A Semismooth Newton-CG Dual Proximal Point Algorithm for Matrix Spectral Norm Approximation Problems

Caihua Chen[*], Yong-Jin Liu[†], Defeng Sun[‡] and Kim-Chuan Toh[§]

November 17, 2012

Abstract. We consider a class of matrix spectral norm approximation problems for finding an affine combination of given matrices having the minimal spectral norm subject to some prescribed linear equality and inequality constraints. These problems arise often in numerical algebra, engineering and other areas, such as finding Chebyshev polynomials of matrices and fastest mixing Markov chain models. Based on the classical analysis of proximal point algorithm and recent developments on semismooth analysis of the spectral operator, we propose a semismooth Newton-CG dual proximal point algorithm for solving the matrix norm approximation problems. Furthermore, when the primal constraint nondegeneracy condition holds for the subproblems, our semismooth Newton-CG method is proven to have a superlinear convergence rate. We also design efficient implementations for our proposed algorithm to solve a variety of instances and compare its performance with the nowadays popular first order alternating direction method of multipliers (ADMM). The results show that our algorithm substantially outperforms the ADMM, especially for the constrained cases and it is able to solve the problems efficiently to a relatively high accuracy.

Keywords: Spectral norm approximation, proximal point algorithm, spectral operator, semismooth Newton-CG method.

[*]School of Management and Engineering, Nanjing University, Nanjing, People Republic of China (cchenhuayx@gmail.com). This author's research was supported by the Chinese government CSC scholarship while visiting the National University of Singapore.

[†]Faculty of Science, Shenyang Aerospace University, Shenyang, People Republic of China (yongjin.liu.77@gmail.com). The author is supported by the National Young Natural Science Foundation of China under project grant No. 11001180 and the Program for Liaoning Excellent Talents in University.

[‡]Department of Mathematics and Risk Management Institute, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076 (mathsundf@nus.edu.sg). This author's research was supported in part by Academic Research Fund under grant R-146-000-149-112.

[§]Department of Mathematics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076 (mattohkc@nus.edu.sg).

1 Introduction

Let $\Re^{m\times n}$ be the space of $m \times n$ real matrices equipped with the standard inner product $\langle X, Y\rangle = \mathrm{Tr}(X^TY)$ for $X, Y \in \Re^{m\times n}$. Given a family of matrices $\{A_1, A_2, \ldots, A_p\}$ in $\Re^{m\times n}$, define the linear operator $\mathcal{A}$ and its adjoint $\mathcal{A}^*$, respectively, by

$$\mathcal{A}(X) := [\langle A_1, X\rangle, \ldots, \langle A_p, X\rangle]^T, \ \mathcal{A}^*(y) := \sum_{k=1}^{p} y_k A_k, \quad \forall\, X \in \Re^{m\times n},\ y \in \Re^p.$$

In this paper, we consider the following matrix norm approximation (abbreviated as MNA) problem

$$\min_{y\in\Re^p} \Big\{ \|A_0 - \mathcal{A}^*y\|_2 \mid By - b \in \mathcal{Q} \Big\}, \tag{1}$$

where $A_0 \in \Re^{m\times n}$, $B \in \Re^{(n_1+n_2)\times p}$ are given matrices, $b \in \Re^{n_1+n_2}$, $\mathcal{Q} = \{0\}^{n_1} \times \Re^{n_2}_+$ is a polyhedral cone, and $\|\cdot\|_2$ denotes the spectral norm. Without loss of generality, we assume that $m \le n$. We use $\|\cdot\|$ to denote the Frobenius norm of a matrix throughout this paper.

For the unconstrained version of (1), some theoretical questions such as uniqueness and characterizations of the best approximation have been analysed in [23, 46]. In the more general setting of a normed linear space, a general characterization theorem for the best approximation of an element in a normed linear space by elements of a finite dimensional subspace was established in [34, p. 170]. Here, in our setting, the feasible set is no longer a linear subspace.

The MNA problems arise from a variety of fields, such as numerical algebra and engineering. An illustrative example is the problem of finding the degree-t Cheybshev polynomial of a given matrix $A \in \Re^{n\times n}$, as studied in [13, 24, 38]. In this problem, one is interested in finding the degree-t monic polynomial p_t^* which minimizes the spectral norm of $p_t(A)$, i.e.,

$$\min\{\|p_t(A)\|_2 \mid p_t \text{ is a monic polynomial of degree } t\}. \tag{2}$$

Since $p_t(A)$ is an affine combination of the matrices $I, A, \ldots, A^t$, the matrix Chebyshev polynomial approximation problem is actually a special case of the MNA problem (1).

In contrast to the unconstrained example (2), some other problems may have prescribed linear constraints, for example, the fastest mixing Markov chain (FMMC) problem studied in [2,3]. Let $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ be an undirected connected graph with n nodes. The FMMC problem is to find a symmetric stochastic matrix P with $P_{ij} = 0$ for $(i, j) \notin \mathcal{E}$ that minimizes $\mu(P)$, where

$$\mu(P) = \max_{i=2,\ldots,n} |\lambda_i(P)|$$

and $\lambda_i(P)$ is the ith largest eigenvalue of P in magnitude. Let d be the vector of transition probabilities on the non-self-loop edges (labeled by $l = 1, 2, \ldots, p$). Define the matrix $B \in \Re^{n\times m}$ by

$$B_{il} := \begin{cases} 1, & \text{if edge } l \text{ is incident to vertex } i, \\ 0, & \text{otherwise} \end{cases}$$

and the matrix $E^{(l)}$ by

$$E_{ij}^{(l)} := \begin{cases} 1, & \text{if edge } l \text{ is incident to vertices } i \text{ and } j,\ i \neq j \\ -1, & \text{if edge } l \text{ is incident to vertex } i,\ i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Then by the analysis in [2, 3], the FMMC problem can be rewritten as a MNA problem in terms of d as follows:

$$\min \Big\{ \|I - (1/n)\mathbf{1}\mathbf{1}^T + \sum_{l=1}^{p} d_l E^{(l)}\|_2 \mid d \geq 0,\ Bd \leq 1 \Big\}. \tag{4}$$

Note that if the constraints in (4) are dropped, the resulting unconstrained problem

$$\min_{d} \quad \|I - (1/n)\mathbf{1}\mathbf{1}^T + \textstyle\sum_{l=1}^{p} d_l E^{(l)}\|_2 \tag{5}$$

is a mathematical model for the fastest distributed linear averaging (FDLA) problem with symmetric weights [42].

The above examples serve to motivate the study of numerical algorithms for solving the MNA problems. It is easy to show that the MNA problem (1) can be converted to the following semidefinite programming (SDP):

$$\min \Big\{ t \mid \begin{bmatrix} tI_m & A_0 - \mathcal{A}^* y \\ (A_0 - \mathcal{A}^* y)^T & tI_n \end{bmatrix} \succeq 0, \quad By \in b + \mathcal{Q} \Big\}. \tag{6}$$

Thus it may be solved by standard SDP packages such as SDPT3 [37], SeDuMi [35], or SDPNAL [45]. However, this SDP reformulation is potentially very expensive computationally since one has to deal with $(m+n) \times (m+n)$ matrix variables instead of $m \times n$ matrix variables. The computational cost and memory requirement are especially unnecessarily high when we have large $m + n$, but $m \ll n$. In the extreme case when $m = 1$, it is certainly not wise to solve the MNA problem (1) via (6). Instead one should deal with (1) directly since it is just a second-order cone problem, which requires far lower computational cost to solve compared to the SDP (6).

In this paper, we propose a semismooth Newton-CG dual proximal point algorithm (SNDPPA) to solve the MNA problem (1) directly instead of via the SDP reformulation (6). We also propose to use a first order alternating direction method of multipliers (ADMM) to obtain an initial point to warm start the SNDPPA. As we shall see later, in each iteration of the ADMM, the subproblem involved can either be solved by a fast algorithm or it has a closed form solution, due to recent advances in [7]. Hence, the ADMM can easily be applied to solve the MNA problem. For the dual proximal point algorithm (PPA), we note that the subproblem in each iteration is an unconstrained minimization problem whose objective function is convex continuously differentiable but not twice continuously differentiable. However, since the corresponding gradient is strongly semismooth, we are able to apply the inexact semismooth Newton method to solve the unconstrained minimization subproblem with a superlinear convergence rate. At each iteration of the semismooth Newton method, the Newton direction is computed by a preconditioned conjugate

gradient (CG) method. In each CG iteration, one would calculate each matrix-vector multiplication in at most $O(m^2n+pmn)+O(p(n_1+n_2))$ flops. This is in contrast to the inexact interior point proposed in [36] for the SDP reformulation where the matrix-vector multiplication would require more than $O((m+n)^3+p(m+n)^2)+O(p(n_1+n_2))$ flops. We also designed efficient implementations for our proposed algorithm to solve a variety of large scale matrix norm approximation problems and compared its performance with the popular first order ADMM. The numerical results show that our semismooth Newton-CG dual PPA substantially outperforms the ADMM and it is able to solve a variety of large scale instances with p up to 19176 and m, n up to 2000 robustly and efficiently to a relatively high accuracy.

A problem related to (1) is to minimize the largest eigenvalue ($\lambda_1(A_0 - \mathcal{A}^*y)$) of an affine combination of given symmetric matrices. In [29], the authors studied the convex smooth approximations of the nonsmooth function $\lambda_1(A_0 - \mathcal{A}^*y)$. We should mention that it is possible to modify our proposed algorithm to solve the problem of minimizing $\lambda_1(A_0 - \mathcal{A}^*y)$ subject to polyhedral constraints on y. But this is not the focus of this paper.

The remaining parts of this paper are organized as follows. In Section 2, we list some preliminaries on semismoothness mappings, the Moreau-Yosida regularization, and the spectral operator associated with the spectral norm. In Section 3, we introduce the framework of the inexact dual PPA for solving the MNA problem and establish its global and local convergence under certain conditions. In Section 4, we present a semismooth Newton-CG method for solving the subproblems in the dual PPA, and we establish its suplinear convergence under the primal constraint nondegeneracy condition of the subproblems. In addition, some numerical issues pertaining to the efficient implementation of the semismooth Newton-CG method are also addressed. In Section 5, we first give the details on the implementation of the classical ADMM for the MNA problem and then report numerical experiments for our proposed algorithm and the comparison of its performance against the ADMM. We conclude our paper in Section 6.

Notation. For any given positive integer m and n, we denote by I_n, $\mathbf{1}_{m\times n}$ and $\mathbf{0}_{m\times n}$ the $n \times n$ identity matrix, the $m \times n$ matrix of ones and zeros, respectively. We also use $\mathbf{1}_n$ and $\mathbf{0}_n$ to denote the vector of ones and zeros, respectively. We frequently drop m, n from the above notations when their dimensions are clear from the context. For any $x \in \Re^n$, Diag(x) denotes the diagonal matrix with diagonal entries $x_i, i = 1, \ldots, n$, while for any $X \in \Re^{m\times n}$, diag(X) denotes the main diagonal of X. For any $\eta > 0$, we write $\mathbb{B}_\eta := \{x \in \Re^m \mid \|x\|_1 \le \eta\}$ and $\mathcal{B}_\eta := \{X \in \Re^{m\times n} \mid \|X\|_* \le \eta\}$, where $\|\cdot\|_*$ denotes the nuclear norm of a matrix which is defined as the sum of its singular values. If $\eta = 1$, we just use $\mathbb{B}$ and $\mathcal{B}$ to denote the unit l_1 norm ball and nuclear norm ball, respectively. Let $\alpha \subseteq \{1, \ldots, n\}$ be an index set, we use $|\alpha|$ to represent the cardinality of α and X_α to denote the sub-matrix of X obtained by removing all the columns of X not in α. Let $\beta \subseteq \{1, \ldots, n\}$ be another index set, we use $X_{\alpha\beta}$ to denote the $|\alpha| \times |\beta|$ sub-matrix of X obtained by removing all the rows of $X \in \Re^{m\times n}$ not in α and all the columns of X not in β. The Hardamard product between matrices is denoted by "$\circ$", i.e., for any two matrices X and Y in $\Re^{m\times n}$, the (i, j)-th entry of $Z := X \circ Y$ is $Z_{ij} = X_{ij}Y_{ij}$.

2 Preliminaries

In this section, we review and develop some results on semismooth mappings, the Moreau-Yosida regularization and the spectral operator associated with the spectral norm, which are useful for our subsequent discussion.

Let $\mathcal{X}$ and $\mathcal{Y}$ be two finite dimensional real Euclidean spaces and $\mathcal{O}$ be an open set in $\mathcal{X}$. Suppose that $\Phi : \mathcal{O} \to \mathcal{Y}$ is a locally Lipschitz continuous function on the open set $\mathcal{O}$. By Rademacher's theorem, Φ is almost everywhere F(Fréchet)-differentiable in $\mathcal{O}$. Let $\mathcal{D}_\Phi$ be the set of points where Φ is differentiable. Let $\Phi'(x)$ be the derivative of Φ at $x \in \mathcal{D}_\Phi$. Then the B-subdifferential of Φ at $x \in \mathcal{O}$ is defined by $\partial_B \Phi(x) := \{\lim_{\mathcal{D}_\Phi \ni x^k \to x} \Phi'(x^k)\}$ and the Clarke's generalized Jacobian [4] of Φ at $x \in \mathcal{O}$, denoted by $\partial\Phi(x)$, is the convex hull of $\partial_B \Phi(x)$.

