\rangle \\ &= \left\langle V_{|\beta|}(0, \Delta\tilde{X}_{\beta\beta} - \Delta\tilde{Z}_{\beta\beta}), \left(\Delta\tilde{X}_{\beta\beta} - \Delta\tilde{Z}_{\beta\beta}\right) - V_{|\beta|}(0, \Delta\tilde{X}_{\beta\beta} - \Delta\tilde{Z}_{\beta\beta}) \right\rangle \ge 0 \end{aligned}$$

which implies that

$$\begin{aligned} \langle \Delta X, \Delta Z\rangle = \left\langle \tilde{\Delta}X, \tilde{\Delta}Z \right\rangle &= \left\langle \Delta\tilde{X}_{\beta\beta}, \Delta\tilde{Z}_{\beta\beta} \right\rangle + 2\left\langle \Delta\tilde{X}_{\alpha\gamma}, \Delta\tilde{Z}_{\alpha\gamma} \right\rangle \\ &\le 2\left\langle \Delta\tilde{X}_{\alpha\gamma}, \Delta\tilde{Z}_{\alpha\gamma} \right\rangle. \end{aligned}$$

However, simple calculations show that $\Gamma_{\bar{X}}(-\bar{Z}, \Delta X) = 2\left\langle \Delta\tilde{X}_{\alpha\gamma}, \Delta\tilde{Z}_{\alpha\gamma} \right\rangle$. This proves the lemma. □

Using the above lemma, we can establish the following equivalent relations.

Proposition 13 *Let $(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z})$ be such that $\widehat{\mathcal{E}}(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z}) = 0$. Then the following statements are equivalent to each other.*

1. *The primal constraint nondegenerate condition holds at $\bar{X}$, and the dual constraint nondegenerate condition holds at $(\bar{y}, \bar{Z})$.*
2. *Every element in $\partial_B\widehat{\mathcal{E}}(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z})$ is nonsingular.*
3. *Every element in $\partial\widehat{\mathcal{E}}(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z})$ is nonsingular.*

Proof It is obvious that part 3 implies part 2. From (17), we see that for any $V_0 \in \partial_B \Pi_{\mathbb{S}^n_+}(W)$, $W \in \mathbb{S}^n$, there exists $V \in \partial_B \Phi(0, W)$ such that

$$V_0(H) = V(0, H), \quad \forall H \in \mathbb{S}^n.$$

Therefore, by Theorem 2, part 2 implies part 1. So, it suffices to show that part 1 implies part 3.

Now, suppose that part 1 holds. Recall that the primal constraint nondegeneracy condition at $\bar{X}$ implies that $\mathcal{M}(\bar{X}) = \{(\bar{y}, \bar{Z})\}$, i.e., the dual multiplier is unique. Now by using Lemma 3, we can see that the dual constraint nondegeneracy at $(\bar{y}, \bar{Z})$ implies that the strong second-order sufficient condition holds at $\bar{X}$. In particular, it holds that

$$-\Gamma_{\bar{X}}(-\bar{Z}, H) > 0, \quad \forall 0 \neq H \in \mathrm{app}(\bar{y}, \bar{Z}). \tag{31}$$

Let $U \in \partial \widehat{\mathcal{E}}(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z})$ (recall that $\bar{\epsilon} = 0$). Then there exists $V \in \partial \Phi(0, \bar{X} - \bar{Z})$ such that

$$U(\Delta\epsilon, \Delta X, \Delta y, \Delta Z) = \begin{pmatrix} \Delta\epsilon \\ \mathcal{A}\Delta X \\ -\mathcal{A}^*\Delta y - \Delta Z \\ \Delta X - V(\Delta\epsilon, \Delta X - \Delta Z) \end{pmatrix}, \quad (\Delta\epsilon, \Delta X, \Delta y, \Delta Z) \in \mathbb{R} \times \mathbb{X}.$$

To show that U is nonsingular, it suffices to show that $U(\Delta\epsilon, \Delta X, \Delta y, \Delta Z) = 0$ implies that $(\Delta\epsilon, \Delta X, \Delta y, \Delta Z) = 0$. So, let us assume that $U(\Delta\epsilon, \Delta X, \Delta y, \Delta Z) = 0$, i.e.,

$$\begin{pmatrix} \Delta\epsilon \\ \mathcal{A}\Delta X \\ -\mathcal{A}^*\Delta y - \Delta Z \\ \Delta X - V(\Delta\epsilon, \Delta X - \Delta Z) \end{pmatrix} = 0. \tag{32}$$

We first prove that $\Delta X = 0$ by contradiction. To this end, we assume that $\Delta X \neq 0$. By Lemma 6 and (32), one can verify that

$$\Delta X \in \mathrm{app}(\bar{y}, \bar{Z}). \tag{33}$$

Then by (31), (33) implies that

$$-\Gamma_{\bar{X}}(-\bar{Z}, \Delta X) > 0. \tag{34}$$

On the other hand, by (32), it holds that $\langle \Delta X, \Delta Z \rangle = \langle \Delta X, -\mathcal{A}^*\Delta y \rangle = \langle \mathcal{A}\Delta X, -\Delta y \rangle = 0$, which together with Lemma 12, yields that

$$\Gamma_{\bar{X}}(-\bar{Z}, \Delta X) \geq \langle \Delta X, \Delta Z \rangle = 0. \tag{35}$$

However, (35) contradicts to (34). Thus, $\Delta X = 0$ and $V(0, -\Delta Z) = 0$.

We next prove $\Delta y = 0$ and $\Delta Z = 0$. From Lemma 6, $V(0, -\Delta Z) = 0$ implies that that

$$P_\alpha^T \Delta Z P_\alpha = 0, \quad P_\alpha^T \Delta Z P_\beta = 0, \quad P_\alpha^T \Delta Z P_\gamma = 0.$$

On the other hand, from (32), we get $\mathcal{A}^*\Delta y + \Delta Z = 0$. Since the primal constraint nondegeneracy condition (9) holds at $\bar{X}$, there exist $X \in \mathbb{S}^n$ and $Z \in \mathrm{lin}(\mathcal{T}_{\mathbb{S}^n_+}(\bar{X}))$ such that $\mathcal{A}X = \Delta y$, and $X + Z = \Delta Z$. As a consequence, it holds that

$$\begin{aligned} \langle \Delta y, \Delta y \rangle + \langle \Delta Z, \Delta Z \rangle &= \langle \mathcal{A}X, \Delta y \rangle + \langle X + Z, \Delta Z \rangle \\ &= \langle \mathcal{A}X, \Delta y \rangle + \big\langle X, -\mathcal{A}^*(\Delta y) \big\rangle + \langle Z, \Delta Z \rangle \\ &= \langle Z, \Delta Z \rangle = \Big\langle P^T Z P, P^T \Delta Z P \Big\rangle = 0, \end{aligned}$$

where we have used the fact that $P_\alpha^T \Delta Z P_\alpha = 0$, $P_\alpha^T \Delta Z P_\beta = 0$, $P_\alpha^T \Delta Z P_\gamma = 0$, $Z \in \mathrm{lin}(\mathcal{T}_{\mathbb{S}^n_+}(\bar{X}))$. Thus, $\Delta y = 0$, $\Delta Z = 0$ and hence U is nonsingular. Thus, the proof is completed. □

Typically, under certain regularity conditions, one is able to establish the local fast convergence rate of Newton-type methods. Indeed, we show in the next theorem that Algorithm 1 admits a superlinear convergence rate under the primal and dual constraint nondegenerate conditions.

Theorem 14 *Let $(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z})$ be an accumulation point of the infinite sequence $\{(\epsilon^k, X^k, y^k, Z^k)\}$ generated by Algorithm 1. Suppose that the primal constraint nondegeneracy condition holds at $\bar{X}$, and the dual constraint nondegeneracy condition holds at $(\bar{y}, \bar{Z})$. Then, the whole sequence $\{(\epsilon^k, X^k, y^k, Z^k)\}$ converges to $(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z})$ superlinearly, i.e.,*

$$\begin{aligned}&\left\|(\epsilon^{k+1} - \bar{\epsilon}, X^{k+1} - \bar{X}, y^{k+1} - \bar{y}, Z^{k+1} - \bar{Z})\right\| \\ &= O\left(\left\|(\epsilon^k - \bar{\epsilon}, X^k - \bar{X}, y^k - \bar{y}, Z^k - \bar{Z})\right\|^{1+\tau}\right).\end{aligned}$$

Proof By Theorem 11, we see that $\widehat{\mathcal{E}}(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z}) = 0$ and in particular, $\bar{\epsilon} = 0$. Then by Proposition 13, the primal and dual constraint nondegeneracy conditions imply that every element of $\partial\widehat{\mathcal{E}}(0, \bar{X}, \bar{y}, \bar{Z})$ (hence, of $\partial_B\widehat{\mathcal{E}}(0, \bar{X}, \bar{y}, \bar{Z})$) is nonsingular. As a consequence (see e.g., [26, Proposition 3.1]), for all k sufficiently large, it holds that

$$\left\|\widehat{\mathcal{E}}'(\epsilon^k, X^k, y^k, Z^k)^{-1}\right\| = O(1). \tag{36}$$

For simplicity, in this proof, we denote $\bar{w} := (\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z})^T$, $w^k := (\epsilon^k, X^k, y^k, Z^k)^T$, $\Delta w^k := (\Delta\epsilon^k, \Delta X^k, \Delta y^k, \Delta Z^k)^T$, $\widehat{\mathcal{E}}_k := \widehat{\mathcal{E}}(\epsilon^k, X^k, y^k, Z^k)$, $\widehat{\mathcal{E}}_* := \widehat{\mathcal{E}}(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z})$ and $\mathcal{J}_k := \widehat{\mathcal{E}}'(\epsilon^k, X^k, y^k, Z^k)$ for $k \geq 0$. Then, we can verify that

$$\begin{aligned}\left\|w^k + \Delta w^k - \bar{w}\right\| &= \left\|w^k + \mathcal{J}_k^{-1}\left(\begin{pmatrix}\zeta_k\hat{\epsilon}\\ \mathcal{R}_k\end{pmatrix} - \widehat{\mathcal{E}}_k\right) - \bar{w}\right\| \\ &= \left\|-\mathcal{J}_k^{-1}\left(\widehat{\mathcal{E}}_k - \mathcal{J}_k(w^k - \bar{w}) - \begin{pmatrix}\zeta_k\hat{\epsilon}\\ \mathcal{R}_k\end{pmatrix}\right)\right\| \\ &= O\left(\left\|\widehat{\mathcal{E}}_k - \mathcal{J}_k(w^k - \bar{w})\right\|\right) + O\left(\left\|\widehat{\mathcal{E}}_k\right\|^{1+\tau}\right) + O(\|\mathcal{R}_k\|). \end{aligned} \tag{37}$$

Since $\widehat{\mathcal{E}}$ is locally Lipschitz continuous at $(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z})$, for all k sufficiently large, it holds that

$$\left\|\widehat{\mathcal{E}}_k\right\| = \left\|\widehat{\mathcal{E}}_k - \widehat{\mathcal{E}}_*\right\| = O\left(\left\|w^k - \bar{w}\right\|\right). \tag{38}$$

Moreover, since $\mathcal{E}'_\epsilon$ is bounded near $(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z})$, it holds that

$$\begin{aligned}\|\mathcal{R}_k\| &\leq \eta_k \left\|\mathcal{E}(\epsilon^k, X^k, y^k, Z^k) + \mathcal{E}'_\epsilon(\epsilon^k, X^k, y^k, Z^k)\Delta\epsilon^k\right\| \\ &\leq O\left(\left\|\widehat{\mathcal{E}}_k\right\|^\tau\right)\left(\left\|\mathcal{E}(\epsilon^k, X^k, y^k, Z^k)\right\| + O\left(\left|\Delta\epsilon^k\right|\right)\right) \\ &\leq O\left(\left\|\widehat{\mathcal{E}}_k\right\|^\tau\right)\left(\left\|\mathcal{E}(\epsilon^k, X^k, y^k, Z^k)\right\| + O\left(\left|-\epsilon^k + \zeta_k\hat{\epsilon}\right|\right)\right) \\ &\leq O\left(\left\|\widehat{\mathcal{E}}_k\right\|^{1+\tau}\right) = O\left(\left\|\widehat{\mathcal{E}}_k - \widehat{\mathcal{E}}_*\right\|^{1+\tau}\right) = O\left(\left\|w^k - \bar{w}\right\|^{1+\tau}\right).\end{aligned} \tag{39}$$

Since Φ is strongly semismooth everywhere (see Proposition 4), $\widehat{\mathcal{E}}$ is also strongly semismooth at $(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z})$. Thus, for k sufficiently large, it holds that

$$\left\|\widehat{\mathcal{E}}_k - \mathcal{J}_k(w^k - \bar{w})\right\| = O\left(\left\|w^k - \bar{w}\right\|^2\right),$$

which together with (37)–(39) implies that

$$\left\|w^k + \Delta w^k - \bar{w}\right\| = O\left(\left\|w^k - \bar{w}\right\|^{1+\tau}\right). \tag{40}$$

Now by using the strong semismoothness of $\widehat{\mathcal{E}}$ again and (36), we can show that for k sufficiently large,

$$\left\|w^k - \bar{w}\right\| = O\left(\left\|\widehat{\mathcal{E}}_k\right\|\right). \tag{41}$$

Combining (40)–(41) and the fact that $\widehat{\mathcal{E}}$ (hence ψ) is locally Lipschitz continuous at $(\bar{\epsilon}, \bar{X}, \bar{y}, \bar{Z})$, we get for k sufficiently large that

$$\begin{aligned}
&\psi(\epsilon^k + \Delta\epsilon^k, X^k + \Delta X^k, y^k + \Delta y^k, Z^k + \Delta Z^k) \\
&= \left\|\widehat{\mathcal{E}}(\epsilon^k + \Delta\epsilon^k, X^k + \Delta X^k, y^k + \Delta y^k, W^k + \Delta W^k) - \widehat{\mathcal{E}}_*\right\|^2 \\
&= O\left(\left\|w^k + \Delta w^k - \bar{w}\right\|^2\right) = O\left(\left\|w^k - \bar{w}\right\|^{2(1+\tau)}\right) = O\left(\left\|\widehat{\mathcal{E}}_k\right\|^{2(1+\tau)}\right) \\
&= O\left(\left\|\psi(\epsilon^k, X^k, y^k, Z^k)\right\|^{1+\tau}\right) = o\left(\left\|\psi(\epsilon^k, X^k, y^k, Z^k)\right\|\right).
\end{aligned}$$

This shows that, for k sufficiently large, $w^{k+1} = w^k + \Delta w^k$, i.e., the unit step size is eventually accepted. Therefore, the proof is completed. □

4.3 Numerical implementation

Given $\nu > 0$, one can check that the condition $X - \Pi_{\mathbb{S}^n_+}(X - Z) = 0$ is equivalent to the condition $X - \Pi_{\mathbb{S}^n_+}(X - \nu Z) = 0$. Thus in our implementation, we in fact solve the following nonlinear system (with a slight abuse of the notation $\mathcal{E}$):

$$\mathcal{E}(\epsilon, X, y, Z) = \begin{pmatrix} \mathcal{A}X + \kappa_p\,|\epsilon|\,y - b \\ -\mathcal{A}^* y - Z + C \\ (1 + \kappa_c\,|\epsilon|)X - \Phi(\epsilon, X - \nu Z) \end{pmatrix}, \quad \forall (\epsilon, X, y, Z) \in \mathbb{R} \times \mathbb{X}.$$

Our numerical experience shows that introducing the parameter ν is important as it balances the norms of X and Z, and thus improving the performance of the algorithm. However, the convergence properties established in the previous sections are not affected.

Note that one of the key computational tasks in Algorithm 1 is to solve a system of linear equations for computing the search direction at each iteration. Here, we briefly explain how we solve the linear system of the following form:

$$\begin{pmatrix} \mathcal{A} & \mu_p I_m & 0 \\ 0 & -\mathcal{A}^* & -\mathcal{I} \\ (1 + \mu_c)\mathcal{I} - V & 0 & \nu V \end{pmatrix} \begin{pmatrix} \Delta X \\ \Delta y \\ \Delta Z \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \\ R_3 \end{pmatrix}, \tag{42}$$

where $\mu_p := \kappa_p |\epsilon| > 0$, $\mu_c := \kappa_c |\epsilon| > 0$, $V{:=}\Phi_2'(\epsilon, X - \nu Z)$, $\mathcal{I}$ is the identity mapping over $\mathbb{S}^n$ and $(R_1, R_2, R_3) \in \mathbb{R}^m \times \mathbb{S}^n \times \mathbb{S}^n$ is the right-hand-side vector constructed from the current iterate $(X, y, Z) \in \mathbb{X}$. From the last two equations of (42), we have

$$\Delta Z = -\mathcal{A}^* \Delta y - R2, \quad \Delta X = [(1+\mu_c)\mathcal{I} - V]^{-1} (R_3 - \nu V \Delta Z),$$

which implies that $\Delta X = [(1+\mu_c)\mathcal{I} - V]^{-1} (R_3 + \nu V \mathcal{A}^* \Delta y + \nu V R_2)$. Substituting this equality into the first equation in (42), we get

$$\mathcal{A} [(1+\mu_c)\mathcal{I} - V]^{-1} (R_3 + \nu V \mathcal{A}^* \Delta y + \nu V R_2) + \mu_p \Delta y = R_1,$$

which leads to the following smaller $m \times m$ symmetric positive definite system:

$$\begin{aligned} &\left(\mu_p I_m + \nu \mathcal{A} [(1+\mu_c)\mathcal{I} - V]^{-1} V \mathcal{A}^*\right) \Delta y \\ = {}& R_1 - \mathcal{A} [(1+\mu_c)\mathcal{I} - V]^{-1} (\nu V R_2 + R_3). \end{aligned} \tag{43}$$

We then apply the preconditioned conjugate gradient (PCG) method to solve the last linear system to get Δy. After obtaining Δy, we can compute ΔX and ΔZ in terms of Δy. In our experiments, we set κ_p to be a small number, say $\kappa_p = 10^{-10}$, while κ_c should be larger and may be changed for different classes of problems.

Next, we shall see how to evaluate the matrix-vector products involving $[(1+\mu_c)\mathcal{I} - V]^{-1} V$ which are needed when solving (43). Suppose that (ϵ, X, y, Z) with $\epsilon > 0$ is given and that $W := X - Z$ has the spectral decomposition in (7). Then, the linear mapping $V : \mathbb{S}^n \to \mathbb{S}^n$ takes the following form: $V(H) = P\left[\Omega(\epsilon, d) \circ \left(P^T H P\right)\right] P^T$, $\forall H \in \mathbb{S}^n$, where $\Omega(\epsilon, d) \in \mathbb{S}^n$ is defined in (13) and $[\Omega(\epsilon, d)]_{ij} \in [0, 1]$, for $1 \le i, j \le n$. Consider the following three index sets: $\alpha := \{i \,:\, d_i \ge \epsilon\}$, $\beta := \{i \,:\, 0 < d_i < \epsilon\}$, and $\gamma := \{i \,:\, d_i \le 0\}$. Then, we can simply write $\Omega(\epsilon, d)$ as

$$\Omega(\epsilon, d) = \begin{pmatrix} E_{\alpha\alpha} & \Omega_{\alpha\beta} & \Omega_{\alpha\gamma} \\ \Omega_{\alpha\beta}^T & \Omega_{\beta\beta} & 0 \\ \Omega_{\alpha\gamma}^T & 0 & 0 \end{pmatrix},$$

where $E_{\alpha\alpha} \in \mathbb{R}^{|\alpha| \times |\alpha|}$ is the matrix of all ones, and $\Omega_{\alpha\beta} \in \mathbb{R}^{|\alpha| \times |\beta|}$, $\Omega_{\beta\beta} \in \mathbb{R}^{|\beta| \times |\beta|}$ and $\Omega_{\alpha\gamma} \in \mathbb{R}^{|\alpha| \times |\gamma|}$ have all the entries belonging to the interval $(0, 1)$. We also partition the orthogonal matrix P as $P = \begin{pmatrix} P_\alpha & P_\beta & P_\gamma \end{pmatrix}$ accordingly. Define the matrix $\hat{\Omega} \in \mathbb{S}^n$ as $[\hat{\Omega}]_{ij} := [\Omega(\epsilon, d)]_{ij} / (1 + \mu_c - [\Omega(\epsilon, d)]_{ij})$ for $i, j = 1, \ldots, n$, which takes the following form

$$\hat{\Omega} = \begin{pmatrix} \frac{1}{\mu_c} E_{\alpha\alpha} & \hat{\Omega}_{\alpha\beta} & \hat{\Omega}_{\alpha\gamma} \\ \hat{\Omega}_{\alpha\beta}^T & \hat{\Omega}_{\beta\beta} & 0 \\ \hat{\Omega}_{\alpha\gamma}^T & 0 & 0 \end{pmatrix},$$

Then one can check that $[(1+\mu_c)\mathcal{I} - V]^{-1} V(H) = P\left[\hat{\Omega} \circ \left(P^T H P\right)\right] P^T$, $\forall H \in \mathbb{S}^n$. If $|\alpha| + |\beta| \ll n$, one may use the following scheme to compute

$$
\begin{aligned}
& [(1+\mu_c)\mathcal{I} - V]^{-1} V(H) \\
= & \frac{1}{\mu_c} P_\alpha \left(P_\alpha^T H P_\alpha\right) P_\alpha^T + P_\beta \left(\hat{\Omega}_{\beta\beta} \circ (P_\beta^T H P_\beta)\right) P_\beta^T \\
& + P_\alpha \left(\hat{\Omega}_{\alpha\beta} \circ (P_\alpha^T H P_\beta)\right) P_\beta^T + \left(P_\alpha \left(\hat{\Omega}_{\alpha\beta} \circ (P_\alpha^T H P_\beta)\right) P_\beta^T\right)^T \\
& + P_\alpha \left(\hat{\Omega}_{\alpha\gamma} \circ (P_\alpha^T H P_\gamma)\right) P_\gamma^T + \left(P_\alpha \left(\hat{\Omega}_{\alpha\gamma} \circ (P_\alpha^T H P_\gamma)\right) P_\gamma^T\right)^T.
\end{aligned}
$$

On the other hand, if $n - |\alpha| \ll n$, one may consider using the following scheme to compute

$$
[(1+\mu_c)\mathcal{I} - V]^{-1} V(H) = \frac{1}{\mu_c} H - P\left[\tilde{\Omega} \circ \left(P^T H P\right)\right] P^T,
$$

where $\tilde{\Omega} := \frac{1}{\mu_c} E - \hat{\Omega}$ would have more zero entries than $\hat{\Omega}$. As a consequence, since the Huber function maps any non-positive number to zero, we can exploit the sparsity structure in $\hat{\Omega}$ or $\tilde{\Omega}$ to cut down the computational cost. However, if the CHKS-smoothing function is used, we will get dense matrices and the aforementioned sparsity structure will be lost. This also explains why we choose the Huber function instead of the CHKS function to perform the smoothing of the projection operator.

We observe from our numerical tests that when the dual iterates (y^k, Z^k) does not make a significant progress, it is helpful for us to project the primal iterate X^k onto the affine subspace $\mathcal{H}_k := \left\{X \in S^n : \mathcal{A}(X) = b, \langle C, X\rangle = \left\langle b, y^k \right\rangle\right\}$. To perform such a projection operation, we only need to performance a Cholesky factorization for a certain operator (depending only on $\mathcal{A}$ and C) once at the beginning of the Algorithm 1. However, in the case when the factorization fails (i.e., the operator is no positive definite) or m is too large, we will not perform such projections.

5 Numerical experiments

To evaluate the practical performance of Algorithm 1 described in the last section, we conduct numerical experiments to solve various classes of SDPs that are commonly tested in the literature. We will compare our proposed algorithm with the general purposed solver `SDPNAL+` [27].

5.1 Experimental settings and implementation

Similar to [27], the following relative KKT residues are used as the termination criteria of our algorithm:

$$\eta_p := \frac{\|\mathcal{A}X - b\|}{1 + \|b\|}, \quad \eta_d := \frac{\|\mathcal{A}^* y + Z - C\|}{1 + \|C\|}, \quad \eta_c := \frac{\left\|X - \Pi_{\mathbb{S}^n_+}(X - Z)\right\|}{1 + \|X\| + \|Z\|}.$$

Particularly, we terminate the algorithm as long as $\eta_{kkt} := \max\{\eta_p, \eta_d, \eta_c\} \leq$ `tol` where `tol` > 0 is a given tolerance. Moreover, denote the maximum number of iterations of Algorithm 1 as `maxiter`. We also stop the algorithm when the iteration count reaches this number. In our experiments, we set `tol` $= 10^{-6}$ and `maxiter` $= 50$. Note that the same stopping tolerance is used for `SDPNAL+`.

For more efficiency, we apply a certain first-order method to generate a starting point (X^0, y^0, Z^0) to warmstart Algorithm 1. Our choice is the routine based on a semi-proximal ADMM method [27], namely `admmplus`, which is included in the package of `SDPNAL+`. The stopping tolerance (in terms of the maximal relative KKT residual, i.e., η_{kkt}) and the maximum number of iterations for `admmplus` are denoted by $\texttt{tol}_0$ and $\texttt{maxiter}_0$, respectively. In our experiments, we set $\texttt{maxiter}_0 = 2000$. We also notice that the performance of Algorithm 1 depends sensitively on the choice of $\texttt{tol}_0$. Hence, we set the value of $\texttt{tol}_0$ differently for different classes of SDPs for more efficiency.