Definition 2.1 (c.f. [9, 28]) *Let $\Phi : \mathcal{O} \subset \mathcal{X} \to \mathcal{Y}$ be a locally Lipschitz continuous function on the open set $\mathcal{O}$ and x be a point in $\mathcal{O}$. The function Φ is said to be G-semismooth at $x \in \mathcal{O}$ if for any $y \to x$ and $V \in \partial\Phi(y)$, $\Phi(y) - \Phi(x) - V(y-x) = o(\|y-x\|)$. The function Φ is said to be strongly G-semismooth at x if for any $y \to x$ and $V \in \partial\Phi(y)$, $\Phi(y) - \Phi(x) - V(y-x) = O(\|y-x\|^2)$. Furthermore, if the (strongly) G-semismooth function Φ is also directionally differentiable at x, then Φ is said to be (strongly) semismooth at x.*

Let $f : \mathcal{X} \to (-\infty, +\infty]$ be a closed proper convex function [32] and $\eta > 0$ be a positive constant. Then the Moreau-Yosida regularization [27, 44] of f at $x \in \mathcal{X}$ associated with η is defined by

$$\psi_f^\eta(x) := \min_{y \in \mathcal{X}} \{ f(y) + \frac{1}{2\eta}\|y - x\|^2 \}. \tag{7}$$

The unique optimal solution to (7), denoted by $P_f^\eta(x)$, is called the proximal point at x associated with f and *eta*. Let $\mathcal{C}$ be a closed convex set in $\mathcal{X}$ and $\chi_{\mathcal{C}}(\cdot)$ be the indicator function of $\mathcal{C}$. Then, for any $\eta > 0$, the proximal point of x associated with $\chi_{\mathcal{C}}(\cdot)$ is the metric projection of x onto $\mathcal{C}$, which is denoted by $\Pi_{\mathcal{C}}(x)$.

Proposition 2.1 *(c.f. [18, 21]) Let $f : \mathcal{X} \to (-\infty, +\infty]$ be a closed proper convex function and $\eta > 0$ be a positive constant. Let ψ_f^η be the Moreau-Yosida regularization associated with f and η and P_f^η be the corresponding proximal point mapping. Then, the following properties hold.*

(i) *ψ_f^η is a continuously differentiable convex function, and*

$$\nabla\psi_f^\eta(x) = x - P_f^\eta(x), \ \ \forall\, x \in \mathcal{X}.$$

(ii) *Let $g : \mathcal{X} \to (-\infty, +\infty]$ be defined by*

$$g(x) = f^*(x/\eta), \quad \forall\, x \in \mathcal{X},$$

where f^* is the conjugate of f. Then any $x \in \mathcal{X}$ has the following unique decomposition

$$x = P_f^\eta(x) + P_g^\eta(x).$$

In what follows, we shall calculate the proximal point mapping associated with the spectral norm, which plays a crucial role in our numerical implementation. Let $X \in \Re^{m\times n}$ be given with the following singular value decomposition (SVD):

$$X = U[\mathrm{Diag}(\sigma(X))\ 0]V^T, \tag{8}$$

where $U \in \Re^{m\times m}$, $V \in \Re^{n\times n}$ are orthogonal matrices, $\sigma_1(X) \geq \ldots \geq \sigma_m(X) \geq 0$ are the singular values of X, and $\sigma(X) := (\sigma_1(X), \sigma_2(X), \ldots, \sigma_m(X))^T$. In later discussions, when the dependence of σ_i on X is clear from the context, we will drop X from these notations.

Proposition 2.2 *Let $f(\cdot) := \|\cdot\|_2$ be defined on $\Re^{m\times n}$ and $\eta > 0$. Let $X \in \Re^{m\times n}$ be given with the SVD as in* (8)*. Then it holds that*

$$P_f^\eta(X) = X - \Pi_{\mathcal{B}_\eta}(X), \tag{9}$$

where $\Pi_{\mathcal{B}_\eta}(X) = U[\mathrm{Diag}(\Pi_{\mathbb{B}_\eta}(\sigma(X)))\ 0]V^T$.

Proof: By direct computations, we have $f^* = \chi_{\mathcal{B}}$. Hence (9) follows from part (ii) of Proposition 2.1. Note that $\Pi_{\mathcal{B}_\eta}(X)$ is the unique solution of the following optimization problem: $\min\{\frac{1}{2}\|Y - X\|^2 \mid \|Y\|_* \leq \eta\}$. By von Neumann's trace inequality [39]:

$$\|\sigma(Y) - \sigma(X)\| \leq \|Y - X\| \quad \forall Y \in \Re^{m\times n},$$

we know that

$$\Pi_{\mathcal{B}_\eta}(X) = U[\mathrm{Diag}\big(\Pi_{\mathbb{B}_\eta}(\sigma(X))\big)\ 0]V^T,$$

where $\Pi_{\mathbb{B}_\eta}(\sigma(X))$ is the unique optimal solution to $\min\{\frac{1}{2}\|y - \sigma(X)\|^2 \mid \|y\|_1 \leq \eta\}$. □

With the above preparation, we are ready to give the exact expression of the projection $\Pi_{\mathcal{B}_\eta}(\cdot)$. Let X admit the SVD as in (8). Define the vector $s(\sigma)$ by

$$s_i(\sigma) = \frac{1}{i}\Big(\textstyle\sum_{j=1}^i \sigma_j - \eta\Big),\ i = 1, 2, \ldots, m.$$

Let $k_1(\sigma)$ and $k_2(\sigma)$ denote, respectively, the maximal indices of the following two sets:

$$\{i \,|\, \sigma_i > s_i(\sigma),\ 1 \leq i \leq m\}, \quad \{i \,|\, \sigma_i \geq s_i(\sigma),\ 1 \leq i \leq m\}.$$

From the breakpoint search algorithm in [17, Section 5], it follows that

$$\Pi_{\mathbb{B}_\eta}(\sigma) = \begin{cases} \sigma, & \text{if } \|\sigma\|_1 \leq \eta, \\ \max\{\sigma - s_{k_1(\sigma)}(\sigma), 0\}, & \text{otherwise,} \end{cases} \tag{10}$$

and hence $\Pi_{\mathcal{B}_\eta}(X)$ can be computed analytically. For notational convenience, we write

$$g(\sigma) = (g_1(\sigma), g_2(\sigma), \ldots, g_m(\sigma))^T := \Pi_{\mathbb{B}_\eta}(\sigma).$$

and use $\mu_1 > \mu_2 > \ldots > \mu_t$ to denote the nonzero singular values of X. Define the index sets α_k, $k = 1, 2, \ldots, t+1$ by

$$\begin{aligned} \alpha_k &:= \{i \,|\, \sigma_i = \mu_k,\ 1 \le i \le m\}, \quad k = 1, 2, \ldots, t \\ \alpha_{t+1} &:= \{i \,|\, \sigma_i = 0,\ 1 \le i \le m\}. \end{aligned}$$

If $\Pi_{\mathbb{B}_\eta}$ is F-differentiable at σ, then we define three matrices $\Omega(\sigma), \Gamma(\sigma) \in \Re^{m\times m}$ and $\mathcal{F}(\sigma) \in \Re^{m\times m}$ by

$$[\Omega(\sigma)]_{ij} := \begin{cases} \frac{g_i(\sigma)-g_j(\sigma)}{\sigma_i-\sigma_j}, & \text{if } \sigma_i \neq \sigma_j, \\ (g'(\sigma))_{ii} - (g'(\sigma))_{i,i+1}, & \text{if } \sigma_i = \sigma_j, i \in \alpha_k, |\alpha_k| \neq 1, \\ (g'(\sigma))_{ii}, & \text{otherwise}, \end{cases} \tag{11}$$

$$[\Gamma(\sigma)]_{ij} := \begin{cases} \frac{g_i(\sigma)+g_j(\sigma)}{\sigma_i+\sigma_j}, & \text{if } \sigma_i + \sigma_j \neq 0, \\ (g'(\sigma))_{ii}, & \text{otherwise}, \end{cases} \tag{12}$$

$$[\mathcal{F}(\sigma)]_{ij} := \begin{cases} (g'(\sigma))_{ij}, & \text{if } i \neq j, \\ (g'(\sigma))_{i,i+1}, & \text{if } i = j \in \alpha_k, |\alpha_k| \neq 1, \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

and the vector $\Upsilon(\sigma) \in \Re^m$ by

$$[\Upsilon(\sigma)]_i = \begin{cases} \frac{g_i(\sigma)}{\sigma_i}, & \text{if } \sigma_i \neq 0, \\ (g'(\sigma))_{ii}, & \text{otherwise.} \end{cases}$$

Define the two linear operators S and T by

$$S(A) := \frac{1}{2}(A + A^T), \quad T(A) := \frac{1}{2}(A - A^T), \ \forall\, A \in \Re^{m\times m}.$$

Proposition 2.3 *Suppose that $X \in \Re^{m\times n}$ has the SVD as in (8). Then*

(i) *The matrix operator $\Pi_{\mathcal{B}_\eta}$ is F-differentiable at X if and only if σ satisfies $\|\sigma\|_1 < \eta$ or $\|\sigma\|_1 > \eta$ but $k_1(\sigma) = k_2(\sigma)$. In this case, g is F-differential at σ and the derivative of $\Pi_{\mathcal{B}_\eta}$ at X satisfies that for any $H \in \Re^{m\times n}$,*

$$\begin{aligned} \Pi'_{\mathcal{B}_\eta}(X)H &= U\Big[\Omega(\sigma) \circ S(\widetilde{H}_1) + \mathrm{Diag}\big(\mathcal{F}(\sigma)\mathrm{diag}(\widetilde{H}_1)\big) + \Gamma(\sigma) \circ T(\widetilde{H}_1) \ \ \mathrm{Diag}(\Upsilon(\sigma))\widetilde{H}_2\Big]V^T \\ &= U\Big[\Omega(\sigma) \circ S(\widetilde{H}_1) + \mathrm{Diag}\big(\mathcal{F}(\sigma)\mathrm{diag}(\widetilde{H}_1)\big) + \Gamma(\sigma) \circ T(\widetilde{H}_1)\Big]V_1^T \\ &\quad + U\mathrm{Diag}(\Upsilon(\sigma))U^T H_2(V_2V_2^T) \end{aligned}$$

where $\widetilde{H}_1 \in \Re^{m\times m}$, $\widetilde{H}_2 \in \Re^{m\times(n-m)}$ and $[\widetilde{H}_1 \ \ \widetilde{H}_2] = U^T HV$; $V_1 \in \Re^{m\times m}$, $V_2 \in \Re^{m\times n-m}$ and $V = [V_1 \ \ V_2]$.

(ii) $\Pi_{\mathcal{B}_\eta}$ *is strongly G-semismooth everywhere on* $\Re^{m\times n}$.

Proof: By equation (10), it can be easily verified that $\Pi_{\mathbb{B}_\eta}(\cdot)$ is F-differentiable at X if and only if $\|\sigma\|_1 < \eta$ or $\|\sigma\|_1 > \eta$ but $k_1(\sigma) = k_2(\sigma)$. Then invoking [6, Theorem 3.6], one can establish part (i) of the assertion. Moreover, since $\Pi_{\mathbb{B}_\eta}(\cdot)$ is a piecewise linear function, we know $\Pi_{\mathbb{B}_\eta}(\cdot)$ is strongly G-semismooth. This, together with [6, Theorem 3.12], proves part (ii) of the assertion. □

Proposition 2.4 (c.f. [22]) *Suppose that* $X \in \Re^{m\times n}$ *has the SVD as in* (8). *Let* r *be the number of positive singular values of* X. *Then the orthogonal matrices* P *and* W *satisfy*

$$P[\mathrm{Diag}(\sigma)\ 0] = [\mathrm{Diag}(\sigma)\ 0]W$$

if and only if there exist orthogonal matrices $Q \in \Re^{r\times r}$, $Q' \in \Re^{(m-r)\times(m-r)}$ *and* $Q'' \in \Re^{(n-r)\times(n-r)}$ *such that*

$$P = \begin{bmatrix} Q & 0 \\ 0 & Q' \end{bmatrix} \quad \text{and} \quad W = \begin{bmatrix} Q & 0 \\ 0 & Q'' \end{bmatrix}.$$

3 A dual proximal point algorithm framework

In this section, we shall introduce the framework of the inexact dual PPA for solving the MNA problem and establish its global and local convergence.

3.1 Proximal point algorithm

In various fields of applied mathematics, many problems can be equivalently formulated as a maximal monotone inclusion problem, that is, given a, possibly multi-valued, maximal monotone operator $\mathcal{T} : \mathcal{X} \to 2^{\mathcal{X}}$ where $\mathcal{X}$ is a finite dimensional real Euclidean space, it is to find an $x \in \mathcal{X}$ such that $0 \in \mathcal{T}(x)$. The proximal point algorithm (PPA) [33] applied to the maximal monotone inclusion problem takes the following scheme

$$x^{k+1} \approx p_{\lambda_k}(x^k) := (I + \lambda_k \mathcal{T})^{-1}(x^k),$$

where the parameter $\lambda_k > 0$ is bounded away from zero. Rockafellar [33] suggested computing x^{k+1} only approximately to satisfy the following accuracy criteria:

$$\|x^{k+1} - p_{\lambda_k}(x^k)\| \le \varepsilon_k, \quad \varepsilon_k > 0, \quad \textstyle\sum_{k=1}^{\infty} \varepsilon_k < \infty, \tag{14}$$

$$\|x^{k+1} - p_{\lambda_k}(x^k)\| \le \delta_k \|x^{k+1} - x^k\|, \quad \delta_k > 0, \quad \textstyle\sum_{k=1}^{\infty} \delta_k < \infty. \tag{15}$$

In [33], he showed that the sequence generated above converges (in the weak topology) to a zero point of $\mathcal{T}$, if it exists. Moreover, if $\lambda_k \uparrow \lambda_\infty \le \infty$ and $\mathcal{T}^{-1}$ is Lipschitz continuous at 0, then condition (15) ensures that the local convergence is linear and the rate is approximately proportional to $1/\lambda_\infty$. If in addition $\lambda_\infty = \infty$, then the convergence becomes superlinear.