For the parameters required in Algorithm 1, we set $r = \hat{r} = 0.6$, $\eta = \tau = 0.2$, $\rho = 0.5$ and $\sigma = 10^{-8}$. However, since the initial smoothing parameter $\hat{\epsilon} > 0$ affects the performance of Algorithm 1, it is chosen differently for different classes of problems.

Finally, we should mention that our algorithm is implemented in MATLAB (R2021b) and all the numerical experiments are conducted on a Linux PC having Intel Xeon E5-2680 (v3) cores.

5.2 Testing examples

Example 1 (MaxCut-SDP) Consider the SDP relaxation [56] of the maximum cut problem of a graph which takes the form of

$$\min_{X \in \mathbb{S}^n} \ \langle C, X \rangle \quad \text{s.t.} \quad \operatorname{diag}(X) = e, \ X \in \mathbb{S}^n_+,$$

where $e \in \mathbb{R}^n$ denotes the vector of all ones and $C := -(\operatorname{diag}(We_n) - W)/4$ with W being the weighted adjacency matrix of the underlying graph. The above SDP problem has been a commonly used test problem for evaluating the performance of different SDP solvers. A popular data set for this problem is the GSET collection of randomly generated graphs. The GSET is available at: https://web.stanford.edu/~yyye/yyye/Gset/.

Example 2 (Theta-SDP) Let $G = (V, E)$ be a graph with n nodes V and edges E. A stable set of G is a subset of V containing no adjacent nodes. The stability number

$\alpha(G)$ is the cardinality of a maximal stable set of G. More precisely, it holds that

$$\alpha(G) := \max_{x \in \mathbb{R}^n} \left\{ e^T x : x_i x_j = 0, (i,j) \in E, x \in \{0,1\}^n \right\}.$$

However, computing $\alpha(G)$ is difficult. A notable lower bound of $\alpha(G)$ is called the Lovász theta number [57] which is defined as

$$\theta(G) := \max_{X \in \mathbb{S}^n} \left\langle ee^T, X \right\rangle \quad \text{s.t.} \quad \left\langle E_{ij}, X \right\rangle = 0, (i,j) \in E, \ \langle I, X \rangle = 1, \ X \in \mathbb{S}^n_+,$$

where $E_{ij} = e_i e_j^T + e_j e_i^T \in \mathbb{S}^n$. For our experiments, the test data sets are chosen from [18, Section 6.3].

Example 3 (BIQ-SDP) The NP-hard binary integer quadratic programming (BIQ) problem has the following form:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \langle x, Qx \rangle + \langle c, x \rangle \quad \text{s.t.} \quad x \in \{0,1\}^n.$$

It has many important practical applications due to its modeling power in representing the structure of graphs; see for example [58] for a recent survey. Under some mild conditions, Burer showed in [59] the BIQ problem can be reformulated as a completely positive conic programming problem. Since the completely positive cone is numerically intractable, we consider its SDP relaxation:

$$\min_{X \in \mathbb{S}^n, x \in \mathbb{R}^n, \alpha \in \mathbb{R}} \frac{1}{2} \langle Q, X \rangle + \langle c, x \rangle \quad \text{s.t.} \quad \operatorname{diag}(X) = x, \ \alpha = 1, \begin{pmatrix} X & x^T \\ x & \alpha \end{pmatrix} \in \mathbb{S}^{n+1}_+.$$

In our experiments, the matrix Q and the vector c is obtained from the BIQMAC library [60].

Example 4 (Clustering-SDP) The clustering problem is to group a set of data points into several clusters. The problem is in general NP-hard. According to [61], we can consider the following SDP relaxation of the clustering problem:

$$\min_{X \in \mathbb{S}^n} - \langle W, X \rangle \quad \text{s.t.} \quad X e_n = e_n, \ \langle I, X \rangle = k, \ X \in \mathbb{S}^n_+,$$

where W is the affinity matrix whose entries represent the pairwise similarities of the objects in the input data set. In our experiments, the test data sets are obtained from the UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets.html. For more information on how we generate the test SDP problems from the raw input data sets, readers are refered to [27].

Example 5 (Tensor-SDP) Consider the following SDP relaxation of a rank-1 tensor approximation problem [62]:

$$\max_{y \in \mathbb{R}^{\mathbb{N}^n_m}} \langle f, y \rangle \quad \text{s.t.} \quad M(y) \in \mathbb{S}^n_+, \ \langle g, y \rangle = 1,$$

where $\mathbb{N}^n_m := \{t = (t_1, \ldots, t_n) \in \mathbb{N}^n : t_1 + \cdots + t_n = m\}$ and $M(y)$ is a linear pencil in y. The dual of the above problem is given by

$$\min_{\gamma \in \mathbb{R}, X \in \mathbb{S}^n} \gamma \quad \text{s.t.} \quad \gamma g - f = M^*(X), \ X \in \mathbb{S}^n_+,$$

where M^* is the adjoint of M. The above problem can be transformed into a standard SDP [27].

5.3 Computational results

In this subsection, we present our numerical results for each class of SDP problems in detail. We use tables to summarize our results. In our tables, we report the sizes (i.e., m and n) of the tested problems, number of iterations (i.e., `it0`, `it1` and `it2`), total computational times (i.e., `cpu`) in seconds, relative KKT residues (i.e., η_p, η_d and η_c), and objective function values (i.e., $\langle C, X\rangle$). In particular, for the number of iterations, `it0` represents the number of iterations for `admmplus`, `it1` is the number of main iterations and `it2` is the total number of PCG iterations needed for solving the linear systems in (24). Under the column "solver", we use "a" and "b" to denote the solver `SDPNAL+` and Algorithm 1, respectively. As mentioned before, since the performance of our algorithm may depend sensitively on the choices of parameters $\texttt{tol}_0$, $\hat{\epsilon}$ and κ_c, we also report the values for these parameters in the captions of the presented tables. The computational results for Example 1–5 are presented in Table 1–5, respectively.

From these results in Table 1–4, we observe that our proposed algorithm performs better than `SDPNAL+`. In fact, Algorithm 1 is usually several times more efficient than `SDPNAL+`. These results have indeed shown that Algorithm 1 is efficient. In term of accuracy, we see that both algorithms are able to compute optimal solutions with $\eta < 10^{-6}$ for almost all the tested problems, which further shows that both algorithms are robust and suitable for solving SDPs relatively accurately. We observe that for the SDPs in Table 1–4, their optimal solutions are usually of very low-rank. Consequently, the primal constraint nondegenerate condition usually fails to hold, and `SDPNAL+` requires more computational effort to converge than Algorithm 1.

From Table 5, we see that Algorithm 1 is able to solve all the problems but perform much worse than `SDPNAL+` in terms of computational time. The reason is that for those SDPs, the ranks of the optimal solutions are high. In fact, we always observe that the rank is nearly $n-1$, where n denotes the matrix dimension. In such cases, the primal constraint nondegenerate condition usually holds, making `SDPNAL+` to have a very fast convergence rate. On the other hand, for these SDPs, because the dual constraint nondegeneracy condition typically fails to hold, Algorithm 1 would require computational effort than `SDPNAL+` to solve the problems.

Based on the relative performance of `SDPNAL+` and Algorithm 1 in Table 1–5, we may conclude that when the primal constraint nondegeneracy condition is likely to hold at the optimal solution, `SDPNAL+` would be more preferable. However, when such a prior information is not available, Algorithm 1 could be a good choice based on the promising numerical performance demonstrated in Table 1–4, even though the primal constraint nondegeneracy condition may not hold.

Note finally that for our algorithm, the errors η_p and η_d are usually very small while η_c is driven to a value slightly smaller than $\texttt{tol} = 10^{-6}$. On the contrary, `SDPNAL+` keeps η_c to be very small while progressively decreases η_p and η_d.

Table 1: Computational results for Example 1 with $\mathtt{tol}_0 = 5 \times 10^{-1}$, $\hat{\epsilon} = 5 \times 10^{-1}$, $\nu = 10^4$ and $\kappa_c = 1$.

problem size	solver	it it0, it1, it2	cpu [s]	kkt η_p, η_d, η_c	obj $\langle C, X\rangle$
g1	a	800, 13, 140	54.4	1.3e-07, 8.8e-08, 2.7e-13	-1.20831976e+04
800, 800	b	2, 14, 548	14.4	0.0e+00, 9.1e-17, 1.2e-07	-1.20831961e+04
g2	a	800, 12, 144	48.2	1.4e-07, 1.3e-07, 1.4e-12	-1.20894297e+04
800, 800	b	2, 14, 540	14.4	0.0e+00, 9.0e-17, 2.5e-07	-1.20894257e+04
g3	a	400, 15, 161	28.7	3.3e-07, 4.9e-07, 2.7e-12	-1.20843329e+04
800, 800	b	2, 14, 539	13.8	0.0e+00, 9.5e-17, 3.0e-07	-1.20843279e+04
g4	a	400, 14, 176	29.0	3.2e-07, 8.5e-07, 6.5e-12	-1.21114510e+04
800, 800	b	2, 14, 472	13.4	0.0e+00, 1.0e-16, 9.0e-07	-1.21114346e+04
g5	a	909, 11, 131	49.6	5.9e-15, 1.0e-06, 3.3e-07	-1.20998869e+04
800, 800	b	2, 14, 532	13.5	0.0e+00, 9.7e-17, 1.5e-07	-1.20998846e+04
g22	a	821, 18, 147	403.6	9.5e-07, 3.5e-07, 6.1e-12	-1.41359474e+04
2000, 2000	b	2, 15, 677	164.1	0.0e+00, 1.0e-16, 2.0e-07	-1.41359397e+04
g23	a	821, 18, 148	395.5	8.0e-07, 2.6e-07, 6.8e-12	-1.41421096e+04
2000, 2000	b	2, 15, 722	181.0	0.0e+00, 1.0e-16, 1.5e-07	-1.41421056e+04
g24	a	821, 18, 171	399.5	1.6e-07, 9.7e-07, 1.5e-13	-1.41408558e+04
2000, 2000	b	2, 15, 637	169.9	0.0e+00, 1.1e-16, 1.6e-07	-1.41408511e+04
g25	a	821, 18, 156	402.7	2.4e-08, 3.8e-07, 5.7e-12	-1.41442454e+04
2000, 2000	b	2, 15, 613	165.6	0.0e+00, 1.0e-16, 1.4e-07	-1.41442426e+04
g26	a	821, 18, 152	398.6	1.7e-07, 9.3e-07, 1.1e-11	-1.41328706e+04
2000, 2000	b	2, 15, 670	177.3	0.0e+00, 8.7e-17, 1.7e-07	-1.41328660e+04
g43	a	400, 16, 197	43.9	6.4e-07, 2.5e-07, 3.7e-12	-7.03222176e+03
1000, 1000	b	2, 14, 612	23.3	0.0e+00, 9.8e-17, 5.1e-07	-7.03221524e+03
g44	a	400, 14, 187	44.7	9.3e-07, 9.9e-07, 3.6e-12	-7.02788487e+03
1000, 1000	b	2, 14, 614	23.0	0.0e+00, 1.1e-16, 2.7e-07	-7.02788112e+03
g45	a	555, 14, 200	59.0	9.5e-07, 8.5e-07, 9.3e-07	-7.02478572e+03
1000, 1000	b	2, 14, 626	23.1	0.0e+00, 9.7e-17, 2.0e-07	-7.02477888e+03
g46	a	400, 16, 187	44.8	7.1e-07, 2.6e-07, 1.1e-12	-7.02993319e+03
1000, 1000	b	2, 14, 558	21.4	0.0e+00, 9.1e-17, 7.5e-07	-7.02992466e+03
g47	a	703, 14, 178	69.2	8.0e-07, 1.0e-06, 4.0e-07	-7.03665811e+03
1000, 1000	b	2, 14, 609	22.8	0.0e+00, 1.0e-16, 3.3e-07	-7.03665816e+03
g48	a	3451, 22, 60	10002.0	1.3e-16, 8.4e-03, 2.5e-02	-6.05098926e+03
3000, 3000	b	3, 16, 45	205.7	0.0e+00, 1.3e-16, 4.2e-08	-5.99999689e+03
g49	a	6192, 29, 88	10001.8	1.1e-15, 1.1e-05, 2.3e-07	-5.99989318e+03
3000, 3000	b	3, 16, 47	204.5	0.0e+00, 1.3e-16, 1.0e-07	-5.99999943e+03
g50	a	3128, 15, 62	10003.9	6.6e-14, 1.3e-05, 6.9e-06	-5.98757867e+03
3000, 3000	b	3, 15, 45	202.3	0.0e+00, 1.3e-16, 1.4e-07	-5.98816920e+03

Table 2: Computational results for Example 2 with $\mathtt{tol}_0 = 1 \times 10^{-3}$, $\hat{\epsilon} = 1 \times 10^{-3}$, $\nu = 10^2$ and $\kappa_c = 1 \times 10^3$.

problem size	solver	it it0, it1, it2	cpu [s]	kkt η_p, η_d, η_c	obj $\langle C, X\rangle$
theta4	a	451, 9, 351	3.3	3.9e-08, 1.6e-07, 4.0e-16	-5.03212233e+01
200, 1949	b	172, 3, 210	1.2	1.4e-19, 2.4e-08, 2.1e-07	-5.03211624e+01
theta42	a	176, 4, 496	1.5	1.2e-07, 9.1e-08, 4.0e-15	-2.39317078e+01
200, 5986	b	77, 3, 210	0.8	1.4e-16, 2.3e-08, 4.0e-07	-2.39317215e+01
theta6	a	286, 4, 152	3.5	6.3e-07, 3.9e-07, 5.1e-15	-6.34771580e+01
300, 4375	b	187, 3, 210	2.6	1.3e-16, 2.1e-08, 1.3e-07	-6.34770467e+01
theta62	a	180, 3, 365	2.7	3.6e-07, 8.0e-07, 1.6e-15	-2.96412514e+01
300, 13390	b	75, 4, 310	2.0	1.3e-16, 1.3e-08, 1.8e-07	-2.96412833e+01
theta8	a	299, 3, 214	6.1	2.2e-07, 7.7e-07, 4.0e-15	-7.39535658e+01
400, 7905	b	200, 3, 210	4.6	5.3e-16, 1.9e-08, 1.4e-07	-7.39535163e+01
theta82	a	179, 3, 352	4.8	1.0e-07, 2.1e-07, 2.4e-16	-3.43668939e+01
400, 23872	b	82, 3, 210	3.1	5.8e-16, 2.2e-08, 1.2e-07	-3.43668925e+01
theta83	a	265, 4, 739	9.2	1.2e-07, 7.4e-08, 1.2e-15	-2.03018925e+01
400, 39862	b	65, 3, 210	3.5	2.4e-16, 2.7e-08, 7.4e-07	-2.03019697e+01
theta10	a	200, 4, 595	8.1	1.5e-07, 5.1e-07, 1.7e-15	-8.38059660e+01
500, 12470	b	202, 3, 210	6.7	7.0e-16, 2.6e-08, 1.1e-07	-8.38059211e+01
theta102	a	281, 4, 522	11.3	1.4e-07, 6.2e-08, 5.5e-15	-3.83905444e+01
500, 37467	b	78, 5, 410	7.0	1.6e-16, 3.0e-07, 8.0e-07	-3.83904989e+01
theta103	a	165, 3, 510	9.0	6.6e-07, 1.7e-07, 1.7e-15	-2.25285599e+01
500, 62516	b	65, 3, 210	5.6	2.2e-16, 2.7e-08, 8.2e-07	-2.25287026e+01
theta104	a	166, 3, 651	9.9	9.1e-07, 2.6e-07, 1.6e-15	-1.33361353e+01
500, 87245	b	66, 5, 410	9.5	2.2e-16, 4.1e-10, 2.5e-07	-1.33363826e+01
theta12	a	400, 8, 428	19.1	1.7e-07, 5.0e-07, 2.5e-15	-9.28016868e+01
600, 17979	b	193, 4, 310	10.9	5.3e-16, 1.2e-08, 2.5e-07	-9.28015772e+01
theta123	a	166, 3, 455	13.7	1.6e-07, 6.1e-08, 2.0e-15	-2.46686518e+01
600, 90020	b	66, 4, 310	11.5	2.4e-16, 2.0e-08, 8.6e-07	-2.46688660e+01
theta162	a	169, 3, 709	32.3	8.8e-08, 3.7e-08, 1.6e-16	-3.70097368e+01
800, 127600	b	68, 4, 310	19.7	4.1e-19, 2.2e-08, 9.4e-07	-3.70100248e+01
MANN-a27	a	300, 4, 31	4.5	6.4e-07, 5.0e-07, 1.5e-14	-1.32760995e+02
378, 703	b	165, 7, 55	3.3	3.3e-16, 2.6e-10, 7.2e-07	-1.32761828e+02
johnson8-4-4	a	123, 2, 4	0.2	5.0e-08, 4.9e-09, 1.1e-15	-1.39999979e+01
70, 561	b	85, 3, 9	0.1	1.5e-23, 2.2e-08, 2.3e-07	-1.39999894e+01
johnson16-2-4	a	117, 0, 0	0.3	4.5e-07, 6.9e-09, 5.0e-07	-7.99993991e+00
120, 1681	b	55, 3, 9	0.1	2.0e-24, 2.0e-08, 3.8e-08	-7.99999922e+00
san200-0.7-1	a	300, 5, 669	2.2	1.3e-07, 3.7e-07, 1.1e-15	-2.99999931e+01
200, 5971	b	1000, 4, 22	3.3	7.0e-17, 2.5e-09, 8.5e-08	-2.99999443e+01
sanr200-0.7	a	175, 4, 560	1.6	7.3e-08, 2.3e-07, 2.4e-16	-2.38361575e+01
200, 6033	b	76, 3, 210	0.8	7.0e-17, 2.5e-08, 2.8e-07	-2.38361666e+01

Table 2 continued from previous page

problem	solver	it	cpu	kkt	obj
size		it0, it1, it2	[s]	η_p, η_d, η_c	$\langle C, X \rangle$
c-fat200-1	a	229, 2, 221	1.4	5.0e-08, 1.8e-07, 6.8e-16	-1.19999991e+01
200, 18367	b	141, 4, 310	1.2	4.9e-16, 7.4e-10, 2.2e-07	-1.19999906e+01
hamming-6-4	a	67, 2, 4	0.1	9.7e-11, 1.2e-11, 1.8e-16	-5.33333333e+00
64, 1313	b	35, 3, 12	0.1	9.3e-24, 8.5e-09, 2.8e-07	-5.33332949e+00
hamming-8-4	a	172, 1, 2	1.6	1.0e-10, 6.1e-07, 2.7e-16	-1.60000000e+01
256, 11777	b	77, 3, 15	0.8	3.9e-17, 2.5e-08, 6.3e-08	-1.59999969e+01
hamming-9-8	a	200, 3, 14	7.3	1.8e-07, 7.7e-08, 1.2e-14	-2.24000081e+02
512, 2305	b	139, 4, 15	4.0	1.6e-16, 2.4e-07, 4.8e-08	-2.23999937e+02
hamming-10-2	a	200, 3, 17	34.5	2.1e-07, 2.9e-08, 3.1e-15	-1.02400016e+02
1024, 23041	b	23, 12, 161	21.3	6.5e-16, 2.6e-16, 8.1e-08	-1.02399963e+02
hamming-7-5-6	a	300, 2, 6	0.5	3.2e-08, 7.0e-07, 4.0e-15	-4.26666634e+01
128, 1793	b	294, 3, 11	0.6	5.7e-17, 2.6e-08, 2.7e-07	-4.26666131e+01
hamming-8-3-4	a	117, 2, 6	0.8	5.8e-08, 3.8e-07, 4.5e-16	-2.56000038e+01
256, 16129	b	93, 3, 11	0.7	1.2e-16, 2.6e-08, 1.1e-07	-2.55999900e+01
hamming-9-5-6	a	200, 3, 13	5.6	1.0e-07, 3.5e-07, 9.0e-15	-8.53333428e+01
512, 53761	b	53, 4, 16	2.2	1.1e-16, 8.8e-08, 8.5e-08	-8.53332673e+01
brock200-1	a	187, 4, 283	1.3	7.7e-07, 2.7e-07, 2.5e-15	-2.74566741e+01
200, 5067	b	88, 3, 210	0.9	1.4e-16, 2.7e-08, 2.4e-07	-2.74566332e+01
brock200-4	a	273, 4, 666	2.2	2.4e-07, 2.3e-07, 3.0e-15	-2.12934762e+01
200, 6812	b	72, 3, 210	0.8	1.4e-16, 2.6e-08, 7.2e-07	-2.12934991e+01
brock400-1	a	194, 3, 347	5.0	1.6e-07, 3.9e-07, 1.4e-15	-3.97018998e+01
400, 20078	b	95, 3, 210	3.2	3.9e-16, 2.0e-08, 1.3e-07	-3.97018904e+01
keller4	a	172, 3, 420	0.9	2.4e-07, 6.4e-07, 4.1e-16	-1.40122363e+01
171, 5101	b	76, 4, 310	0.7	3.9e-16, 2.6e-08, 5.2e-07	-1.40122928e+01
p-hat300-1	a	191, 6, 6728	18.5	8.7e-07, 5.9e-07, 3.1e-16	-1.00679872e+01
300, 33918	b	92, 6, 510	3.8	2.1e-16, 3.9e-10, 7.5e-07	-1.00684728e+01
1dc.64	a	300, 5, 2559	1.6	6.3e-07, 7.7e-07, 1.8e-15	-1.00000371e+01
64, 544	b	258, 50, 4910	1.9	7.5e-21, 2.1e-16, 6.2e-06	-1.00030855e+01
1et.64	a	204, 3, 67	0.4	1.2e-07, 6.0e-07, 4.4e-16	-1.88000018e+01
64, 265	b	105, 3, 105	0.1	1.2e-16, 2.1e-08, 7.3e-07	-1.87999012e+01
1tc.64	a	224, 3, 332	0.5	6.5e-07, 9.9e-07, 1.6e-15	-1.99999867e+01
64, 193	b	126, 50, 4884	1.6	1.7e-16, 2.0e-12, 1.1e-06	-2.00000555e+01
1dc.128	a	2943, 60, 38588	24.6	9.8e-07, 9.8e-07, 2.6e-15	-1.68419006e+01
128, 1472	b	230, 50, 4910	4.0	5.7e-17, 2.2e-16, 1.2e-06	-1.68432284e+01
1et.128	a	273, 3, 69	0.8	5.4e-07, 8.0e-07, 1.4e-15	-2.92308924e+01
128, 673	b	173, 4, 296	0.6	7.4e-21, 2.5e-08, 6.5e-07	-2.92307806e+01
1tc.128	a	300, 5, 60	0.7	2.5e-07, 1.5e-07, 6.9e-16	-3.80000070e+01
128, 513	b	371, 4, 234	0.9	1.1e-16, 7.5e-10, 1.7e-07	-3.79998952e+01
1zc.128	a	200, 3, 30	0.6	8.6e-07, 4.1e-07, 2.0e-15	-2.06667050e+01
128, 1121	b	108, 3, 58	0.3	5.7e-17, 2.6e-08, 1.6e-07	-2.06666495e+01
2dc.128	a	363, 13, 12752	9.0	9.3e-07, 2.9e-07, 2.4e-17	5.24240971e+00
130, 3083	b	175, 50, 4910	5.4	4.5e-17, 1.4e-17, 7.9e-04	5.24263373e+00
1dc.256	a	1040, 10, 2709	9.4	8.5e-11, 4.7e-07, 1.8e-15	-3.00000000e+01
256, 3840	b	328, 10, 910	4.2	3.9e-17, 6.4e-15, 8.1e-07	-2.99999030e+01
1et.256	a	300, 7, 7288	11.7	6.1e-07, 1.5e-07, 1.1e-14	-5.51142364e+01
256, 1665	b	396, 4, 310	2.9	2.4e-16, 8.0e-10, 2.4e-07	-5.51162561e+01
1tc.256	a	522, 19, 13335	20.6	6.8e-07, 3.9e-07, 1.4e-07	-6.33997469e+01
256, 1313	b	424, 10, 910	4.9	3.9e-17, 2.2e-11, 9.1e-07	-6.34023351e+01
1zc.256	a	349, 5, 115	3.2	1.3e-07, 3.6e-07, 6.6e-15	-3.79999929e+01
256, 2817	b	140, 3, 90	1.0	3.9e-17, 1.7e-08, 7.8e-08	-3.79999825e+01
2dc.256	a	450, 18, 15933	35.8	8.8e-07, 8.4e-07, 4.9e-17	7.46181510e+00
256, 15713	b	173, 50, 4910	16.4	6.5e-17, 1.1e-16, 6.0e-04	7.47121358e+00
1dc.512	a	434, 29, 19405	101.1	9.6e-07, 5.6e-07, 3.1e-07	-5.30300531e+01
512, 9728	b	441, 4, 310	14.5	5.5e-17, 8.1e-10, 3.5e-07	-5.30334075e+01
1et.512	a	200, 10, 7244	39.3	4.9e-07, 2.9e-07, 6.5e-15	-1.04424017e+02
512, 4033	b	395, 5, 410	14.0	9.7e-19, 2.4e-11, 9.7e-08	-1.04434087e+02
1tc.512	a	610, 31, 20500	128.1	9.7e-07, 5.9e-07, 8.5e-08	-1.13400931e+02
512, 3265	b	638, 9, 810	23.5	2.2e-16, 1.5e-13, 7.3e-07	-1.13420135e+02
2dc.512	a	586, 27, 19171	175.1	8.7e-07, 7.4e-07, 2.3e-07	-1.17676507e+01
512, 54896	b	338, 17, 1610	29.0	3.3e-16, 1.5e-16, 1.0e-06	-1.17858696e+01
1zc.512	a	200, 4, 280	9.8	2.3e-07, 1.0e-07, 3.8e-15	-6.87499407e+01
512, 6913	b	230, 3, 165	8.6	1.1e-16, 1.8e-08, 4.0e-08	-6.87499812e+01
1dc.1024	a	500, 21, 9984	285.1	5.7e-07, 8.9e-07, 1.6e-15	-9.59854174e+01
1024, 24064	b	687, 4, 310	100.0	1.1e-16, 8.1e-10, 4.5e-07	-9.59889922e+01
1et.1024	a	467, 24, 16162	493.3	6.9e-07, 9.4e-07, 5.8e-08	-1.84227049e+02
1024, 9601	b	423, 4, 310	66.9	2.7e-16, 8.1e-10, 2.5e-07	-1.84260153e+02
1tc.1024	a	300, 26, 21575	505.3	9.2e-07, 7.7e-07, 6.4e-15	-2.06305132e+02
1024, 7937	b	743, 9, 810	128.2	7.6e-17, 4.6e-12, 1.3e-07	-2.06335860e+02
1zc.1024	a	200, 4, 325	39.5	6.8e-07, 4.0e-07, 9.5e-16	-1.28667295e+02
1024, 16641	b	401, 2, 110	59.0	4.6e-16, 5.6e-07, 6.8e-07	-1.28666054e+02
1dc.2048	a	200, 17, 11885	1785.4	7.1e-07, 4.5e-07, 6.1e-15	-1.74730215e+02
2048, 58368	b	1000, 5, 410	839.2	1.1e-16, 2.5e-11, 9.1e-08	-1.74745972e+02
1et.2048	a	400, 33, 23372	3829.5	4.3e-07, 8.4e-07, 1.4e-15	-3.42029537e+02
2048, 22529	b	974, 5, 410	857.1	8.0e-16, 6.1e-10, 1.3e-07	-3.42061413e+02
1tc.2048	a	1200, 31, 25455	4673.7	8.2e-07, 9.9e-07, 6.9e-15	-3.74646698e+02
2048, 18945	b	999, 10, 910	965.9	1.6e-16, 2.2e-11, 1.9e-07	-3.74699047e+02
1zc.2048	a	200, 6, 735	441.2	2.5e-08, 3.4e-08, 8.7e-15	-2.37400024e+02
2048, 39425	b	193, 5, 410	434.2	8.6e-16, 4.0e-08, 7.0e-07	-2.37402601e+02
1zc.4096	a	200, 7, 890	3124.7	2.3e-07, 8.4e-07, 7.6e-16	-4.49165510e+02
4096, 92161	b	237, 6, 510	3247.2	1.1e-15, 9.2e-07, 1.5e-07	-4.49173618e+02