Possibly due to its versatility and effectiveness, the PPA has received continuous attention from numerous researchers and is well accepted as a powerful tool for solving various classes of convex optimization problems, see, e.g. [14,25,31,40,45]. In this section, we consider the dual PPA, i.e., applying the idea to the maximal monotone operator associated with the dual problem. By rewriting (1) as

$$\min\Big\{\|X\|_2 \;\mid\; \mathcal{A}^*y + X = A_0,\; By - b \in \mathcal{Q}\Big\}, \tag{16}$$

we can easily derive the following explicit form of its dual

$$\min\Big\{ -\langle A_0, Z\rangle - \langle b, w\rangle \mid \mathcal{A}Z + B^T w = 0, \quad \|Z\|_* \le 1, \quad w \in \mathcal{Q}^*\Big\}, \tag{17}$$

where $\mathcal{Q}^*$ is the dual cone of $\mathcal{Q}$. For the convergence analysis later, we assume that the Slater condition for (17) holds, i.e., there exists $(Z, w) \in \Re^{m\times n} \times \Re^{n_1+n_2}$ such that

$$\mathcal{A}Z + B^T w = 0, \quad \|Z\|_* < 1, \quad w_i > 0,\; i = n_1+1, \ldots, n_1+n_2. \tag{18}$$

If in addition, the Slater condition also holds for (16), i.e., $\exists\, \hat{y}$ such that $B\hat{y} - b \in \mathrm{relint}(\mathcal{Q})$, where relint denotes the relative interior, then strong duality holds for (16) and (17), and (X, Z, w) is an optimal solution of the primal and dual problems if and only if it is the solution to the following KKT conditions:

$$\begin{gathered}
\mathcal{A}^*y + X = A_0,\; By - b \in \mathcal{Q},\\
\mathcal{A}Z + B^T w = 0,\; \|Z\|_* \le 1,\; w \in \mathcal{Q}^*,\\
\langle X,\, Z\rangle - \|X\|_2 = 0, \quad \langle By - b,\, w\rangle = 0.
\end{gathered}$$

Write

$$\mathcal{T}_l(X, y) = -\begin{pmatrix} \partial\|X\|_2 \\ 0 \end{pmatrix} + \partial\chi_{\mathcal{F}_1}(X, y), \quad \forall\, X \in \Re^{m\times n},\, y \in \Re^p,$$

$$\mathcal{T}_g(Z, w) = -\begin{pmatrix} A_0 \\ b \end{pmatrix} + \partial\chi_{\mathcal{F}_2}(Z, w), \quad \forall\, Z \in \Re^{m\times n},\, w \in \Re^{n_1+n_2},$$

and

$$p_\lambda(Z, w) = (I + \lambda\mathcal{T}_g)^{-1}(Z, w), \quad \forall\, Z \in \Re^{m\times n},\, w \in \Re^{n_1+n_2},$$

where $\mathcal{F}_1$ and $\mathcal{F}_2$ are the feasible sets of (16) and (17) respectively.

For any given $Z^k \in \Re^{m\times n}$, $w^k \in \Re^{n_1+n_2}$ and $\lambda_k > 0$, it is easy to see that $p_{\lambda_k}(Z^k, w^k)$ is the unique optimal solution to the following minimization problem

$$\begin{aligned}
\min \quad & -\langle A_0, Z\rangle - \langle b, w\rangle + \frac{1}{2\lambda_k}\|Z - Z^k\|^2 + \frac{1}{2\lambda_k}\|w - w^k\|^2\\
\text{s.t.} \quad & \mathcal{A}Z + B^T w = 0, \quad \|Z\|_* \le 1, \quad w \in \mathcal{Q}^*.
\end{aligned} \tag{19}$$

We can show that the dual of (19) takes the following form

$$\begin{aligned}
\max_{y\in\Re^p} \theta_k(y) := {} & \frac{1}{2\lambda_k}\|\Pi_{\mathcal{B}}(Z^k - \lambda_k(\mathcal{A}^*y - A_0)) - (Z^k - \lambda_k(\mathcal{A}^*y - A_0))\|^2 \\
& + \frac{1}{2\lambda_k}\Big(\|Z^k\|^2 - \|Z^k - \lambda_k(\mathcal{A}^*y - A_0)\|^2\Big) + \frac{1}{2\lambda_k}\Big(\|w^k\|^2 - \|\Pi_{\mathcal{Q}^*}[w^k - \lambda_k(By - b)]\|^2\Big).
\end{aligned} \tag{20}$$

Clearly, the Slater condition (18) asserts that the optimal solution set of (20) is nonempty. Let y^{k+1} be a minimizer of (20). From the relationship between the primal and dual variables, we have

$$p_{\lambda_k}(Z^k, w^k) = \begin{bmatrix} \Pi_{\mathcal{B}}(Z^k - \lambda_k(\mathcal{A}^* y^{k+1} - A_0)) \\ \Pi_{\mathcal{Q}^*}(w^k - \lambda_k(By^{k+1} - b)) \end{bmatrix}. \tag{21}$$

Therefore, to implement the PPA, one needs to solve (20) and then update the variable (Z, w) by

$$(Z^{k+1}, w^{k+1}) \approx p_{\lambda_k}(Z^k, w^k).$$

In view of (21), we are able to present an inexact dual PPA framework:

Algorithm 3.1 (A dual PPA framework) *Given (Z^0, w^0, y^0) and $\lambda_0 > 0$, at the k-th iteration, do the following steps:*

Step 1. *For fixed Z^k, w^k and y^k, compute an approximate maximizer*

$$y^{k+1} \approx \arg\max\{\theta_k(y) \mid y \in \Re^p\}$$

where θ_k is defined in (20).

Step 2. *Update the variables Z^{k+1}, X^{k+1} and w^{k+1} via*

$$Z^{k+1} = \Pi_{\mathcal{B}}(Z^k - \lambda_k(\mathcal{A}^* y^{k+1} - A_0)), \quad w^{k+1} = \Pi_{\mathcal{Q}^*}(w^k - \lambda_k(By^{k+1} - b)),$$
$$X^{k+1} = (Z^k - \lambda_k(\mathcal{A}^* y^{k+1} - A_0) - Z^{k+1})/\lambda_k.$$

Step 3. *If $\max\{\|A_0 - \mathcal{A}^* y^{k+1} - X^{k+1}\|, \|\Pi_{\mathcal{Q}^*}(b - By^{k+1})\|\} \le \varepsilon$, stop; else, update λ_k to λ_{k+1}, end.*

3.2 Convergence analysis

In Step 1 of the dual PPA, we use the following stopping criteria suggested in [31],

$$\max \theta_k(y) - \theta_k(y^{k+1}) \le \tfrac{\varepsilon_k^2}{2\lambda_k}, \quad \varepsilon_k > 0, \quad \textstyle\sum_{k=1}^{\infty} \varepsilon_k < \infty, \tag{22}$$

$$\max \theta_k(y) - \theta_k(y^{k+1}) \le \tfrac{\delta_k^2}{2\lambda_k}(\|Z^{k+1} - Z^k\|^2 + \|w^{k+1} - w^k\|^2),\ \delta_k > 0,\ \textstyle\sum_{k=1}^{\infty} \delta_k < \infty, \tag{23}$$

$$\|\nabla_y \theta_k(y^{k+1})\| \le \frac{\delta_k'}{\lambda_k} \left\| \begin{pmatrix} Z^{k+1} - Z^k \\ w^{k+1} - w^k \end{pmatrix} \right\|,\ 0 \le \delta_k' \to 0. \tag{24}$$

Next we present two results on the global and local convergence of the dual PPA. The proofs are omitted since they follow directly from [31, Theorem 4-5].

Theorem 3.1 (Global Convergence) *Let the inexact PPA be executed with stopping criterion* (22)*. Suppose that the primal problem* (16) *satisfies the Slater condition. Then the sequence* $\{(Z^{k+1}, w^{k+1})\} \subset \mathcal{B} \times \mathcal{Q}^*$ *generated by the inexact PPA is bounded and it converges to an optimal solution of* (17)*. Moreover, the sequence* $\{y^k\}$ *is also bounded and any of its accumulation point is an optimal solution of* (1)*.*

Theorem 3.2 (Local Convergence) *Let the dual PPA be executed with stopping criteria* (22) *and* (23)*. Suppose that the Slater condition holds for* (16)*. If* $\mathcal{T}_g^{-1}$ *is Lipschitz continuous at the origin with the modulus* a_g*, then* $\{(Z^{k+1}, w^{k+1})\}$ *converges to an optimal solution* $(\overline{Z}, \bar{w})$ *of* (17)*, and*

$$\left\| \begin{pmatrix} Z^{k+1} - \overline{Z} \\ w^{k+1} - \bar{w} \end{pmatrix} \right\| \le \nu_k \left\| \begin{pmatrix} Z^k - \overline{Z} \\ w^k - \bar{w} \end{pmatrix} \right\|, \quad \text{for all } k \text{ sufficiently large,} \tag{25}$$

where $\nu_k = [a_g(a_g + \lambda_k^2)^{-1/2} + \delta_k](1 - \delta_k)^{-1} \to a_g(a_g^2 + \lambda_\infty^2)^{-1/2} < 1$*. Moreover, the conclusion about* $\{y^k\}$ *in Theorem 3.1 is valid.*

If in addition to (23) *and the condition on* $\mathcal{T}_g^{-1}$*, one also has* (24) *and that* $\mathcal{T}_l^{-1}$ *is Lipschitz continuous at the origin with modulus* a_l $(\ge a_g)$*, then* $\{y^{k+1}\}$ *converges to the unique optimal solution* $\bar{y}$ *of* (1)*, and*

$$\left\| \begin{pmatrix} X^{k+1} - \overline{X} \\ y^{k+1} - \bar{y} \end{pmatrix} \right\| \le \nu_k' \left\| \begin{pmatrix} Z^{k+1} - Z^k \\ w^{k+1} - w^k \end{pmatrix} \right\|, \quad \text{for all } k \text{ sufficiently large,}$$

where $\overline{X} = A_0 - \mathcal{A}^*\bar{y}$*, and* $\nu_k' = a_l(1 + \delta_k')/\lambda_k \to a_l/\lambda_\infty$*.*

4 A semismooth Newton-CG method for the inner problem

In this section, we will apply the well-known inexact semismooth Newton method to approximately solve the unconstrained subproblem (20). Using Proposition 2.1 (i), we know that the first order optimality condition for (20) is given by

$$0 = \nabla\theta_k(y) := \mathcal{A}\Pi_{\mathcal{B}}\big[Z^k - \lambda_k(\mathcal{A}^*y - A_0)\big] + B^T\Pi_{\mathcal{Q}^*}\big[w^k - \lambda_k(By - b)\big].$$

Since $\Pi_{\mathcal{B}}(\cdot)$ and $\Pi_{\mathcal{Q}^*}(\cdot)$ are Lipschitz continuous, $\nabla\theta_k(\cdot)$ is also Lipschitz continuous. Hence Clarke's generalized Jacobian of $\nabla\theta_k$ (which is the generalized Hessian of θ_k and we denote it by $\partial^2\theta_k$) is well defined. Since it is difficult to derive an exact characterization of $\partial^2\theta_k$, we will slightly modify the classical semismooth Newton method by selecting elements in $\hat{\partial}^2\theta_k$ instead of $\partial^2\theta_k$, where $\hat{\partial}^2\theta_k$ is a set-valued mapping defined by

$$\hat{\partial}^2\theta_k(y) := -\lambda_k\big[\mathcal{A}\partial\Pi_{\mathcal{B}}(Z^k - \lambda_k(\mathcal{A}^*y - A_0))\mathcal{A}^* + B^T\partial\Pi_{\mathcal{Q}^*}(w^k - \lambda_k(By - b))B\big], \ y \in \Re^p.$$

Note that by [4, p.75]

$$\partial^2\theta_k(y)h \subseteq \hat{\partial}^2\theta_k(y)h \quad \forall\, h \in \Re^p.$$

Actually we even know from [19, Example 2.5] that

$$\partial^2\theta_k(y)h = \hat{\partial}^2\theta_k(y)h \quad \forall\, h \in \Re^p,$$

which, however, does not mean $\partial^2\theta_k(y) = \hat{\partial}^2\theta_k(y)$.