Table 3: Computational results for Example 3 with $\texttt{tol}_0 = 5 \times 10^{-2}$, $\hat{\epsilon} = 1 \times 10^{-2}$, $\nu = 10^3$ and $\kappa_c = 1 \times 10^1$.

problem size	solver	it it0, it1, it2	cpu [s]	kkt η_p, η_d, η_c	obj $\langle C, X\rangle$
be150.3.1	a	900, 27, 5990	7.9	3.4e-07, 5.6e-08, 6.3e-11	-2.01751700e+04
151, 151	b	357, 7, 300	1.9	3.0e-16, 2.8e-12, 1.1e-07	-2.01749921e+04
be150.8.1	a	800, 29, 6426	8.4	5.8e-07, 5.4e-08, 1.7e-11	-2.96716584e+04
151, 151	b	294, 6, 210	1.3	3.0e-16, 6.4e-12, 9.5e-07	-2.96694342e+04
be200.3.1	a	1990, 24, 8572	23.1	1.0e-14, 1.0e-06, 7.2e-07	-2.82755919e+04
201, 201	b	383, 7, 303	2.9	3.6e-16, 3.2e-12, 2.3e-07	-2.82749700e+04
be200.8.1	a	2998, 21, 6129	24.4	1.3e-10, 3.8e-07, 2.0e-06	-5.17181377e+04
201, 201	b	189, 12, 626	1.9	3.5e-16, 4.7e-12, 4.9e-07	-5.17154802e+04
be250.1	a	2100, 30, 11801	38.5	1.0e-07, 3.2e-07, 1.9e-12	-2.55516534e+04
251, 251	b	180, 12, 567	2.5	3.5e-16, 1.3e-13, 3.0e-07	-2.55506173e+04
bqp250-1	a	2653, 27, 8491	37.6	9.8e-07, 4.1e-07, 8.7e-07	-4.87325086e+04
251, 251	b	182, 12, 582	2.5	3.9e-16, 1.9e-13, 6.8e-07	-4.87279423e+04
bqp500-1	a	4933, 32, 2951	196.3	6.1e-14, 7.4e-07, 2.0e-06	-1.28402975e+05
501, 501	b	236, 17, 725	11.2	6.0e-16, 2.3e-16, 7.4e-07	-1.28380065e+05
gka1e	a	2100, 28, 7254	21.6	6.1e-07, 4.5e-07, 1.7e-11	-1.73869700e+04
201, 201	b	155, 11, 518	1.5	3.8e-16, 6.1e-12, 9.9e-07	-1.73850666e+04
gka1f	a	4400, 35, 5082	189.1	2.1e-07, 2.7e-07, 1.3e-11	-6.68177773e+04
501, 501	b	290, 18, 714	15.0	5.2e-16, 2.6e-16, 9.5e-07	-6.68027091e+04

Table 4: Computational results for Example 4 with $\texttt{tol}_0 = 9 \times 10^{-1}$, $\hat{\epsilon} = 1 \times 10^{-3}$, $\nu = 10^3$ and $\kappa_c = 1 \times 10^3$.

problem size	solver	it it0, it1, it2	cpu [s]	kkt η_p, η_d, η_c	obj $\langle C, X\rangle$
soybean-small.2	a	300, 3, 8	0.2	1.2e-07, 8.2e-07, 1.3e-15	-2.85459820e+03
47, 48	b	73, 4, 15	0.1	2.6e-16, 3.6e-08, 6.3e-08	-2.85459785e+03
soybean-large.2	a	300, 5, 23	2.7	8.9e-08, 4.4e-09, 1.5e-16	-1.05819099e+04
307, 308	b	2, 7, 28	0.4	6.3e-16, 3.1e-09, 8.6e-07	-1.05818767e+04
spambase-small.2	a	300, 5, 12	2.4	5.1e-09, 1.7e-07, 2.2e-16	-1.14776479e+08
300, 301	b	1, 6, 25	0.3	6.4e-16, 1.2e-08, 3.2e-07	-1.14775890e+08
spambase-medium.2	a	200, 7, 41	60.8	1.0e-08, 4.3e-07, 4.5e-15	-6.57364585e+08
900, 901	b	1, 6, 25	3.6	1.2e-15, 1.3e-08, 1.4e-07	-6.57361738e+08
spambase-large.2	a	200, 11, 129	367.8	1.4e-08, 5.9e-07, 3.8e-15	-1.37542086e+09
1500, 1501	b	1, 6, 25	13.2	1.4e-15, 1.4e-08, 8.9e-08	-1.37541598e+09
abalone-small.2	a	300, 4, 17	1.2	2.2e-09, 1.3e-07, 1.9e-16	-2.58821083e+04
200, 201	b	2, 7, 28	0.2	5.1e-16, 1.4e-09, 2.1e-07	-2.58820977e+04
abalone-medium.2	a	300, 5, 16	4.3	4.5e-09, 4.0e-07, 6.2e-16	-5.67601863e+04
400, 401	b	2, 5, 20	0.6	6.4e-16, 7.2e-08, 2.6e-07	-5.67601336e+04
abalone-large.2	a	200, 7, 31	48.1	1.2e-08, 4.0e-09, 2.8e-17	-1.36000345e+05
1000, 1001	b	2, 5, 21	4.3	1.1e-15, 7.5e-08, 1.7e-07	-1.36000212e+05
segment-small.2	a	300, 5, 17	3.6	8.4e-09, 3.9e-07, 1.2e-15	-1.88711369e+07
400, 401	b	2, 7, 29	0.7	6.4e-16, 3.4e-09, 2.1e-07	-1.88711158e+07
segment-medium.2	a	200, 6, 38	12.1	8.2e-11, 2.5e-09, 1.4e-15	-3.32233510e+07
700, 701	b	2, 7, 29	2.5	1.0e-15, 3.7e-09, 2.0e-07	-3.32233040e+07
segment-large.2	a	200, 5, 58	44.2	4.1e-08, 1.2e-08, 3.2e-16	-4.74399120e+07
1000, 1001	b	2, 7, 29	5.0	1.2e-15, 3.7e-09, 1.8e-07	-4.74398407e+07
housing.2	a	200, 5, 21	5.9	2.0e-09, 6.6e-07, 8.0e-16	-1.67407790e+08
506, 507	b	2, 7, 29	1.3	8.1e-16, 1.4e-09, 1.3e-07	-1.67407726e+08

Table 5: Computational results for Example 5 with $\texttt{tol}_0 = 1 \times 10^{-4}$, $\hat{\epsilon} = 2 \times 10^{-3}$, $\nu = 5 \times 10^3$ and $\kappa_c = 5 \times 10^2$.

problem size	solver	it it0, it1, it2	cpu [s]	kkt η_p, η_d, η_c	obj $\langle C, X\rangle$
nonsym(7,4)	a	300, 5, 208	3.5	6.5e-08, 4.8e-08, 6.4e-17	5.07407706e+00
343, 21951	b	2000, 5, 277	16.8	2.7e-16, 4.6e-12, 4.0e-07	5.07410884e+00
nonsym(8,4)	a	200, 5, 270	6.9	1.0e-07, 6.7e-08, 1.5e-17	5.74082890e+00
512, 46655	b	2000, 5, 231	41.1	2.3e-16, 3.0e-12, 6.7e-07	5.74091793e+00
nonsym(9,4)	a	200, 6, 749	24.3	2.8e-08, 4.2e-08, 9.6e-18	1.06613332e+00
729, 91124	b	2000, 5, 173	89.4	3.6e-16, 6.0e-12, 7.0e-07	1.06615469e+00
nonsym(10,4)	a	200, 7, 1017	57.3	1.4e-07, 7.4e-08, 6.0e-16	1.69471513e+00
1000, 166374	b	521, 16, 1501	207.6	2.8e-16, 1.5e-16, 2.4e-07	1.69472856e+00
nonsym(11,4)	a	200, 7, 1004	113.9	9.3e-08, 1.1e-07, 1.6e-16	2.91348562e+00
1331, 287495	b	2000, 8, 205	379.5	5.3e-16, 2.0e-16, 7.3e-08	2.91349308e+00
nonsym(5,5)	a	200, 6, 690	14.8	5.5e-07, 2.6e-08, 4.3e-17	3.08257445e+00
625, 50624	b	2000, 6, 250	67.5	2.7e-16, 1.1e-13, 4.1e-07	3.08260164e+00
nonsym(6,5)	a	200, 6, 896	96.4	7.9e-07, 4.6e-07, 5.6e-17	3.09572653e+00
1296, 194880	b	2000, 10, 578	387.2	3.4e-16, 3.0e-16, 5.0e-07	3.09577257e+00
sym_rd(3,25)	a	237, 3, 97	3.3	1.9e-07, 3.9e-07, 5.4e-18	1.62974610e+00
351, 23750	b	598, 8, 2330	23.5	1.3e-16, 2.3e-16, 6.6e-08	1.62974618e+00
sym_rd(3,30)	a	300, 3, 294	9.2	4.3e-07, 6.0e-07, 7.5e-18	1.82416334e+00
496, 46375	b	793, 8, 2209	51.5	1.2e-16, 8.7e-17, 3.1e-07	1.82416600e+00
sym_rd(3,35)	a	200, 3, 790	19.4	5.0e-07, 8.4e-07, 2.3e-17	1.82999294e+00
666, 82250	b	1253, 9, 4270	229.0	1.3e-16, 1.1e-16, 1.1e-07	1.82999326e+00
sym_rd(3,40)	a	200, 3, 651	34.9	8.2e-07, 7.7e-07, 1.5e-17	1.99315615e+00
861, 135750	b	1125, 9, 3102	313.5	1.4e-16, 1.9e-16, 7.7e-08	1.99315460e+00
sym_rd(3,45)	a	200, 3, 1136	73.4	5.1e-07, 8.1e-07, 1.0e-17	2.14077028e+00
1081, 211875	b	1114, 9, 3594	610.6	1.4e-16, 1.5e-16, 7.6e-08	2.14077153e+00

Table 5 continued from previous page

problem size	solver	it it0, it1, it2	cpu [s]	kkt η_p, η_d, η_c	obj $\langle C, X\rangle$
sym_rd(3,50)	a	200, 5, 1375	141.5	8.9e-08, 2.1e-07, 1.6e-17	2.06949920e+00
1326, 316250	b	1042, 9, 3783	1062.8	1.4e-16, 1.1e-16, 5.6e-08	2.06949928e+00
sym_rd(4,25)	a	236, 3, 522	5.4	8.2e-07, 5.9e-08, 1.5e-17	8.56184456e+00
325, 20474	b	992, 7, 2615	26.5	1.1e-16, 4.4e-16, 9.6e-08	8.56184319e+00
sym_rd(4,30)	a	183, 3, 690	9.5	4.0e-07, 6.7e-08, 3.5e-17	9.56021326e+00
465, 40919	b	640, 9, 3848	69.8	1.3e-16, 4.1e-17, 7.4e-08	9.56021733e+00
sym_rd(4,40)	a	280, 12, 876	56.9	5.3e-08, 2.9e-08, 1.6e-17	1.15471511e+01
820, 123409	b	251, 6, 4506	212.3	1.2e-16, 2.2e-14, 7.9e-07	1.15471604e+01
sym_rd(4,45)	a	278, 11, 464	86.6	8.8e-07, 6.0e-09, 4.3e-09	1.18424676e+01
1035, 194579	b	266, 7, 5359	439.3	1.2e-16, 8.1e-16, 7.8e-07	1.18424671e+01
sym_rd(4,50)	a	178, 11, 420	102.9	3.9e-08, 9.8e-07, 1.5e-17	1.30418152e+01
1275, 292824	b	273, 7, 4987	691.5	1.2e-16, 1.3e-15, 3.1e-07	1.30418139e+01
sym_rd(5,15)	a	196, 3, 716	28.1	5.4e-07, 1.5e-07, 4.0e-18	3.49345484e+00
816, 54263	b	321, 12, 7248	464.6	1.4e-16, 7.8e-16, 2.0e-07	3.49346038e+00
sym_rd(5,20)	a	200, 3, 1800	354.8	2.6e-07, 7.9e-08, 1.2e-17	4.17920915e+00
1771, 230229	b	621, 7, 2727	1696.7	1.4e-16, 1.5e-15, 6.5e-07	4.17923349e+00
sym_rd(6,15)	a	200, 3, 1604	29.2	1.1e-07, 6.9e-08, 2.3e-17	2.70986961e+01
680, 38759	b	536, 8, 2518	115.9	1.4e-16, 2.3e-16, 9.2e-08	2.70986955e+01
sym_rd(6,20)	a	185, 3, 1531	218.7	4.4e-08, 4.6e-07, 5.5e-18	3.15083192e+01
1540, 177099	b	1374, 10, 5545	1685.0	1.4e-16, 9.3e-16, 3.2e-07	3.15083629e+01
nsym_rd([20,20,20])	a	300, 4, 855	7.9	2.3e-07, 2.2e-07, 1.1e-16	3.47771560e+00
400, 44099	b	2000, 6, 532	27.2	1.5e-16, 1.0e-13, 2.7e-07	3.47771693e+00
nsym_rd([20,25,25])	a	200, 4, 797	10.6	1.2e-07, 8.7e-08, 3.7e-17	2.78569320e+00
500, 68249	b	1969, 13, 2313	84.0	1.3e-16, 3.7e-16, 1.4e-07	2.78569252e+00
nsym_rd([25,20,25])	a	200, 4, 807	11.3	1.1e-07, 4.4e-08, 1.1e-18	2.77557182e+00
500, 68249	b	1311, 10, 1230	54.1	1.2e-16, 2.8e-17, 3.6e-07	2.77557148e+00
nsym_rd([25,25,20])	a	200, 4, 481	9.3	1.1e-07, 6.2e-09, 1.8e-17	2.87657210e+00
500, 68249	b	1236, 12, 1668	58.6	1.4e-16, 4.6e-17, 1.3e-07	2.87657222e+00
nsym_rd([25,25,25])	a	200, 5, 1886	30.2	9.7e-07, 2.6e-08, 1.2e-17	2.83000543e+00
625, 105624	b	1787, 10, 6095	260.7	1.3e-16, 1.9e-16, 8.8e-08	2.83000239e+00
nsym_rd([30,30,30])	a	200, 5, 995	46.5	8.2e-07, 1.4e-07, 8.7e-18	3.03776026e+00
900, 216224	b	2000, 12, 6889	690.1	1.2e-16, 2.2e-16, 5.8e-07	3.03775583e+00
nsym_rd([35,35,35])	a	200, 4, 1464	136.6	1.2e-07, 8.0e-07, 4.0e-17	3.07047349e+00
1225, 396899	b	2000, 8, 1741	621.2	1.3e-16, 6.4e-16, 8.4e-07	3.07047539e+00
nsym_rd([40,40,40])	a	200, 3, 461	168.8	7.2e-07, 5.2e-07, 3.4e-17	3.87873704e+00
1600, 672399	b	2000, 22, 3925	2311.1	1.2e-16, 2.8e-17, 4.0e-07	3.87873669e+00
nsym_rd([7,7,7,7])	a	300, 4, 220	3.6	4.3e-07, 1.1e-07, 2.2e-17	3.33237018e+00
343, 21951	b	2000, 6, 245	16.8	1.3e-16, 1.7e-13, 1.3e-07	3.33237088e+00
nsym_rd([8,8,8,8])	a	200, 4, 377	8.5	6.1e-08, 3.4e-08, 6.2e-18	2.83768767e+00
512, 46655	b	2000, 10, 1068	60.4	1.2e-16, 9.9e-16, 5.2e-08	2.83768891e+00
nsym_rd([9,9,9,9])	a	200, 4, 520	19.2	3.0e-07, 3.1e-07, 2.4e-17	3.10894878e+00
729, 91124	b	2000, 10, 886	126.4	1.6e-16, 3.3e-15, 8.2e-07	3.10895225e+00

6 Concluding remarks

We have analyzed and implemented a squared smoothing Newton method via the Huber smoothing function for solving semidefinite programming problems (SDPs). With a careful design of the algorithmic framework, our theoretical analysis has shown that the proposed algorithm is well-defined, guarantees global convergence and admits a superlinear convergence rate under the primal and dual constraint nondegenerate conditions. Besides establishing the elegant convergence properties, we have also conducted extensive numerical experiments on solving various classes of SDPs to evaluate the practical performance of our algorithms. We have compared our method with the state-of-the-art SDP solver `SDPNAL+` and the numerical results have demonstrated the excellent efficiency of our algorithm. We note that the current implementation of the algorithm is not as mature as we would hope for since the performance may depend sensitively on some parameters. However, given the promising numerical results on the tested examples, we are inspired to conduct a more robust implementation in our future work.

Acknowledgments. The research of Defeng Sun is supported in part by the Hong Kong Research Grants Council under grant 15304019, and the research of Kim-Chuan Toh is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 3 grant call (MOE-2019-T3-1-010).

Declaration

Conflict of interest. The authors declare that they have no conflict of interest.

A Proof of Proposition 4

Part 1 is a direct consequence of [39, Proposition 4.3]. We next prove part 2. Since $h(\cdot,\cdot)$ is locally Lipschitz continuous on $\mathbb{R}\times\mathbb{R}$, then by [47, Theorem 9.67], there exist continuously differentiable functions $h_\ell : \mathbb{R}\times\mathbb{R}\to\mathbb{R}$, $\ell\geq 1$, converging uniformly to h and satisfying

$$\|h_\ell'(\tau,\xi)\|\leq L,\quad \forall(\tau,\xi)\in\mathcal{J}:=[\epsilon-\nu,\epsilon+\nu]\times(\bigcup_{i=1}^{n}[d_i-\nu_i,d_i+\nu_i]),$$

with some constants $L>0$, $\nu>0$ and $\nu_i>0$. For any $(\tau,H)\in\mathbb{R}\times\mathbb{S}^n$ with $H=\tilde{P}\mathrm{diag}(\tilde{d}_1,\ldots,\tilde{d}_n)\tilde{P}^T$, define

$$\Phi_\ell(\tau,H)=\tilde{P}\mathrm{diag}(h_\ell(\tau,\tilde{d}_1),\ldots,h_\ell(\tau,\tilde{d}_n))\tilde{P}^T,\quad \ell\geq 1.$$

From [63], we may assume that there exists a neighborhood of (ϵ,W), denoted by $\mathcal{U}$, such that $(\tau,\tilde{d}_i)\in\mathcal{J}$, for all $i=1,\ldots,n$, and $(\tau,H)\in\mathcal{U}$. Note that Φ_ℓ converges to Φ uniformly on $\mathcal{U}$. Fix (τ_1,H_1), $(\tau_2,H_2)\in\mathcal{U}$ such that $(\tau_1,H_1)\neq(\tau_2,H_2)$. Then, for any $\hat{L}>0$ and ℓ sufficiently large, it holds that

$$\|\Phi_\ell(\tau,H)-\Phi(\tau,H)\|\leq\hat{L}\,\|(\tau_1,H_1)-(\tau_2,H_2)\|,\quad\forall(\tau,H)\in\mathcal{U}.\tag{44}$$

Using (44), for any $(\tau,H)\in\mathcal{U}$, it follows that

$$\begin{aligned}
&\|\Phi(\tau_1,H_1)-\Phi(\tau_2,H_2)\|\\
\leq\ &\|\Phi(\tau_1,H_1)-\Phi_\ell(\tau_1,H_1)\|+\|\Phi_\ell(\tau_1,H_1)-\Phi_\ell(\tau_2,H_2)\|\\
&+\|\Phi_\ell(\tau_2,H_2)-\Phi(\tau_2,H_2)\|\\
\leq\ &2\hat{L}\,\|(\tau_1,H_1)-(\tau_2,H_2)\|\\
&+\left\|\int_0^1\Phi_\ell'(\tau_1+t(\tau_2-\tau_1),H_1+t(H_2-H_1))(\tau_1-\tau_2,H_2-H_1)dt\right\|\\
\leq\ &(2\hat{L}+L)\,\|(\tau_1,H_1)-(\tau_2,H_2)\|
\end{aligned}$$

for $\ell\geq 1$ sufficiently large. Since $\hat{L}>0$ is arbitrary, we see that Φ is locally Lipschitz continuous with modulus $L>0$.

Then, we turn to prove part 3. Let us recall that $\Phi(0, W) = \sum_{j=1}^{r} h(0,\lambda_j)Q_j$ and write

$$\Phi(t\tau, W + tH) = \sum_{j=1}^{r} h(0,\lambda_j)Q_j(t) + \sum_{i=1}^{n} \left(h(t\tau, d_i(t)) - h(0,d_i)\right) p_i(t)p_i(t)^T.$$

Then, it follows that

$$\begin{aligned}&\lim_{t\downarrow 0} \frac{1}{t}\left(\Phi(t\tau, W+tH) - \Phi(0,W)\right)\\ &= \sum_{j=1}^{r} h(0,\lambda_j) \lim_{t\downarrow 0} \frac{1}{t}\left(Q_j(t) - Q_j\right) + \sum_{i=1}^{n} \lim_{t\downarrow 0} \frac{1}{t}\left(h(t\tau, d_i(t)) - h(0,d_i)\right) p_i(t)p_i(t)^T.\end{aligned}$$

By [64, Eq. (2.9)], for any $j = 1, \ldots, r$, the following holds:

$$\begin{aligned}&\lim_{t\downarrow 0} \frac{1}{t}\left(Q_j(t) - Q_j\right) = Q'(0)(H)\\ &= \frac{1}{2} \sum_{1\le k\ne j\le r} \frac{h(0,\lambda_k) - h(0,\lambda_j)}{\lambda_k - \lambda_j}\left(Q_jHQ_k + Q_kHQ_j\right).\end{aligned}$$

From [64, Proposition 3.1], for any $i = 1, \ldots, n$, we know that $d_i'(W;\ H)$ is well-defined, and it holds that

$$\begin{aligned}&\lim_{t\downarrow 0} \frac{1}{t}\left(h(t\tau, d_i(t)) - h(0,d_i)\right) = h'((0,d_i);\ (\tau, d_i'(W;\ H)))\\ &= \sum_{i\in\alpha} \left(d_i'(W;\ H) - \frac{|\tau|}{2}\right) p_ip_i^T + \sum_{i\in\beta} h(\tau, d_i'(W;\ H))p_ip_i^T,\end{aligned}$$

where we have used the fact that h is directionally differentiable with

$$h'((0,u);\ (\tau,v)) = \begin{cases} v - \frac{|\tau|}{2} & u > 0\\ h(\tau,v) & u = 0\\ 0 & u < 0\end{cases}, \quad \forall(\tau,v)\in\mathbb{R}\times\mathbb{R}.$$

Since $p_i(t) \to p_i$ as $t \downarrow 0$, $\lim_{t\downarrow 0} \frac{1}{t}\left(\Phi(t\tau, W+tH) - \Phi(0,W)\right)$ exists. Hence Φ is directionally differentiable at $(0, W)$, and (15) holds true. Finally, the explicit expression of the directional differential $d_i'(W;\ H)$ $(1 \le i \le n)$ is again obtained from [64, Proposition 3.1].