4.1 Characterization of $\hat{\partial}^2\theta_k$

To obtain the explicit expression of $\hat{\partial}^2\theta_k$, it suffices to characterize $\partial\Pi_{\mathcal{B}}(\cdot)$ and $\partial\Pi_{\mathcal{Q}^*}(\cdot)$. For a given $Y \in \Re^{m\times n}$, suppose that it has the following SVD:

$$Y = U[\mathrm{Diag}(\sigma)\ 0]V^T, \tag{26}$$

where $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_m)^T$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0 = \sigma_{r+1} = \ldots = \sigma_m$. When $\|Y\|_* = 1$, we know from direct calculation that $k_1(\sigma) = r$ and $k_2(\sigma) = m$. For any positive integer $N \in [r, m]$, we define the following four index sets:

$$\begin{array}{ll} \alpha_1 := \{1, 2, \ldots, r\}, & \alpha_2 := \{r+1, r+2, \ldots, N\}, \\ \alpha_3 := \{N+1, N+2, \ldots, m\}, & \alpha_4 := \{m+1, m+2, \ldots, n\} \end{array} \tag{27}$$

In another case that $\|Y\|_* > 1$, it is easily seen that $k_1(\sigma) \leq k_2(\sigma) \leq r$. Given a integer $N \in [k_1(\sigma),\ k_2(\sigma)]$, we partition the set $\{1, 2, \ldots, m\}$ into the following five subsets:

$$\begin{array}{ll} \beta_1 := \{1, 2, \ldots, k_1(\sigma)\}, & \beta_2 := \{k_1(\sigma)+1, k_1(\sigma)+2, \ldots, N\}, \\ \beta_3 := \{N+1, N+2, \ldots, k_2(\sigma)\}, & \beta_4 := \{k_2(\sigma)+1, k_2(\sigma)+2, \ldots, r\}, \\ \beta_5 := \{r+1, r+2, \ldots, m\}, & \end{array}$$

and write

$$\gamma_1 := \beta_1 \cup \beta_2, \quad \gamma_2 := \beta_3 \cup \beta_4 \cup \beta_5.$$

Also, for the sake of simplicity, we will drop the parameter σ in $\Omega(\sigma), \Gamma(\sigma), F(\sigma)$ and $\Upsilon(\sigma)$, and represent the notations by $\Omega, \Gamma, \mathcal{F}$ and Υ, respectively.

Proposition 4.1 *Let $Y \in \Re^{m\times n}$ admit the SVD as in (26).*

(i) *If $\|Y\|_* < 1$, then $\partial_{\boldsymbol{B}}\Pi_{\mathcal{B}}(Y)$ is a singleton set consisting of the identity operator from $\Re^{m\times n}$ to itself.*

(ii) *If $\|Y\|_* = 1$, for $\mathcal{V} \in \partial_{\boldsymbol{B}}\Pi_{\mathcal{B}}(Y)$, either $\mathcal{V}$ is the identity operator or there exist an integer $N \in [r, m]$, $(\Omega^\infty_{\alpha_2\alpha_3}, \Gamma^\infty_{\alpha_2\alpha_2}, \Gamma^\infty_{\alpha_2\alpha_3}, \Upsilon^\infty_{\alpha_2}) \in \mathcal{S}_N$ and singular vector matrices U^∞, V^∞ of Y such that for any $H \in \Re^{m\times n}$,*

$$\mathcal{V}H = U^\infty \left[W^\infty - \frac{\mathrm{Tr}(\widetilde{H}_{11})}{N} \begin{bmatrix} I_N & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathrm{Diag}\begin{bmatrix} \mathbf{1}_r \\ \Upsilon^\infty_{\alpha_2} \\ 0 \end{bmatrix} \widetilde{H}_2 \right] (V^\infty)^T, \tag{28}$$

where the matrix $W^\infty \in \Re^{m\times m}$ is defined by

$$W^\infty := \begin{bmatrix} \mathbf{1}_{N\times N} & \mathbf{1}_{r\times(m-N)} \\ \mathbf{1}_{(m-N)\times r}\ (\Omega^\infty_{\alpha_2\alpha_3})^T & \Omega^\infty_{\alpha_2\alpha_3} \\ & 0 \end{bmatrix} \circ S(\widetilde{H}_1) + \begin{bmatrix} \mathbf{1}_{r\times r} & \mathbf{1}_{r\times(m-r)} \\ \mathbf{1}_{(m-r)\times r} & \Gamma^\infty_{\alpha_2\alpha_2}\ \ \Gamma^\infty_{\alpha_2\alpha_3} \\ & (\Gamma^\infty_{\alpha_2\alpha_3})^T\ \ 0 \end{bmatrix} \circ T(\widetilde{H}_1),$$

with $\widetilde{H}_1 \in \Re^{m\times m}, \widetilde{H}_2 \in \Re^{m\times(n-m)}$, $[\widetilde{H}_1\ \widetilde{H}_2] = (U^\infty)^T H V^\infty$ *and* $\widetilde{H}_{11}$ *is the matrix extracted from the first* N *columns and rows of* $\widetilde{H}_1$, *and* $\mathcal{S}_N$ *is the subset of* $\Re^{(N-r)\times(m-N)} \times \Re^{(N-r)\times(N-r)} \times \Re^{(N-r)\times(m-N)} \times \Re^{N-r}$ *defined by*

$$\mathcal{S}_N := \left\{ \lim_{\varepsilon_N \ni Y^j \to Y} \left(\Omega(\sigma(Y^j))_{\alpha_2\alpha_3}, \Gamma(\sigma(Y^j))_{\alpha_2\alpha_2}, \Gamma(\sigma(Y^j))_{\alpha_2\alpha_3}, \Upsilon(\sigma(Y^j))_{\alpha_2} \right) \right\}$$

with $\varepsilon_N = \{Z \mid Z \in \mathcal{D}_{\Pi_{\mathcal{B}}}, \|Z\|_* > 1,\ k_1(\sigma(Z)) = N\}$.

(iii) *If* $\|Y\|_* > 1$, *for* $\mathcal{V} \in \partial_{\boldsymbol{B}} \Pi_{\mathcal{B}}(Y)$, *there exist an integer* $N \in [k_1(\sigma),\ k_2(\sigma)]$, $\Omega^\infty_{\beta_2\beta_3} \in \mathcal{T}_N$ *and singular vector matrices* U^∞, V^∞ *of* Y *such that for any* $H \in \Re^{m\times n}$,

$$\mathcal{V}H = U^\infty \left[W^\infty - \frac{\mathrm{Tr}(\widetilde{H}_{11})}{N} \begin{bmatrix} I_N & 0 \\ 0 & 0 \end{bmatrix},\quad \mathrm{Diag}\begin{bmatrix} \Upsilon_{\gamma_1} \\ 0 \end{bmatrix} \widetilde{H}_2 \right] (V^\infty)^T, \tag{29}$$

where the matrix $W^\infty \in \Re^{m\times m}$ *is defined by*

$$W^\infty = \begin{bmatrix} \boldsymbol{1}_{N\times N} & \Omega_{\beta_1\beta_3} & \Omega_{\gamma_1(\beta_4\cup\beta_5)} \\ (\Omega_{\beta_1\beta_3})^T\ (\Omega^\infty_{\beta_2\beta_3})^T & \Omega^\infty_{\beta_2\beta_3} & \\ (\Omega_{\gamma_1(\beta_4\cup\beta_5)})^T & & 0 \end{bmatrix} \circ S(\widetilde{H}_1) + \begin{bmatrix} \Gamma_{\gamma_1\gamma_1} & \Gamma_{\gamma_1\gamma_2} \\ (\Gamma_{\gamma_1\gamma_2})^T & 0 \end{bmatrix} \circ T(\widetilde{H}_1),$$

here, $\widetilde{H}_1 \in \Re^{m\times m}, \widetilde{H}_2 \in \Re^{m\times(n-m)}$, $[\widetilde{H}_1\ \widetilde{H}_2] = (U^\infty)^T H V^\infty$ *and* $\widetilde{H}_{11}$ *is the matrix extracted from the first* N *columns and rows of* $\widetilde{H}_1$, *and* $\mathcal{S}_N$ *is the subset of* $\Re^{(N-k_1(\sigma))\times(k_2(\sigma)-N)}$ *defined by*

$$\mathcal{T}_N := \left\{ \lim_{\epsilon_N \ni Y^j \to Y} \Omega(\sigma(Y^j))_{\beta_2\beta_3} \right\},$$

with $\epsilon_N = \{Z \mid Z \in \mathcal{D}_{\Pi_{\mathcal{B}}},\ k_1(\sigma(Z)) = N\}$.

Proof: See the Appendix. □

The characterization of $\partial\Pi_{\mathcal{Q}^*}(\cdot)$ at $z \in \Re^{n_1+n_2}$ is very simple given the special structure of the polyhedral cone $\mathcal{Q}^* = \Re^{n_1} \times \Re^{n_2}_+$. Define the following index sets:

$$\begin{aligned} \mathcal{J}_1 &:= \{i : z_i > 0,\ n_1 + 1 \le i \le n_1 + n_2\} \cup \{1, 2, \ldots, n_1\}, \\ \mathcal{J}_2 &:= \{i : z_i = 0,\ n_1 + 1 \le i \le n_1 + n_2\}, \\ \mathcal{J}_3 &:= \{i : z_i < 0,\ n_1 + 1 \le i \le n_1 + n_2\}. \end{aligned}$$

By direct calculations, we obtain that $\mathcal{V}$ is an element of $\partial\Pi_{\mathcal{Q}^*}(z)$ if and only if there exists a vector $a \in [0,1]^{|\mathcal{J}_2|}$ such that

$$\mathcal{V}h = \begin{bmatrix} h_{\mathcal{J}_1} \\ a \circ h_{\mathcal{J}_2} \\ 0 \end{bmatrix},\ \forall\, h \in \Re^{n_1+n_2}. \tag{30}$$

Remark 4.1 *In the implementation of our semismooth Newton-CG method, we need to select an element* $\mathcal{V}_1^0 \in \partial\Pi_{\mathcal{B}}(Y)$ *and an element* $\mathcal{V}_2^0 \in \partial\Pi_{\mathcal{Q}^*}(z)$. *If* $\|Y\|_* \le 1$, $\mathcal{V}_1^0$ *is chosen as the identity operator from* $\Re^{m\times n}$ *to* $\Re^{m\times n}$. *For the case where* Y *is outside of* $\mathcal{B}$, *we take* $U^\infty = U$, $V^\infty = V$ *and* $N = k_1(\sigma)$ *in* (29). *Thus* $\beta_2 = \emptyset$ *and for any* $H \in \Re^{m\times n}$,

$$\mathcal{V}_1^0 H = U^\infty \left[W^\infty - \frac{\mathrm{Tr}(\widetilde{H}_{11})}{N} \begin{bmatrix} I_N & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathrm{Diag}\begin{bmatrix} \Upsilon_{\gamma_1} \\ 0 \end{bmatrix} \widetilde{H}_2 \right] (V^\infty)^T, \tag{31}$$

where the matrix $W^\infty \in \Re^{m\times m}$ *is defined by*

$$W^\infty = \begin{bmatrix} \mathbf{1}_{N\times N} & \Omega_{\gamma_1\gamma_2} \\ (\Omega_{\gamma_1\gamma_2})^T & 0 \end{bmatrix} \circ S(\widetilde{H}_1) + \begin{bmatrix} \Gamma_{\gamma_1\gamma_1} & \Gamma_{\gamma_1\gamma_2} \\ (\Gamma_{\gamma_1\gamma_2})^T & 0 \end{bmatrix} \circ T(\widetilde{H}_1),$$

with $\widetilde{H}_1 \in \Re^{m\times m}$, $\widetilde{H}_2 \in \Re^{m\times(n-m)}$, $[\widetilde{H}_1\ \widetilde{H}_2] = U^T H V$ *and* $\widetilde{H}_{11}$ *being the matrix extracted from the first* N *columns and rows of* $\widetilde{H}_1$. *As to the selection of* $\mathcal{V}_2^0$, *we take* $a = 0$ *in* (30) *and then*

$$\mathcal{V}_2^0 h = \begin{bmatrix} h_{\mathcal{J}_1} \\ 0 \end{bmatrix}, \ \forall\, h \in \Re^{n_1+n_2}. \tag{32}$$

4.2 Constraint nondegeneracy

For the convergence analysis of the semismooth Newton-CG method, we need the concept of constraint nondegeneracy which is originally introduced by Robinson [30] and extended by Bonnans and Shapiro [1]. Let $\mathcal{X}$ and $\mathcal{Y}$ be two finite dimensional spaces, $\Phi : \mathcal{X} \to \mathcal{Y}$ be a continuously differentiable function and $\mathcal{C}$ be a closed convex set. We use $T_{\mathcal{C}}(x)$ and $\mathrm{lin}(T_{\mathcal{C}}(x))$ to denote the tangent cone of $\mathcal{C}$ at x and its linearity space, respectively. A feasible point $\bar{x}$ to the feasibility problem $\{\Phi(x) \in \mathcal{C},\ x \in \mathcal{X}\}$ is said to be constraint nondegenerate if

$$\Phi'(\bar{x})\mathcal{X} + \mathrm{lin}(T_{\mathcal{C}}(\Phi(\bar{x}))) = \mathcal{Y}.$$

Thus the constraint nondegeneracy condition associated with the minimizer $(\widehat{Z}, \hat{w})$ of (19) has the form

$$\begin{bmatrix} \mathcal{A} & B^T \\ \mathcal{I} & 0 \\ 0 & \mathcal{I} \end{bmatrix} \begin{pmatrix} \Re^{m\times n} \\ \Re^{n_1+n_2} \end{pmatrix} + \begin{bmatrix} \{0\}^p \\ \mathrm{lin}(T_{\mathcal{B}}(\widehat{Z})) \\ \mathrm{lin}(T_{\mathcal{Q}^*}(\hat{w})) \end{bmatrix} = \begin{bmatrix} \Re^p \\ \Re^{m\times n} \\ \Re^{n_1+n_2} \end{bmatrix}, \tag{33}$$

or equivalently,

$$\mathcal{A}\mathrm{lin}(T_{\mathcal{B}}(\widehat{Z})) + B^T \mathrm{lin}(T_{\mathcal{Q}^*}(\hat{w})) = \Re^p. \tag{34}$$

Proposition 4.2 *Let* $(\widehat{Z}, \hat{w})$ *be the unique solution pair of* (19). *Let* $\widehat{Z}$ *have the following SVD:*

$$\widehat{Z} = U[\mathrm{Diag}(\sigma(\widehat{Z}))\ 0]V^T = [U_1\ U_2][\mathrm{Diag}(\sigma(\widehat{Z}))\ 0][V_1\ V_2]^T,$$

where $\sigma_1(\widehat{Z}) \geq \ldots \geq \sigma_r(\widehat{Z}) > 0 = \sigma_{r+1}(\widehat{Z}) = \ldots = \sigma_m(\widehat{Z})$, *and* $U_1 \in \Re^{m\times r}, U_2 \in \Re^{m\times(m-r)}, V_1 \in \Re^{n\times r}, V_2 \in \Re^{n\times(n-r)}$. *Define the following two index sets* κ_1 *and* κ_2 *by*

$$\kappa_1 := \{1, 2, \ldots, n_1\} \cup \{i |\, \hat{w}_i > 0,\, n_1 + 1 \leq i \leq n_1 + n_2\},$$
$$\kappa_2 := \{i |\, \hat{w}_i = 0,\, n_1 + 1 \leq i \leq n_1 + n_2\}.$$

Then it holds that:

(i) *if* $\|\widehat{Z}\|_* < 1$, *the constraint nondegeneracy holds at* $(\widehat{Z}, \hat{w})$ *if and only if*

$$B_{\kappa_1} y = 0,\ \mathcal{A}^* y = 0 \implies y = 0. \tag{35}$$

(ii) *if* $\|\widehat{Z}\|_* = 1$, *the constraint nondegeneracy holds at* $(\widehat{Z}, \hat{w})$ *if and only if, for any given* $k \in \Re$,

$$\left\{ \begin{array}{l} B_{\kappa_1} y = 0,\ (U_1)^T(\mathcal{A}^* y)V_1 = kI_r, \\ (U_1)^T(\mathcal{A}^* y)V_2 = 0,\ (U_2)^T(\mathcal{A}^* y)V_1 = 0 \end{array} \right. \Longrightarrow y = 0. \tag{36}$$

Proof: (i) Under the condition that $\|\widehat{Z}\|_* < 1$, it is easy to see that

$$\mathrm{lin}(T_{\mathcal{B}}(\widehat{Z})) = \Re^{m\times n} \quad \text{and} \quad \mathrm{lin}(T_{\mathcal{Q}^*}(\hat{w})) = \left[\begin{array}{c} \Re^{|\kappa_1|} \\ \{0\}^{|\kappa_2|} \end{array} \right].$$

Thus the constraint nondegeneracy condition (34) is reduced to

$$\mathcal{A}\Re^{m\times n} + B_{\kappa_1}^T \Re^{|\kappa_1|} = \Re^p, \tag{37}$$

which is equivalent to (35).

(ii) Since

$$\mathrm{lin}(T_{\mathcal{Q}^*}(\hat{w})) = \left[\begin{array}{c} \Re^{|\kappa_1|} \\ \{0\}^{|\kappa_2|} \end{array} \right],$$

the constraint nondegeneracy condition (34) is reduced to

$$\mathcal{A}\mathrm{lin}(T_{\mathcal{B}}(\widehat{Z})) + B_{\kappa_1}^T \Re^{|\kappa_1|} = \Re^p, \tag{38}$$

which is equivalent to

$$B_{\kappa_1} y = 0,\ \mathcal{A}^* y \in \mathrm{lin}(T_{\mathcal{B}}(\widehat{Z}))^\perp \implies y = 0. \tag{39}$$

By [4, Proposition 2.3.6,Theorem 2.4.9] and [41, Theorem 1], one can establish that

$$T_{\mathcal{B}}(\widehat{Z}) = \{H \in \Re^{m\times n} |\, \mathrm{Tr}((U_1)^T H V_1) + \|(U_2)^T H V_2\|_* \leq 0\}.$$

Therefore, $\mathrm{lin}(T_{\mathcal{B}}(\widehat{Z})) = \{H \in \Re^{m\times n} |\, HV_1 \in (U_1)^\perp,\ (U_2)^T H V_2 = 0\}$. A simple calculation shows that

$$\mathrm{lin}(T_{\mathcal{B}}(\widehat{Z}))^\perp = \{Y \in \Re^{m\times n} |\, \exists k \in \Re,\ (U_1)^T Y V_1 = kI_r, (U_1)^T Y V_2 = 0, (U_2)^T Y V_1 = 0\},$$

which, together with (39), completes the proof. □

With the above proposition, we next establish a result which exploits the close relationship between the constraint nondegeneracy of the optimal solution of (19) and the negative definiteness of the elements of $\hat{\partial}^2\theta_k$.

Proposition 4.3 *Suppose that the problem* (19) *satisfies the Slater condition* (18). *Let* $(\widehat{Z}, \hat{w})$ *and* $\hat{y}$ *denote, respectively, the optimal solutions of* (19) *and* (20). *Let* $Y := Z^k - \lambda_k(\mathcal{A}^*\hat{y} - A_0)$. *Then the following conditions are equivalent:*

(i) *The constraint nondegeneracy condition* (34) *holds at* $(\widehat{Z}, \hat{w})$.

(ii) *Every element in* $\hat{\partial}^2\theta_k(\hat{y})$ *is symmetric and negative definite.*

(iii) *The operator*

$$\mathcal{V}_0 = -\lambda_k(\mathcal{A}\mathcal{V}_1^1\mathcal{A}^* + B^T\mathcal{V}_2^0 B)$$

is symmetric and negative definite, where $\mathcal{V}_1^1$ *is the same as* $\mathcal{V}_1^0$ *in* (31) *except when in the case of* $\|\widehat{Y}\|_* = 1$, *the operator is defined by* (28) *with* $N = \mathrm{rank}(Y)$.

Proof: Assume that the SVD of $\widehat{Z}$ and the index sets κ_1, κ_2 are given in Proposition 4.2.

"(i) $\Rightarrow$ (ii)". Let $\mathcal{V}$ be an arbitrary element of $\hat{\partial}^2\theta_k(\hat{y})$. Then there exist $\mathcal{V}_1$ and $\mathcal{V}_2$ in $\partial\Pi_{\mathcal{B}}[Z^k - \lambda_k(\mathcal{A}^*\hat{y} - A_0)]$ and $\partial\Pi_{\mathcal{Q}^*}[w^k - \lambda_k(B\hat{y} - b)]$, respectively, such that

$$\mathcal{V} = -\lambda_k[\mathcal{A}\mathcal{V}_1\mathcal{A}^* + B^T\mathcal{V}_2 B].$$

Therefore, $\mathcal{V}$ is self-adjoint. Moreover, it follows from [26, Proposition 1] that for any $h \in \Re^p$,

$$\begin{aligned}\langle h,\, \mathcal{V}h\rangle &= -\lambda_k\langle h,\, \mathcal{A}\mathcal{V}_1\mathcal{A}^*h\rangle - \lambda_k\langle h,\, B^T\mathcal{V}_2 Bh\rangle \;=\; -\lambda_k\langle \mathcal{A}^*h,\, \mathcal{V}_1\mathcal{A}^*h\rangle - \lambda_k\langle Bh,\, \mathcal{V}_2 Bh\rangle \\ &\le -\lambda_k\langle \mathcal{V}_1\mathcal{A}^*h,\, \mathcal{V}_1\mathcal{A}^*h\rangle - \lambda_k\langle \mathcal{V}_2 Bh,\, \mathcal{V}_2 Bh\rangle \;\le\; 0,\end{aligned}$$

which implies that $\mathcal{V}$ is negative semidefinite. To complete the proof of this part, it suffices to show that $\mathcal{V}$ is nonsingular. Consider the following linear system

$$\mathcal{V}h = 0, \quad \text{or equivalently,} \quad \mathcal{V}_1\mathcal{A}^*h = 0,\ \mathcal{V}_2 Bh = 0. \tag{40}$$

Now we proceed to prove that $h = 0$ by considering the following two cases.

Case 1: $\|Y\|_* < 1$. In this case, since $\widehat{Z} = \Pi_{\mathcal{B}}(Y)$, we have $\|\widehat{Z}\|_* < 1$ and $\mathcal{V}_1 = \mathcal{I}$. Then, it follows from (40) that

$$\mathcal{A}^*h = 0,\ B_{\kappa_1}h = 0, \tag{41}$$

which, together with the constraint nondegeneracy assumption (35), implies $h = 0$.

Case 2: $\|Y\|_* \ge 1$ and thus $\|\widehat{Z}\|_* = 1$. We first show the nonsingularity of $\mathcal{V}$ for the choice that $\mathcal{V}_1 = \mathcal{I}$. In this situation, (41) still holds and hence by taking $k = 0$ in (36), we know that $\mathcal{V}$ is negative definite. Next, we turn to the case in which $\mathcal{V}_1$ is another element selected from $\partial_{\mathbf{B}}\Pi_{\mathcal{B}}(Y)$. We consider two sub-cases.

Case 2.1: $\|Y\|_* = 1$. Let $H = \mathcal{A}^* h$. In view of the analysis in the previous subsection, we know from $\mathcal{V}_1 H = 0$ that

$$\begin{aligned} 0 &= U^\infty \left[\left[\begin{array}{cc} \mathbf{1}_{N\times N} & \mathbf{1}_{r\times(m-N)} \\ \mathbf{1}_{(m-N)\times r} & \Omega^\infty_{\alpha_2\alpha_3} \\ (\Omega^\infty_{\alpha_2\alpha_3})^T & 0 \end{array} \right] \circ S(\widetilde{H}_1) \quad \left[\begin{array}{c} \mathbf{1}_{r\times(n-m)} \\ \Upsilon^\infty_{\alpha_2} \mathbf{1}^T_{n-m} \\ 0 \end{array} \right] \circ \widetilde{H}_2 \right] (V^\infty)^T \\ &+ U^\infty \left(\left[\begin{array}{ccc} \mathbf{1}_{r\times r} & \mathbf{1}_{r\times(m-r)} & \\ \mathbf{1}_{(m-r)\times r} & \Gamma^\infty_{\alpha_2\alpha_2} & \Gamma^\infty_{\alpha_2\alpha_3} \\ & (\Gamma^\infty_{\alpha_2\alpha_3})^T & 0 \end{array} \right] \circ T(\widetilde{H}_1) - \frac{\mathrm{Tr}(\widetilde{H}_{11})}{N} \left[\begin{array}{cc} I_N & 0 \\ 0 & 0 \end{array} \right] \right) (V_1^\infty)^T, \end{aligned}$$

where $V_1^\infty \in \Re^{m\times m}$, $V_2^\infty \in \Re^{m\times(n-m)}$ and $V^\infty := [V_1^\infty \; V_2^\infty]$, the index sets $\alpha_1, \alpha_2, \alpha_3$ and α_4 are defined as in (27). This implies that

$$(U^\infty_{\alpha_1})^T(H)V^\infty_{\alpha_1} = \frac{1}{N} I_r, \quad (U^\infty_{\alpha_1})^T(H)V^\infty_{\alpha_2\cup\alpha_3\cup\alpha_4} = 0, \quad (U^\infty_{\alpha_2\cup\alpha_3})^T(H)V^\infty_{\alpha_1} = 0. \tag{42}$$

By Proposition 2.4, there exist orthogonal matrices $Q \in \Re^{r\times r}$, $Q' \in \Re^{(m-r)\times(m-r)}$ and $Q'' \in \Re^{(n-r)\times(n-r)}$ such that

$$U^\infty_{\alpha_1} = U_{\alpha_1} Q, \quad U^\infty_{\alpha_2\cup\alpha_3} = U_{\alpha_2\cup\alpha_3} Q', \quad V^\infty_{\alpha_1} = Q V_{\alpha_1}, \quad V^\infty_{\alpha_2\cup\alpha_3\cup\alpha_4} = Q'' V_{\alpha_2\cup\alpha_3\cup\alpha_4}. \tag{43}$$

Moreover, we know from $\mathcal{V}_2 h = 0$ that

$$B_{\kappa_1} h = 0. \tag{44}$$

Combing (36), (44), (42) with (43), we deduce $h = 0$ and hence $\mathcal{V}$ is negative definite.

Case 2.2: $\|Y\|_* > 1$. The proof of the negative definiteness of $\mathcal{V}$ is similar to that of Case 2.1, with the equality (29) replacing (28).

By taking the convex hull of $\partial_{\mathbf{B}} \Pi_{\mathcal{B}}(Y)$, we complete the proof of the first part.

"(ii) $\Rightarrow$ (iii)". This is trivial since $\mathcal{V}_0 \in \hat{\partial}^2 \theta_k(\hat{y})$.