Finally, we prove part 4. Based on part 1, Φ is continuously differentiable at any (ϵ, W) with $\epsilon \neq 0$. Hence, Φ is naturally strongly semismooth at these points. Thus, we only need to show that Φ is strongly semismooth at $(0, W)$ for any $W \in \mathbb{S}^n$. We have already known that Φ is locally Lipschitz continuous and directionally differentiable everywhere. By [33, Theorem 3.7], we only need to show that for any $(\tau, H) \in \mathbb{R} \times \mathbb{S}^n$ with $\|(\tau, H)\| \to 0$,

$$\Phi(\tau, W + H) - \Phi(0, W) - \Phi'((\tau, W + H);\ (\tau, H)) = O\left(\|(\tau, H)\|^2\right). \quad (45)$$

First, similar to the proof of part 3, one can show for any $(\tau, H) \in \mathbb{R} \times \mathbb{S}^n$,

$$\begin{aligned}\Phi'((\tau, W + H);\ (\tau, H)) &= \sum_{j=1}^{r} h(\tau, \lambda_j(1)) Q_j'(1)(H) \\ &+ \sum_{i=1}^{n} h'((\tau, d_i(1));\ (\tau, d_i'(W + H;\ H))) p_i(1) p_i(1)^T.\end{aligned} \quad (46)$$

Then, by the fact that d_i is strongly semismooth everywhere (see for example [64, Proposition 3.2]), we deduce that

$$\begin{aligned}&\Phi(\tau, W + H) - \Phi(0, W) \\ &= \sum_{j=1}^{r} h(\tau, \lambda_j(1)) \left(Q_j(1) - Q_j\right) + \sum_{i=1}^{n} \left(h(\tau, d_i(1)) - h(0, d_i)\right) p_i p_i^T \\ &= \sum_{j=1}^{r} h(\tau, \lambda_j(1)) Q_j'(1)(H) + O\left(\|H\|^2\right) \\ &\quad + \sum_{i=1}^{n} \left(h'((\tau, d_i(1));\ (\tau, d_i'(W + H;\ H)))\right) p_i p_i^T + O\left(\|(\tau, H)\|^2\right) \\ &= \sum_{j=1}^{r} h(\tau, \lambda_j(1)) Q_j'(1)(H) \\ &\quad + \sum_{i=1}^{n} \left(h'((\tau, d_i(1));\ (\tau, d_i'(W + H;\ H)))\right) p_i(1) p_i(1)^T + O\left(\|(\tau, H)\|^2\right),\end{aligned}$$

which together with (46) and the fact that p_i is analytic around W implies (45). Thus, the proof is completed.

B Proof of Lemma 5

First, let $V \in \partial_B \Phi(0, W)$. By the definition of $\partial_B \Phi(0, W)$, there exists a sequence $\{(\epsilon^k, W^k)\}$ converging to $(0, W)$ with $\epsilon^k \neq 0$ such that $V = \lim_{k \to \infty} \Phi'(\epsilon^k, W^k)$. Let each W^k have the following spectral decomposition:

$$W^k = P^k D^k (P^k)^T, \quad D^k := \operatorname{diag}\left(D_\alpha^k, D_\beta^k, D_\gamma^k\right),$$

where $D^k_\alpha = \mathrm{diag}(d^k_1,\ldots,d^k_{|\alpha|})$, $D^k_\beta = \mathrm{diag}(d^k_{|\alpha|+1},\ldots,d^k_{|\alpha|+|\beta|})$, and $D^k_\gamma = \mathrm{diag}(d^k_{|\alpha|+|\beta|+1},\ldots,d^k_n)$, with $d^k_1 \geq \cdots \geq d^k_n$. For simplicity, denote $d^k = (d^k_1,\ldots,d^k_n)^T \in \mathbb{R}^n$. By taking a subsequence if necessary, we may assume without loss of generality that (a) $\lim_{k\to\infty} D^k = D$, $\lim_{k\to\infty} P^k = P$, and (b) both sequences $\{\Omega(\epsilon^k,d^k)\}$ and $\{\mathcal{D}(\epsilon^k,d^k)\}$ converge, where $\Omega(\epsilon^k,d^k)$ and $\mathcal{D}(\epsilon^k,d^k)$ are defined as in (12). In particular, we see that $\{\Omega(\epsilon^k,d^k)_{\alpha\alpha}\}$, $\{\Omega(\epsilon^k,d^k)_{\alpha\beta}\}$ converge to two matrices of all ones of suitable sizes, respectively, $\{\Omega(\epsilon^k,d^k)_{\alpha\gamma}\}$ converges to $\Omega_0(d)_{\alpha\gamma}$ and the limit of the sequence $\{\Omega(\epsilon^k,d^k)_{\beta\beta}\}$ exists. Moreover, let

$$\mathcal{D}(\epsilon^k,d^k) = \mathrm{diag}(\mathcal{D}(\epsilon^k,d^k)_\alpha, \mathcal{D}(\epsilon^k,d^k)_\beta, \mathcal{D}(\epsilon^k,d^k)_\gamma).$$

It holds that $\{\mathcal{D}(\epsilon^k,d^k)_\gamma\}$ converges to the zero matrix, and the limits of $\{\mathcal{D}(\epsilon^k,d^k)_\alpha\}$ and $\{\mathcal{D}(\epsilon^k,d^k)_\beta\}$ exist.

From Proposition 4 and $\epsilon_k \neq 0$, for any $(\tau,H)\in\mathbb{R}\times\mathbb{S}^n$, we see that

$$\Phi'(\epsilon^k,W^k)(\tau,H) = P^k\left[\Omega(\epsilon^k,d^k)\circ\tilde{H}^k + \tau\mathcal{D}(\epsilon^k,d^k)\right](P^k)^T, \tag{47}$$

where $\tilde{H}_k := (P^k)^T H P^k$. Taking limit of both sides in (47) yields that

$$\begin{aligned} V(\tau,H) = P &\begin{pmatrix} \tilde{H}_{\alpha\alpha} & \tilde{H}_{\alpha\beta} & \Omega_0(d)\circ\tilde{H}_{\alpha\gamma} \\ \tilde{H}^T_{\alpha\beta} & \lim_{k\to\infty}\Omega(\epsilon^k,d^k)_{\beta\beta}\circ\tilde{H}_{\beta\beta} & 0 \\ \tilde{H}^T_{\alpha\gamma}\circ\Omega_0(d)^T & 0 & 0 \end{pmatrix} P^T \\ &+ \tau P\begin{pmatrix} \lim_{k\to\infty}\mathcal{D}(\epsilon^k,d^k)_\alpha & 0 & 0 \\ 0 & \lim_{k\to\infty}\mathcal{D}(\epsilon^k,d^k)_\beta & 0 \\ 0 & 0 & 0 \end{pmatrix} P^T. \end{aligned}$$

For each $k\geq 1$, define $Y^k := P\mathrm{diag}(0, D^k_\beta, 0)P^T$ and $\tilde{Y}^k := P^T Y^k P$. Note that the mapping $\mathcal{L}$ is F-differentiable at (ϵ^k, Y^k) since $\epsilon_k \neq 0$. For k sufficiently large, we have from (16) and Proposition 4 that

$$
\begin{aligned}
&\mathcal{L}'(\epsilon^k, Y^k)(\tau, H)\\
&= \lim_{t\downarrow 0} \frac{\mathcal{L}(\epsilon^k + t\tau, Y^k + tH) - \mathcal{L}(\epsilon^k, Y^k)}{t}\\
&= P \begin{pmatrix} \tilde{H}_{\alpha\alpha} & \tilde{H}_{\alpha\beta} & [\Omega_0(d)]_{\alpha\gamma} \circ \tilde{H}_{\alpha\gamma} \\ \tilde{H}_{\alpha\beta}^T & \Phi'_{|\beta|}(\epsilon^k, D^k_\beta)(\tau, \tilde{H}_{\beta\beta}) & 0 \\ [\Omega_0(d)]_{\alpha\gamma}^T \circ H_{\alpha\gamma}^T & 0 & 0 \end{pmatrix} P^T\\
&\quad - \frac{\tau}{2}\mathrm{sgn}(\epsilon^k) \sum_{i\in\alpha} p_i p_i^T\\
&= P \begin{pmatrix} \tilde{H}_{\alpha\alpha} & \tilde{H}_{\alpha\beta} & [\Omega_0(d)]_{\alpha\gamma} \circ \tilde{H}_{\alpha\gamma} \\ \tilde{H}_{\alpha\beta}^T & \Omega(\epsilon^k, d^k)_{\beta\beta} \circ \tilde{H}_{\beta\beta} & 0 \\ [\Omega_0(d)]_{\alpha\gamma}^T \circ H_{\alpha\gamma}^T & 0 & 0 \end{pmatrix} P^T\\
&\quad + \tau P \begin{pmatrix} \mathcal{D}(\epsilon_k, d^k)_\alpha & 0 & 0 \\ 0 & \mathcal{D}(\epsilon^k, d^k)_\beta & 0 \\ 0 & 0 & 0 \end{pmatrix} P^T,
\end{aligned}
$$

which implies that $V(\tau, H) = \lim_{k\to\infty} \mathcal{L}'(\epsilon^k, Y^k)(\tau, H)$. Hence, $V \in \partial_B \mathcal{L}(0, 0)$.

Conversely, choose $V \in \partial_B \mathcal{L}(0, 0)$. By definition, there exists a sequence $\{(\epsilon^k, Y^k)\}$ converging to $(0, 0)$ with $\epsilon^k \neq 0$ such that $V = \lim_{k\to\infty} \mathcal{L}'(\epsilon^k, Y^k)$. Let $\tilde{Y}^k := P^T Y^k P$. Assume that $\tilde{Y}^k_{\beta\beta}$ has the spectral decomposition: $\tilde{Y}^k_{\beta\beta} = U^k \tilde{D}^k_\beta (U^k)^T$, where $\tilde{D}^k_\beta = \mathrm{diag}(\tilde{z}^k)$, $\tilde{z}^k := (\tilde{z}^k_1, \ldots, \tilde{z}^k_{|\beta|})^T \in \mathbb{R}^{|\beta|}$, $\tilde{z}^k_1 \geq \cdots \geq \tilde{z}^k_{|\beta|}$, and $U^k \in \mathbb{R}^{|\beta|\times|\beta|}$ is orthogonal. Then, for any $(\tau, H) \in \mathbb{R} \times \mathbb{S}^n$ with $\tilde{H} := P^T H P$, we get from (16) and Proposition 4 that

$$
\begin{aligned}
&\mathcal{L}'(\epsilon^k, Y^k)(\tau, H)\\
&= P \begin{pmatrix} \tilde{H}_{\alpha\alpha} & \tilde{H}_{\alpha\beta} & [\Omega_0(d)]_{\alpha\gamma} \circ \tilde{H}_{\alpha\gamma} \\ \tilde{H}_{\alpha\beta}^T & \hat{H}_{\beta\beta} & 0 \\ [\Omega_0(d)]_{\alpha\gamma}^T \circ H_{\alpha\gamma}^T & 0 & 0 \end{pmatrix} P^T - \frac{\tau}{2}\mathrm{sgn}(\epsilon^k) \sum_{i\in\alpha} p_i p_i^T,
\end{aligned}
$$

where $\hat{H}_{\beta\beta} := U^k \left[\Omega(\epsilon^k, \tilde{z}^k) \circ \left((U^k)^T \tilde{H}_{\beta\beta} U^k\right) + \tau \mathcal{D}(\epsilon^k, \tilde{z}^k)\right] (U^k)^T$. For any $k \geq 1$, define

$$
W^k = W + P \begin{pmatrix} 0 & 0 & 0 \\ 0 & \tilde{Y}^k_{\beta\beta} & 0 \\ 0 & 0 & 0 \end{pmatrix} P^T, \quad \tilde{W}^k = P^T W^k P = \begin{pmatrix} D_\alpha & 0 & 0 \\ 0 & \tilde{Y}^k_{\beta\beta} & 0 \\ 0 & 0 & D_\gamma \end{pmatrix}
$$

where

$$
D_\alpha = \mathrm{diag}(d_1, \ldots, d_{|\alpha|}), \quad D_\gamma = \mathrm{diag}(d_{|\alpha|+|\beta|+1}, \ldots, d_n).
$$

Moreover, we partition P as $P = (P_\alpha, P_\beta, P_\gamma)$, set $P^k = (P_\alpha, P_\beta U^k, P_\gamma)$ and construct a vector $d^k \in \mathbb{R}^n$ as follows

$$d_i^k = \begin{cases} d_i & i \in \alpha \cup \gamma \\ \tilde{z}^k_{i-|\alpha|} & i \in \beta \end{cases}.$$