"(iii) $\Rightarrow$ (i)". Assume the contrary that the constraint nondegeneracy condition fails to hold at $(\widehat{Z}, \hat{w})$. Again, we consider two cases.
Case 1: $\|Y\|_* < 1$ and hence $\|\widehat{Z}\|_* < 1$. By assumption, there exists a $z \neq 0$ such that

$$\mathcal{A}^* z = 0, \quad B_{\kappa_1} z = 0.$$

This means that $\mathcal{V}_0$ is singular, which contradicts to (iii).
Case 2: $\|Y\|_* \geq 1$ and hence $\|\widehat{Z}\|_* = 1$. By assumption, there exist $k \in \Re$ and $z \neq 0$ such that

$$B_{\kappa_1} z = 0, \quad (U_1)^T(\mathcal{A}^* z)V_1 = k I_r, \quad (U_1)^T(\mathcal{A}^* z)V_2 = 0, \quad (U_2)^T(\mathcal{A}^* z)V_1 = 0. \tag{45}$$

Using the equalities above, simple computation yields $\mathcal{V}_0 z = 0$. This contradicts to the statement (iii). The proof is completed. □

4.3 A semismooth Newton-CG algorithm

In this subsection, we briefly describe the semismooth Newton-CG algorithm for solving (20). The basic template of the algorithm is given as follows. For simplicity, we drop the outer iteration index k.

Algorithm 4.1 (A Semismooth Newton-CG Method)

Step 0. *Given* $\varsigma \in (0, 0.5), \delta_1, \delta_2, \bar{\eta}, \rho \in (0, 1)$ *and* $\tau \in (0, 1]$. *Choose* $y^0 \in \Re^p$.

Step 1. *For* $j = 0, 1, 2, \ldots,$

Step 1.1. *Apply the the preconditioned conjugate gradient (*PCG*) method to find an approximation solution* d^j *to*

$$(\mathcal{V}_j - \epsilon_j I)d = -\nabla\theta_k(y^j), \tag{46}$$

satisfying the residual condition

$$\|(\mathcal{V}_j - \epsilon_j I)d^j + \nabla\theta_k(y^j)\| \le \eta_j := \min\{\bar{\eta}, \|\nabla\theta_k(y^j)\|^{1+\tau}\}. \tag{47}$$

Here $\mathcal{V}_j$ *is an element of* $\hat{\partial}^2\theta_k(y^j)$ *and* $\epsilon_j = \delta_1 \min\{\delta_2, \|\nabla\theta_k(y^j)\|\}$.

Step 1.2. *Let* m_j *be the smallest nonnegative integer* m *satisfying*

$$\theta_k(y^j + \rho^m d^j) - \theta_k(y^j) \ge \varsigma\rho^m\langle\nabla\theta_k(y^j), d^j\rangle.$$

Set $\alpha_j := \rho^{m_j}$ *and* $y^{j+1} := y^j + \alpha_j d^j$.

From the structures of $\mathcal{V}_1^0$ and $\mathcal{V}_2^0$, we know that $\mathcal{V}_j$ is always negative semidefinite. Hence $\mathcal{V}_j - \epsilon_j I$ is always negative definite as long as $\nabla_y\theta_k(y^j) \neq 0$. So, it is reasonable for us to apply the PCG method to solve (46). Furthermore, by noting the strong semismoothness of $\Pi_{\mathcal{B}}(\cdot)$ and $\Pi_{\mathcal{Q}^*}(\cdot)$, and using the proof similar to [45, Theorem 3.4], we can easily derive the following convergence results for Algorithm 4.1.

Theorem 4.1 *Suppose that the Slater condition holds for* (19). *Then the semismooth Newton-CG algorithm* 4.1 *is well defined and any accumulation point* $\hat{y}$ *of* $\{y^j\}$ *generated by algorithm* 4.1 *is an optimal solution to the inner subproblem* (20).

Theorem 4.2 *Assume that the Slater condition holds for* (19). *Let* $\hat{y}$ *be an accumulation point of the infinite sequence* $\{y^j\}$ *generated by the semismooth Newton-CG algorithm for solving* (20). *Suppose that at each step* $j \ge 0$, *the residual condition (47) is satisfied. Assume that the constraint nondegeneracy condition* (34) *holds at* $Z^k - \lambda_k(\mathcal{A}^*(\hat{y}) - A_0)$. *Then the whole sequence* $\{y^j\}$ *converges to* $\hat{y}$ *and*

$$\|y^{j+1} - \hat{y}\| = O(\|y^j - \hat{y}\|^{1+\tau}).$$

4.4 Numerical issues

In applying the semismooth Newton-CG method to solve the inner problem (20), the most expensive step is to compute the Newton direction from the linear equation involving the operator $\mathcal{V}_j$ in (46). As is well known, the basic operation in implementing the PCG method is to calculate the multiplication $\mathcal{V}_j y$ for any given $y \in \Re^p$. From the analysis in subsection 4.1, the computation appears to require the full SVD of an $m \times n$ matrix. For a problem in which m is moderate but n is large, the full SVD computation would be expensive and huge memory space is also needed to store the large and dense matrix V. To alleviate this difficulty, the authors in [20] suggested computing the full SVD indirectly via a reduced SVD and a QR factorization by Householder transformations. But in fact, we can completely avoid the computation of V_2 by carefully analyzing the structure of (31) as shown next. The part $\mathrm{Diag}\begin{bmatrix} \Upsilon_{\gamma_1} \\ 0 \end{bmatrix} \widetilde{H}_2 V_2^T$ in (31) is given as follows:

$$\begin{aligned} \mathrm{Diag}\begin{bmatrix} \Upsilon_{\gamma_1} \\ 0 \end{bmatrix} \widetilde{H}_2 V_2^T &= \begin{bmatrix} \mathrm{diag}(\Upsilon_{\gamma_1}) U_{\gamma_1}^T H (V_2 V_2^T) \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{diag}(\Upsilon_{\gamma_1}) U_{\beta_1}^T H (I - V_1 V_1^T) \\ 0 \end{bmatrix}. \end{aligned} \tag{48}$$

From (48), it is clear that V_2 can be avoided when evaluating $\mathcal{V}_0^1 H$ in (31). Given the economical SVD of $Z^k - \lambda_k(\mathcal{A}^* y - A_0)$, we can compute $\mathcal{V}_0^1 H$ in at most $k_1(\sigma)(18mn + 4m^2)$ flops for any given $H \in \Re^{m \times n}$. The above complexity shows that our algorithm is able to utilize any low rank or flat rectangular structure of a matrix to reduce the computational cost.

Next, we introduce two diagonal preconditioners to accelerate the convergence of the CG method applied to solve the linear system (46). Let $\mathbf{A}$ and $\mathbf{V}$ be the matrix representations of the linear mappings $\mathcal{A}$ and $\mathcal{V}_1^0$, respectively. Then the coefficient matrix in (46) has the following form

$$W = -\lambda \mathbf{A}\mathbf{V}\mathbf{A}^T - \lambda B_{\mathcal{J}_1}^T B_{\mathcal{J}_1} - \epsilon I.$$

Note that we have omitted the iteration index for brevity. Let the standard basis in $\Re^{m \times n}$ be $\{E_{ij} \in \Re^{m \times n} : 1 \le i \le m,\ 1 \le j \le n\}$, where E_{ij} is the matrix whose (i,j)-th entry is one and zero otherwise. The diagonal element of $\mathcal{V}_1^0$ with respect to the standard basis is given by

$$\mathbf{V}_{(i,j),(i,j)} = \big((U \circ U)\Lambda^\infty (V \circ V)\big)_{ij} - \frac{1}{k_1(\sigma)}\big((U_1' V_1'^T) \circ (U_1' V_1'^T)\big)_{ij} + \frac{1}{2}\langle H_{ij} \circ H_{ij}^T,\ \Omega^\infty - \Gamma^\infty \rangle, \tag{49}$$

where

$$\Lambda^\infty = \begin{bmatrix} \frac{1}{2}(\Omega^\infty + \Gamma^\infty) & \Upsilon^\infty \mathbf{1}^T_{k_1(\sigma) \times (n-m)} \\ \mathbf{0}_{(m-k_1(\sigma)) \times (n-k_2(\sigma))} & \end{bmatrix}, \quad H_{ij} = U^T E_{ij} V_1,$$

and U_1' and V_1' are the matrices formed by the first $k_1(\sigma)$ columns of U and V, respectively. To avoid excessive computational cost, we only calculate the first two terms on the right

side of (49)

$$\mathbf{D}_{(i,j),(i,j)} = \big((U \circ U)\Lambda^{\infty}(V \circ V)\big)_{ij} - \frac{1}{k_1(\sigma)}\big((U_1' V_1'^T) \circ (U_1' V_1'^T)\big)_{ij}, \tag{50}$$

as a good approximation of (49). Thus we propose the following diagonal preconditioner for the coefficient matrix:

$$M = \lambda \mathrm{Diag}(\mathbf{A}\mathbf{D}\mathbf{A}^T + B_{\mathcal{J}_1}^T B_{\mathcal{J}_1}) + \epsilon I. \tag{51}$$

Clearly, to use the preconditioner above, we need the explicit form of V, which may lead to memory difficulty when n is large. Thus when n is too large for V to be stored explicitly, we just use the following simple diagonal preconditioner

$$M' = \lambda \mathrm{Diag}(\mathbf{A}\mathbf{A}^T + B_{\mathcal{J}_1}^T B_{\mathcal{J}_1}) + \epsilon I. \tag{52}$$

5 Numerical results

In this section, we will apply our algorithm SNDPPA (semismooth Newton-CG dual PPA) to solve four different types of the MNA problems. For the purpose of numerical comparison, we will also report the performance of an alternating direction method of multipliers (ADMM) for solving the same set of problems. All the codes are written in MATLAB 7.11 and run an Intel Xeon 2.80GHz (quadcore) PC with 24GB memory.

The ADMM was first introduced in [10, 12]. Since then a number of variants have been studied theoretically and employed in many applications, see, e.g. [8, 11, 15, 16, 43]. Here we shall employ the classical ADMM [10,12] to solve the MNA problems. Note that problem (1) can be expressed in the following equivalent form:

$$\min\Big\{\|X\|_2 \mid \mathcal{A}^* y + X = A_0,\ By - b = z,\ z \in \mathcal{Q}\Big\}. \tag{53}$$

The augmented Lagrangian function associated with (53) is given by

$$\begin{aligned}\mathcal{L}_\beta(y, X, z; Z, w) := \|X\|_2 - \langle Z, \mathcal{A}^* y + X - A_0\rangle - \langle w, By - b - z\rangle \\ + \frac{\beta}{2}\|\mathcal{A}^* y + X - A_0\|^2 + \frac{\beta}{2}\|By - b - z\|^2,\end{aligned} \tag{54}$$

where Z and w are Lagrangian multipliers, and $\beta > 0$ is the penalty parameter. Given $X^0, Z^0 \in \Re^{m\times n}$, $z^0, w^0 \in \Re^{n_1+n_2}$, and $\beta_0 > 0$, the k-th iteration of the ADMM for (53) is given as follows:

$$\begin{aligned} y^{k+1} &= \arg\min\{\mathcal{L}_{\beta_k}(y, X^k, z^k; Z^k, w^k) \mid y \in \Re^p\}, \\ (X^{k+1}, z^{k+1}) &= \arg\min\{\mathcal{L}_{\beta_k}(y^{k+1}, X, z; Z^k, w^k) \mid (X, z) \in \Re^{m\times n} \times \mathcal{Q}\}, \\ Z^{k+1} &= Z^k - \varrho\beta_k(\mathcal{A}^* y^{k+1} + X^{k+1} - A_0),\ w^{k+1} = w^k - \varrho\beta_k(By^{k+1} - b - z^{k+1}), \end{aligned}$$

where $\varrho \in (0, \frac{1+\sqrt{5}}{2})$. It is easy to see that the minimizer y^{k+1} is the solution of the following linear system of equations:

$$(\mathcal{A}\mathcal{A}^* + B^T B)y^{k+1} = \mathcal{A}(A_0 - X^k + Z^k/\beta_k) + B^T(b + z^k + w^k/\beta_k). \tag{55}$$

Since $\mathcal{L}_{\beta_k}(y^{k+1}, X, z; Z^k, w^k)$ is separable in X and z, simple algebraic manipulations then give

$$\begin{aligned} X^{k+1} &= A_0 - \mathcal{A}^* y^{k+1} + Z^k/\beta_k - \Pi_{\mathcal{B}_\beta}(A_0 - \mathcal{A}^* y^{k+1} + Z^k/\beta_k), \\ z^{k+1} &= \Pi_{\mathcal{Q}}(By^k - b - w^k/\beta_k). \end{aligned}$$

As analyzed before, X^{k+1} can be computed analytically, and z^{k+1} is just a simple projection over $\mathcal{Q}$.

We use R_p, R_d and gap to denote, respectively, the primal infeasibility, dual infeasibility and primal-dual relative gap, namely

$$R_p = \frac{\|[\mathcal{A}^* y + X - A_0; \Pi_{Q^*}(b - By)]\|}{1 + \|[A_0; b]\|}, \; R_d = \frac{\|\mathcal{A}Z + B^T w\|}{1 + \|[\mathcal{A}; B^T]\|}, \; \text{gap} = \frac{|\text{pobj} - \text{dobj}|}{1 + |\text{pobj}| + |\text{dobj}|},$$

where pobj and dobj are the primal and dual objective values, respectively.

In our numerical experiments, we start the ADMM from the point $(X, y, z, Z, w) = (0, 0, 0, 0, 0)$ and terminate it when

$$\max\{R_p, R_d\} \le 10^{-6} \tag{56}$$

or the maximum number of iterations exceeds 2000. The penalty parameter β in the ADMM is adjusted dynamically. Starting from the initial value of 10, we adjust β at every fifth step as follows:

$$\beta_{k+1} = \begin{cases} \min(10^3, 2\beta_k), & \text{if } R_p^k/R_d^k < 0.1, \\ \max(10^{-2}, 0.5\beta_k), & \text{if } R_p^k/R_d^k > 10, \\ \beta_k, & \text{otherwise.} \end{cases} \tag{57}$$

For the SNDPPA, we use ADMM to generate an initial point satisfying $\max\{R_p, R_d\} < 5 \times 10^{-3}$. The total number of the ADMM steps is capped at 50. We terminate the SNDPPA when the condition (56) is met. For each PPA iteration, we cap the number of Newton-CG iterations for solving an inner subproblem to 40. In solving the linear system associated with the Newton direction, the maximal number of PCG steps is set as 500. As the parameter λ plays a critical role in the convergence speed of a PPA-based algorithm, we need to tune it with care. In our implementation, the parameter λ is initialized as 10 and updated according to the following rule:

$$\lambda_{k+1} = \begin{cases} 3\lambda_k, & R_p^{k+1}/R_p^k > 0.5 \text{ and } R_p^{k+1} > 10^{-4}, \\ 2\lambda_k, & R_p^{k+1}/R_p^k > 0.5 \text{ and } R_p^{k+1} < 10^{-4}, \\ \lambda_k, & \text{otherwise,} \end{cases} \tag{58}$$

where λ_k denotes the penalty parameter value at the kth PPA iteration.

5.1 Random matrix norm approximation

We first consider randomly generated MNA problems with/without constraints. In the experiments, the matrices $A_0, A_1, \ldots, A_p$ are generated independently from the multivariate uniform distribution on $[0,1]^{m\times n}$.

In Table 1, we report the numerical performance of the SNDPPA and the ADMM for solving different random matrix approximation instances without constraints. The number of outer iterations (iter), primal infeasibility (R_p), dual infeasibility (R_d), primal objective value (pobj), relative gap (gap), and the CPU time (time) taken are listed in the table. To better understand the performance of the SNDPPA, we also report the number of Newton systems solved (itersub) and the average number PCG steps (pcg) taken to solve each of the systems.

$p \mid m \mid n$	Algo.	it (itersub | pcg)	pobj | gap	$R_p \mid R_d$	time
300 | 300 | 300	PPA	14(15|3.9)	9.44515934 0| 2.8-6	4.4-7| 3.2-8	7.5
	ADMM	300	9.44520938 0| 4.2-6	9.7-7| 2.5-7	25.1
500 | 500 | 500	PPA	17(18|3.9)	1.22905150 1| 3.7-6	4.3-7| 2.3-8	38.8
	ADMM	619	1.22905586 1| 1.9-5	6.8-7| 9.9-7	234.2
100 | 100 | 3000	PPA	16(18|3.8)	1.83807818 1| 8.3-6	9.4-7| 4.1-8	10.7
	ADMM	821	1.83807914 1| 6.5-6	9.9-7| 1.7-7	94.9
100 | 100 | 5000	PPA	16(17|3.8)	2.31039070 1| 5.6-6	9.3-7| 4.0-8	17.5
	ADMM	443	2.31040515 1| 3.2-6	9.9-7| 9.7-7	87.6
100 | 100 | 10000	PPA	18(19|3.8)	3.16771120 1| 2.8-6	5.4-7| 1.1-7	39.4
	ADMM	740	3.16774836 1| 7.4-6	9.5-7| 9.9-7	299.9
100 | 100 | 20000	PPA	16(17|3.8)	4.37704442 1| 1.2-7	7.3-7| 2.7-9	83.0
	ADMM	654	4.37704413 1| 9.9-6	4.6-7| 9.9-7	666.6

Table 1: Results for unconstrained random matrix norm approximation problems.

As can be observed in Table 1, both the ADMM and SNDPPA are able to solve the unconstrained random matrix approximation problems to relatively high accuracy. The SNDPPA substantially outperforms the ADMM in terms of the CPU time taken to solve the problems. For example, the ADMM takes about 35 minutes to solve the last instance while our SNDPPA solves it in 4.5 minutes and with better accuracy in dual infeasibility and objective gap. Note that the random matrix approximation problems can also be solved by the interior point package SDPT3 [36] via the SDP reformulation (6). However, the interior solver may encounter computational and memory difficulties when solving large problems. For example, SDPT3 takes about 5 minutes to solve the problem with $(m,n,p) = (300,300,300)$ while our SNDPPA solves it in only 8 seconds. For the larger instance with $(m,n,p) = (100,5000,100)$, SDPT3 is not able to solve the problem due

to excessive computer memory required. It is worth noting that for the instances with $(p, m) = (100, 100)$, the CPU time taken by each iteration of the SNDPPA and the ADMM increases almost linearly with n. But for a solver (say the algorithm in [45]) that attempts to solve (1) via the SDP reformulation (6), the cost per iteration would grow at least quadratically in n. This observation is consistent with the fact mentioned in the previous section that our SNDPPA is capable of exploiting the flat rectangular structure of the matrices involved.

Next, we test our SNDPPA on the MNA problems with constraints. A simple example is to find a convex combination of given matrices $A_0, A_1, \ldots, A_p$ having the minimal spectral norm, i.e.,

$$\min \Big\{ \|A_0 - \mathcal{A}^* y\|_2 \mid \textstyle\sum_{i=1}^p y_i = 1, \quad y \geq 0 \Big\}. \tag{59}$$

In what follows, we investigate the performance of the SNDPPA and the ADMM applied to (59) where the matrices $A_1, \ldots, A_p$ are randomly generated as before. Table 2 lists the numerical results obtained by the SNDPPA and the ADMM. For this collection of problems, we can easily see the superiority of the SNDPPA over the first order algorithm ADMM. While our SNDPPA solves all the tested instances to the accuracy of 10^{-6} within 36 semismooth Newton-CG iterations, the ADMM fails to achieve the required accuracy even after 2000 iterations. For the instance with $(m, n, p) = (100, 20000, 100)$, the ADMM fails to achieve the accuracy of 10^{-6} after running for 33 minutes while our SNDPPA is able to solve the problem in less than 2 minutes. As one may deduce from the results in Table 2, the ADMM may encounter both computational and accuracy difficulties even only simple constraints are imposed on y.

$p \mid m \mid n$	Algo.	it (itersub | pcg)	pobj | gap	$R_p \mid R_d$	time
300 | 300 | 300	PPA	17(32|13.0)	9.59409299 0| 2.7-5	8.0-7| 3.2-8	22.4
	ADMM	2000	9.59309978 0| 3.8-6	4.2-6| 6.8-6	169.5
500 | 500 | 500	PPA	19(36|14.6)	1.24537637 1| 5.1-5	7.3-7| 6.0-7	133.7
	ADMM	2000	1.24556416 1| 1.2-4	1.1-5| 1.1-5	708.9
100 | 100 | 3000	PPA	21(27|4.7)	1.83873310 1| 3.5-6	5.2-7| 2.4-8	16.0
	ADMM	2000	1.83863304 1| 1.6-4	7.1-6| 6.7-6	239.9
100 | 100 | 5000	PPA	18(24|5.2)	2.31091411 1| 3.5-6	4.2-7| 2.8-8	24.6
	ADMM	2000	2.31077010 1| 1.0-4	8.3-6| 2.8-6	390.9
100 | 100 | 10000	PPA	19(25|4.7)	3.16803831 1| 6.2-6	8.7-7| 1.6-8	49.2
	ADMM	2000	3.16798808 1| 1.0-5	3.0-6| 9.9-7	787.3
100 | 100 | 20000	PPA	21(25|3.9)	4.37736720 1| 2.2-5	7.8-7| 9.3-7	105.6
	ADMM	2000	4.37716525 1| 6.2-5	4.5-6| 2.1-6	2001.5

Table 2: Results for the matrix norm approximation problem (59).

5.2 Chebyshev polynomials of matrices

In this subsection, we apply the proposed SNDPPA to compute the Chebyshev polynomials of a given matrix $A \in \Re^{n\times n}$. Since the power basis $I, A, \ldots, A^t$ is usually highly ill conditioned, in [38] the authors suggested replacing this basis by a better-conditioned alternative $Q_1, Q_2, \ldots, Q_{t+1}$ and consider the resulting problem

$$\min_{y\in\Re^t} \|Q_{t+1} - \textstyle\sum_{i=1}^t y_i Q_i\|_2. \tag{60}$$

From the solution of (60), one can easily compute the coefficients of the Chebyshev polynomials via Theorem 2 in [38]. In our experiments, the test examples are taken from Section 6 in [38] and $Q_1, Q_2, \ldots, Q_{t+1}$ is the orthogonal basis corresponding to the power basis of A.

problem	Algo.	$n \mid t$	it (itersub | pcg)	pobj | gap	$R_p \mid R_d$	time
Rand	PPA	500 | 50	16(18|7.7)	2.19977200-1| 3.6-7	3.4-7| 1.8-7	13.6
		1000 | 100	14(15|8.7)	1.84595015-1| 1.4-7	2.5-7| 1.8-7	78.7
	ADMM	500 | 50	354	2.19977287-1| 6.2-8	3.0-7| 8.7-7	45.9
		1000 | 100	661	1.84594835-1| 3.1-7	1.1-7| 9.1-7	492.8
Randtri	PPA	500 | 50	6(9|10.9)	4.14987173-1| 3.5-8	4.1-8| 2.5-7	8.2
		1000 | 100	9(12|14.1)	3.56509694-1| 6.6-8	3.2-7| 6.7-7	79.5
	ADMM	500 | 50	663	4.14987166-1| 1.4-6	7.1-8| 8.8-7	81.4
		1000 | 100	786	3.56509344-1| 1.1-6	7.3-8| 9.1-7	566.4
Diag	PPA	500 | 50	16(44|9.3)	7.20404744-2| 5.8-8	1.4-9| 9.3-8	15.8
		1000 | 100	15(27|9.1)	4.85090599-2| 1.8-7	3.1-8| 1.1-7	60.8
	ADMM	500 | 50	2000	7.20772064-2| 3.0-4	3.0-5| 3.6-4	90.4
		1000 | 100	396	4.85093725-2| 2.7-7	4.6-7| 8.7-7	105.6
Bidiag	PPA	500 | 50	11(37|18.8)	1.90877134-1| 4.5-7	2.0-7| 3.3-7	30.5
		1000 | 100	18(83|38.6)	1.38036105-1| 3.2-7	2.0-7| 1.6-7	654.1
	ADMM	500 | 50	1482	1.90877146-1| 2.2-7	2.0-7| 8.8-7	171.7
		1000 | 100	2000	1.38036596-1| 5.4-7	1.0-6| 9.1-7	1179.7
Ellipse	PPA	500 | 50	9(14|3.8)	5.51257423-2| 6.3-10	2.0-7| 2.0-9	6.3
		1000 | 100	11(20|4.4)	3.90141290-2| 6.4-11	4.7-7| 4.2-10	53.0
	ADMM	500 | 50	269	5.51257424-2| 4.4-7	4.7-7| 8.7-7	35.7
		1000 | 100	370	3.90141292-2| 3.6-7	4.8-7| 5.5-7	212.6
Grcar	PPA	500 | 50	17(42|8.6)	7.19041068-2| 6.5-8	3.8-7| 3.2-8	25.1
		1000 | 100	11(25|8.4)	5.07326772-2| 2.5-7	4.0-7| 4.0-7	98.0
	ADMM	500 | 50	2000	7.19051700-2| 2.2-6	2.4-6| 7.3-7	260.3
		1000 | 100	865	5.07326793-2| 9.0-7	2.6-7| 9.1-7	624.1
Lemniscate2	PPA	500 | 50	17(78|9.2)	8.09231048-2| 8.7-8	2.7-7| 3.0-7	49.3
		1000 | 100	20(74|18.4)	3.33334478-2| 1.3-6	3.8-7| 2.0-7	359.8
	ADMM	500 | 50	1104	8.09229960-2| 3.1-8	2.1-7| 8.7-7	140.8
		1000 | 100	1154	3.33337454-2| 1.8-6	4.9-7| 8.6-7	759.5

problem	Algo.	$n \mid t$	it (itersub | pcg)	pobj | gap	$R_p \mid R_d$	time
Wilkinson	PPA	500 | 50	13(25|6.0)	2.02888114-1| 2.9-8	1.9-7| 6.8-8	11.1
		1000 | 100	13(29|7.6)	1.92544545-1| 5.8-7	1.5-7| 2.7-7	75.9
	ADMM	500 | 50	859	2.02888114-1| 1.2-6	5.0-7| 8.5-7	80.8
		1000 | 100	1723	1.92544547-1| 9.2-8	5.8-8| 9.1-7	940.6
Chebyshev	PPA	500 | 50	10(15|8.7)	2.24960671-1| 2.8-6	1.5-7| 7.2-7	10.1
		1000 | 100	12(18|11.7)	2.06618544-1| 8.4-7	3.7-7| 7.7-7	103.5
	ADMM	500 | 50	788	2.24960549-1| 1.3-6	3.4-7| 8.7-7	80.5
		1000 | 100	2000	2.06629894-1| 2.7-4	4.8-6| 2.4-5	1159.0

Table 3: Chebyshev polynomials of matrices.