Clearly, it holds that $W^k = P^k \mathrm{diag}(d^k)(P^k)^T$. Since $\epsilon^k \neq 0$, Φ is F-differentiable at (ϵ^k, W^k), and

$$\Phi'(\epsilon^k, W^k)(\tau, H) = P^k \left[\Omega(\epsilon^k, d^k) \circ \left((P^k)^T H P^k\right) + \tau \mathcal{D}(\epsilon^k, d^k)\right] (P^k)^T.$$

Let us now assume without loss of generality that the three sequences $\{U^k\}$, $\{\Omega(\epsilon^k, d^k)\}$ and $\{\mathcal{D}(\epsilon, d^k)\}$ converge (since they are all uniformly bounded). Then, simple calculations show that

$$\lim_{k\to\infty} [\Omega(\epsilon^k, d^k)]_{ij} = \begin{cases} 1 & i \in \alpha, j \in \alpha \cup \beta \\ \Omega_0(d) & i \in \alpha, j \in \gamma \\ 0 & i \in \beta \cup \gamma, j \in \gamma \\ [\lim\limits_{k\to\infty} \Omega(\epsilon^k, \tilde{z}^k)]_{(i-|\alpha|)(j-|\alpha|)} & i \in \beta, j \in \beta \end{cases},$$

and that

$$\lim_{k\to\infty} \mathcal{D}(\epsilon^k, d^k) = \begin{pmatrix} \lim\limits_{k\to\infty} \mathcal{D}(\epsilon^k, d_\alpha) & 0 & 0 \\ 0 & \lim\limits_{k\to\infty} \mathcal{D}(\epsilon^k, \tilde{z}^k) & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

where $d_\alpha := (d_1, \ldots, d_{|\alpha|})^T \in \mathbb{R}^{|\alpha|}$. As a consequence, we get

$$\lim_{k\to\infty} (P^k)^T \left(\mathcal{L}'(\epsilon^k, Y^k)(\tau, H) - \Phi'(\epsilon^k, W^k)(\tau, H)\right) P^k = 0, \quad \forall (\tau, H) \in \mathbb{R} \times \mathbb{S}^n,$$

which further implies that $V(\tau, H) = \lim\limits_{k\to\infty} \Phi'(\epsilon^k, W^k)(\tau, H)$, for all $(\tau, H) \in \mathbb{R} \times \mathbb{S}^n$. Then, $V \in \partial_B \Phi(0, W)$. Therefore, the proof is completed.

References

[1] Vandenberghe, L., Boyd, S.: Semidefinite programming. SIAM review **38**(1), 49–95 (1996)

[2] Bonnans, J.F., Shapiro, A.: Perturbation Analysis of Optimization Problems. Springer, New York (2013)

[3] Chan, Z.X., Sun, D.: Constraint nondegeneracy, strong regularity, and nonsingularity in semidefinite programming. SIAM Journal on optimization **19**(1), 370–396 (2008)

[4] Majumdar, A., Hall, G., Ahmadi, A.A.: Recent scalability improvements for semidefinite programming with applications in machine learning, control, and robotics. Annual Review of Control, Robotics, and Autonomous Systems **3**, 331–360 (2020)

[5] Sun, D.: The strong second-order sufficient condition and constraint nondegeneracy in nonlinear semidefinite programming and their implications. Mathematics of Operations Research **31**(4), 761–776 (2006)

[6] Wolkowicz, H., Saigal, R., Vandenberghe, L.: Handbook of Semidefinite Programming: Theory, Algorithms, and Applications vol. 27. Springer, New York (2012)

[7] Todd, M.J.: Semidefinite optimization. Acta Numerica **10**, 515–560 (2001)

[8] Toh, K.-C.: Solving large scale semidefinite programs via an iterative solver on the augmented systems. SIAM Journal on Optimization **14**(3), 670–698 (2004)

[9] Toh, K.-C., Kojima, M.: Solving some large scale semidefinite programs via the conjugate residual method. SIAM Journal on Optimization **12**(3), 669–691 (2002)

[10] Kim, S., Kojima, M., Mevissen, M., Yamashita, M.: Exploiting sparsity in linear and nonlinear matrix inequalities via positive semidefinite matrix completion. Mathematical programming **129**(1), 33–68 (2011)

[11] Kočvara, M.: Decomposition of arrow type positive semidefinite matrices with application to topology optimization. Mathematical Programming **190**(1-2), 105–134 (2021)

[12] Helmberg, C., Rendl, F.: A spectral bundle method for semidefinite programming. SIAM Journal on Optimization **10**(3), 673–696 (2000)

[13] Jiang, K., Sun, D., Toh, K.-C.: An inexact accelerated proximal gradient method for large scale linearly constrained convex SDP. SIAM Journal on Optimization **22**(3), 1042–1064 (2012)

[14] Yang, H., Liang, L., Carlone, L., Toh, K.-C.: An inexact projected gradient method with rounding and lifting by nonlinear programming for solving rank-one semidefinite relaxation of polynomial optimization. Mathematical Programming, 1–64 (2022)

[15] Renegar, J.: Efficient first-order methods for linear programming and semidefinite programming. arXiv preprint arXiv:1409.5832 (2014)

[16] Povh, J., Rendl, F., Wiegele, A.: A boundary point method to solve semidefinite programs. Computing **78**, 277–286 (2006)

[17] Sun, D., Toh, K.-C., Yang, L.: A convergent 3-block semiproximal alternating direction method of multipliers for conic programming with 4-type constraints. SIAM Journal on Optimization **25**(2), 882–915 (2015)

[18] Zhao, X.-Y., Sun, D., Toh, K.-C.: A Newton-CG augmented Lagrangian method for semidefinite programming. SIAM Journal on Optimization **20**(4), 1737–1765 (2010)

[19] Zheng, Y., Fantuzzi, G., Papachristodoulou, A., Goulart, P., Wynn, A.: Chordal decomposition in operator-splitting methods for sparse semidefinite programs. Mathematical Programming **180**(1-2), 489–532 (2020)

[20] Yurtsever, A., Tropp, J.A., Fercoq, O., Udell, M., Cevher, V.: Scalable semidefinite programming. SIAM Journal on Mathematics of Data Science **3**(1), 171–200 (2021)

[21] Ding, L., Yurtsever, A., Cevher, V., Tropp, J.A., Udell, M.: An optimal-storage approach to semidefinite programming using approximate complementarity. SIAM Journal on Optimization **31**(4), 2695–2725 (2021)

[22] Jarre, F., Rendl, F.: An augmented primal-dual method for linear conic programs. SIAM Journal on Optimization **19**(2), 808–823 (2008)

[23] Malick, J., Povh, J., Rendl, F., Wiegele, A.: Regularization methods for semidefinite programming. SIAM Journal on Optimization **20**(1), 336–356 (2009)

[24] Rockafellar, R.T.: Augmented Lagrangians and applications of the proximal point algorithm in convex programming. Mathematics of operations research **1**(2), 97–116 (1976)

[25] Rockafellar, R.T.: Monotone operators and the proximal point algorithm. SIAM journal on control and optimization **14**(5), 877–898 (1976)

[26] Qi, L., Sun, J.: A nonsmooth version of Newton's method. Mathematical programming **58**(1-3), 353–367 (1993)

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

[28] Burer, S., Monteiro, R.D.: A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. Mathematical

Programming **95**(2), 329–357 (2003)

[29] Boumal, N., Voroninski, V., Bandeira, A.: The non-convex Burer-Monteiro approach works on smooth semidefinite programs. Advances in Neural Information Processing Systems **29** (2016)

[30] Burer, S., Monteiro, R.D.: Local minima and convergence in low-rank semidefinite programming. Mathematical programming **103**(3), 427–444 (2005)

[31] Pataki, G.: On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. Mathematics of operations research **23**(2), 339–358 (1998)

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

[33] Sun, D., Sun, J.: Semismooth matrix-valued functions. Mathematics of Operations Research **27**(1), 150–169 (2002)

[34] Kanzow, C., Pieper, H.: Jacobian smoothing methods for nonlinear complementarity problems. SIAM Journal on Optimization **9**(2), 342–373 (1999)

[35] Chen, X., Qi, L., Sun, D.: Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities. Mathematics of computation **67**(222), 519–540 (1998)

[36] Qi, L., Chen, X.: A globally convergent successive approximation method for severely nonsmooth equations. SIAM Journal on control and Optimization **33**(2), 402–418 (1995)

[37] Gao, Y., Sun, D.: Calibrating least squares semidefinite programming with equality and inequality constraints. SIAM Journal on Matrix Analysis and Applications **31**(3), 1432–1457 (2010)

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

[39] Chen, X., Qi, H., Tseng, P.: Analysis of nonsmooth symmetric-matrix-valued functions with applications to semidefinite complementarity problems. SIAM Journal on Optimization **13**(4), 960–985 (2003)

[40] Chen, X., Tseng, P.: Non-interior continuation methods for solving semidefinite complementarity problems. Mathematical Programming

95(3), 431–474 (2003)

[41] Robinson, S.M.: Normal maps induced by linear transformations. Mathematics of Operations Research **17**(3), 691–714 (1992)

[42] Chen, B., Harker, P.T.: A non-interior-point continuation method for linear complementarity problems. SIAM Journal on Matrix Analysis and Applications **14**(4), 1168–1190 (1993)

[43] Kanzow, C.: Some noninterior continuation methods for linear complementarity problems. SIAM Journal on Matrix Analysis and Applications **17**(4), 851–868 (1996)

[44] Smale, S.: Algorithms for solving equations. In: The Collected Papers of Stephen Smale vol. 3, pp. 1263–1286. World Scientific, Singapore (2000)

[45] Sun, D., Qi, L.: On NCP-functions. Computational Optimization and Applications **13**, 201–220 (1999)

[46] Pinar, M.Ç., Zenios, S.A.: On smoothing exact penalty functions for convex constrained optimization. SIAM Journal on Optimization **4**(3), 486–511 (1994)

[47] Rockafellar, R.T., Wets, R.J.-B.: Variational Analysis vol. 317. Springer, Berlin (2009)

[48] Qi, L.: Convergence analysis of some algorithms for solving nonsmooth equations. Mathematics of operations research **18**(1), 227–244 (1993)

[49] Mifflin, R.: Semismooth and semiconvex functions in constrained optimization. SIAM Journal on Control and Optimization **15**(6), 959–972 (1977)

[50] Pereyra, V., Reinoza, A., Robinson, S.M.: Local structure of feasible sets in nonlinear programming, part I: Regularity. In: Numerical Methods: Proceedings of the International Workshop Held at Caracas, June 14–18, 1982, pp. 240–251. Springer, Berlin (1983)

[51] Robinson, S.M.: Local Structure of Feasible Sets in Nonlinear Programming, Part II: Nondegeneracy. Springer, Berlin (1984)

[52] Robinson, S.M.: Local Structure of Feasible Sets in Nonlinear Programming, Part III: Stability and Sensitivity. Springer, Berlin (1987)

[53] Robinson, S.M.: Constraint nondegeneracy in variational analysis. Mathematics of Operations Research **28**(2), 201–232 (2003)

[54] Alizadeh, F., Haeberly, J.-P.A., Overton, M.L.: Complementarity and

nondegeneracy in semidefinite programming. Mathematical programming **77**(1), 111–128 (1997)

[55] Zhao, J.-Y.: The Smoothing Function of the Nonsmooth Matrix Valued Function. Master thesis, National University of Singapore (2004)

[56] Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. Journal of the ACM (JACM) **42**(6), 1115–1145 (1995)

[57] Lovász, L.: On the Shannon capacity of a graph. IEEE Transactions on Information theory **25**(1), 1–7 (1979)

[58] Kochenberger, G., Hao, J.-K., Glover, F., Lewis, M., Lü, Z., Wang, H., Wang, Y.: The unconstrained binary quadratic programming problem: a survey. Journal of combinatorial optimization **28**, 58–81 (2014)

[59] Burer, S.: On the copositive representation of binary and continuous nonconvex quadratic programs. Mathematical Programming **120**(2), 479–495 (2009)

[60] Wiegele, A.: Biq Mac Library—A collection of Max-Cut and quadratic 0-1 programming instances of medium size. Preprint **51** (2007)

[61] Peng, J., Wei, Y.: Approximating k-means-type clustering via semidefinite programming. SIAM journal on optimization **18**(1), 186–205 (2007)

[62] Nie, J., Wang, L.: Semidefinite relaxations for best rank-1 tensor approximations. SIAM Journal on Matrix Analysis and Applications **35**(3), 1155–1179 (2014)

[63] Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, New York (2012)

[64] Shapiro, A.: On differentiability of symmetric matrix valued functions. School of Industrial and Systems Engineering, Georgia Institute of Technology (2002)