Table 3 shows that for most of the test instances, both the SNDPPA and the ADMM are capable of achieving the accuracy of less than 10^{-6}. However, for examples such as Bidiag with $n = 1000$ and diag with $n = 500$, the ADMM fails to solve them within 2000 iterations while the SNDPPA succeeds in achieving the required accuracy for all the instances. This illustrates that our SNDPPA performs much more robustly than the ADMM. Moreover, the SNDPPA is much more efficient than the ADMM in terms of computing time. Specifically, the former is about 5 to 10 times faster than the latter. Also, the performance of our SNDPPA is superior to the interior-point solver SDPT3 [36] applied to the SDP reformulation in terms of CPU time. For example, SDPT3 takes about 2 minutes to achieve a solution with an accuracy of 10^{-6} for the problem randN500t50 while our SNDPPA solves it in 13 seconds. This is not surprising since for most instances, the SNDPPA takes less than 30 semismooth Newton-CG iterations to generate a highly accurate solution and the average number of PCG steps needed to solve each of the Newton systems is less than 15.

5.3 FMMC/FDLA

In this subsection, we investigate the numerical performance of the two algorithms for solving the fastest Markov mixing chain (FMMC) problem (4) and the fastest distributed linear averaging (FDLA) problem (5). The tested graphs are taken from the sparse matrix collection [5] but some are slightly modified to make them connected. The data set is available at http://www2.research.att.com/~gyifanhu/GALLERY/GRAPHS/search.html.

problem	$p \mid n$	Algo.	it (itersub | pcg)	pobj | gap	$R_p \mid R_d$	time
FDLA–Cage	2562| 366	PPA	4(6|1.3)	4.58554209-1| 9.2-6	6.8-7| 3.0-9	1.8
		ADMM	2000	4.75216058-1| 3.4-1	1.1-3| 1.3-3	93.5
FMMC–Cage	2562| 366	PPA	5(6|1.5)	4.58545022-1| 5.7-7	4.1-8| 7.6-10	1.6
		ADMM	2000	5.87384195-1| 4.1-1	7.0-5| 1.5-3	95.8

problem	$p \mid n$	Algo.	it (itersub | pcg)	pobj | gap	$R_p \mid R_d$	time
FDLA–Erdos981	1381| 485	PPA	6(6|2.7)	1.00000000 0| 2.8-14	0.0-8| 0.0-16	2.6
		ADMM	20	1.00000000 0| 3.7-5	7.1-7| 3.0-8	2.2
FMMC–Erdos981	1381| 485	PPA	7(10|5.4)	1.00000000 0| 6.2-14	0.0-9| 1.4-7	4.5
		ADMM	23	1.00000000 0| 7.5-6	7.4-7| 1.5-7	2.5
FDLA–G3	19176| 800	PPA	11(16|31.8)	2.40597702-1| 3.3-4	9.4-7| 9.4-7	58.2
		ADMM	2000	2.41026286-1| 3.9-4	1.1-4| 5.3-6	773.3
FMMC–G3	19176| 800	PPA	16(24|43.5)	2.40914609-1| 4.0-7	5.3-7| 1.0-8	111.0
		ADMM	2000	2.41009134-1| 8.7-4	5.4-6| 3.0-5	792.7
FDLA–NotreDame_yeast	2203| 2114	PPA	12(17|4.3)	1.00000000 0| 1.3-11	0.0-9| 0.0-16	417.8
		ADMM	97	1.00000000 0| 1.9-11	5.8-7| 1.3-8	647.1
FMMC–NotreDame_yeast	2203| 2114	PPA	8(8|3.0)	1.00000000 0| 1.3-13	0.0-8| 0.0-16	199.9
		ADMM	26	1.00000000 0| 3.7-6	8.1-7| 2.0-8	146.6
FDLA–G46	9990| 1000	PPA	12(32|32.6)	4.17345194-1| 2.8-5	6.1-7| 5.1-7	185.5
		ADMM	2000	4.17421122-1| 3.4-4	4.5-6| 1.8-5	1308.1
FMMC–G46	9990| 1000	PPA	12(29|39.9)	4.19936229-1| 8.2-5	7.7-7| 7.6-8	196.2
		ADMM	2000	4.21142429-1| 9.4-4	1.2-4| 6.2-5	1328.8
FDLA–G15	4661| 800	PPA	13(54|37.3)	7.31900340-1| 1.2-4	8.8-7| 2.5-7	184.8
		ADMM	1122	7.31899758-1| 4.1-4	3.7-7| 9.4-7	449.1
FMMC–G15	4661| 800	PPA	15(78|89.8)	7.85238471-1| 2.0-5	1.3-7| 1.4-7	551.5
		ADMM	2000	7.85529701-1| 2.9-3	1.3-6| 5.7-6	797.7
FDLA–G54	5916| 1000	PPA	13(78|40.7)	7.32242244-1| 4.5-6	2.0-7| 1.2-7	522.0
		ADMM	2000	7.33611791-1| 2.0-4	7.8-5| 2.9-5	1481.4
FMMC–G54	5916| 1000	PPA	14(84|94.4)	7.86520875-1| 4.2-5	8.8-7| 3.6-7	1115.2
		ADMM	2000	7.88923019-1| 1.7-3	4.2-6| 3.5-5	1459.3
FDLA–G43	9990| 1000	PPA	14(28|32.3)	4.21302337-1| 1.1-4	2.0-7| 1.4-7	161.0
		ADMM	2000	4.21415022-1| 5.7-4	2.3-6| 2.3-5	1309.7
FMMC–G43	9990| 1000	PPA	16(47|52.4)	4.25983577-1| 4.7-6	9.4-7| 1.3-7	386.9
		ADMM	2000	4.26209610-1| 8.5-4	6.3-6| 3.3-5	1326.6

Table 4: Performance of the SNDPPA and the ADMM for FMMC/FDLA problems on connected graphs.

Table 4 shows that our SNDPPA is able to achieve the required accuracy of less than 10^{-6} for all the test examples. However, by comparing the results for FMMC/FDLA with those for the random matrix approximation and Chebyshev polynomial problems, we see that the SNDPPA is slower for the former cases. This behavior is understandable because for FMMC/FDLA problems, the average PCG steps taken to compute the Newton directions and the total number of semismooth Newton-CG iterations are significantly larger. It is also not surprising that the ADMM fails to obtain solutions with the desired accuracy after 2000 iterations for most of the instances. In fact, the ADMM can

only obtain an approximate solution with the primal-dual accuracy in the order 10^{-4} to 10^{-5} for about 50% of the instances and and the objective gap more than 10^{-4} for about 80% of the instances. The performance of the ADMM is especially poor for the problems `FDLA-Cage` and `FMMC-Cage`.

6 Conclusion

We proposed a semismooth Newton-CG dual proximal point algorithm (SNDPPA) to solve large scale matrix spectral norm approximation problems. In each iteration, the dual PPA solves its subproblem by a semismooth Newton-CG method and the Newton direction is computed inexactly by a PCG solver. Theoretical results to guarantee the global convergence and local superlinear convergence of the dual PPA are established based on the classical analysis of proximal point algorithms. Capitalizing on the recent advances on spectral operator and related perturbation analysis, we also characterize the nonsingularity of the semismooth Newton systems. The latter property is an important condition for the fast convergence of the semismooth Newton-CG method. Extensive numerical experiments on problems arising from different areas are conducted to evaluate the performance of the SNDPPA against the ADMM. The numerical results show that the SNDPPA is very efficient and robust, and it substantially outperforms the ADMM.

References

[1] J. F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems*, Springer Verlag, New York, 2000.

[2] S. Boyd, P. Diaconis, J. Sun, and L. Xiao, *Fastest mixing Markov chain on a path*, Amer. Math. Monthly, 113 (2006), pp. 70–74.

[3] S. Boyd, P. Diaconis, and L. Xiao, *Fastest mixing Markov chain on a graph*, SIAM Rev., 46 (2004), pp. 667–689.

[4] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Second Edition, SIAM, Philadelphia, PA, 1990.

[5] T. A. Davis and Y. Hu, University of Florida Sparse Matrix Collection, 2009.

[6] C. Ding, *An Introduction to A Class of Matrix Optimization Problems*, PhD thesis, National University of Singapore, Singapore, 2012.

[7] C. Ding, D. F. Sun, and K.-C. Toh, *An introduction to a class of matrix cone programming*, Available at http://www.math.nus.edu.sg/∼mattohkc/publist.html, 2010.

[8] J. Eckstein and D. P. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Program., Ser. A, 55 (1992), pp. 293–318.

[9] F. Facchinei and J.-S. Pang, *Finite-dimensional Variational Inequalities and Complementarity Problems*, Vol. II, Springer-Verlag, New York, 2003.

[10] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximations*, Comp. Math. Appl., 2 (1976), pp. 17–40.

[11] R. Glowinski and P. L. Tallec, *Augmented Lagrangian and Operator Splitting Methods in Nonlinear Mechanics*, SIAM, Philadelphia, PA, 1989.

[12] R. Glowinski and A. Marrocco, *Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité, d'une classe de problèmes de Dirichlet non-linéaires*, Rev. Française Automat. Informat. Recherche Opérationnelle Anal. Numér., 9 (1975), pp. 41–76.

[13] A. Greenbaum and L. N. Trefethen, *GMRES/CR and Arnoldi/Lanczos as matrix approximation problems*, SIAM J. Sci. Comput., 15 (1994), pp. 359–368.

[14] C. D. Ha, *A generalization of the proximal point algorithm*, SIAM J. Control Optim., 28 (1990), pp. 503–512.

[15] B. S. He, L.-Z. Liao, D. R. Han, and H. Yang, *A new inexact alternating directions method for monotone variational inequalities*, Math. Program. Ser. A, 92 (2002), pp. 103–118.

[16] B. S. He, M. H. Xu, and X. M. Yuan, *Solving large-scale least squares semidefinite programming by alternating direction methods*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 136–152.

[17] M. Held, P. Wolfe, and H. P. Crowder, *Validation of subgradient optimization*, Math. Program., 6 (1974), pp. 62–88.

[18] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Vols. I and II, Springer, Berlin, 1993.

[19] J. -B. Hiriart-Urruty, J. -J. Strodiot, V. H. Nguyen, *Generalized Hessian matrix and second-order optimality conditions for problems with $C^{1,1}$ data, Appl. Math. Optim., 11 (1984), pp. 43Ũ56.*

[20] K. F. Jiang, Algorithms for Large Scale Nuclear Norm Minimization and Convex Quadractic Semidefinite Programming Problems. *PhD thesis, National University of Singapore, Singapore, 2012.*

[21] C. Lemaréchal and C. Sagastizábal, Practical aspects of the Moreau-Yosida regularization: theoretical preliminaries*, SIAM J. Optim., 7 (1997), pp. 367–385.*

[22] A. S. Lewis and H. S. Sendov, Nonsmooth analysis of singular values. II: applications*, Set-Valued Anal., 13 (2005), pp. 243–264.*

[23] J. Liese and P. Tichý, On best approximations of polynomials in matrices in the matrix 2-norm, *SIAM J. Matrix Anal. Appl., 31 (2009), pp. 853–863.*

[24] V. Faber, J. Liesen, P. Tichý, On Chebyshev polynomials of matrices, *SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2205–2221.*

[25] Y.-J. Liu, D. F. Sun, and K.-C. Toh, An implementable proximal point algorithmic framework for nuclear norm minimization, *Math. Program. Ser. A, 133 (2012), pp. 399–436.*

[26] F. W. Meng, D. F. Sun, and G. Y. Zhao, Semismoothness of solutions to generalized equations and the Moreau-Yosida regularization, *Math. Program. Ser. B, 104 (2005), pp. 561–581.*

[27] J. J. Moreau, Proximite et dualite dans un espace hilbertien, *Bull. Soc. Math. France, 93 (1965), pp. 273–299.*

[28] L. Q. Qi and J. Sun, A nonsmooth version of Newton's method, *Math. Program. Ser. A, 58 (1993), pp. 353–367.*

[29] X. Chen, H.-D. Qi, L. Qi and K.-L. Teo, Smooth convex approximation to the maximum eigenvalue function, *J. Global Optim., 30 (2004), pp. 253–270.*

[30] S. M. Robinson, Local structure of feasible sets in nonlinear programming. II: nondegeneracy, *Math. Program. Stud., 22 (1984), pp. 217–230.*

[31] R. T. Rockafellar, Augmented Lagrangians and applications of the proximal point algorithm in convex programming, *Math. Oper. Res., 1 (1976), pp. 97–116.*

[32] R. T. Rockafellar, Convex Analysis, *Princeton University Press, Princeton, New Jork, 1970.*

[33] R. T. Rockafellar, Monotone operators and the proximal point algorithm, *SIAM J. Control Optim., 14 (1976), pp. 877–898.*

[34] I. Singer, *Best Approximation in Normed Linear Spaces by Elements of Linear Subspaces, Springer Verlag, Berlin, 1970.*

[35] J. F. Sturm, Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, *Optim. Methods Softw., 11-12 (1999), pp. 625–653.*

[36] K.-C. Toh, Solving large scale semidefinite programs via an iterative solver on the augmented systems, *SIAM J. Optim., 14 (2003), pp. 670–698.*

[37] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, SDPT3—a MATLAB software package for semidefinite programming, version 1.3, *Optim. Methods Softw., 11-12 (1999), pp. 545–581.*

[38] K.-C. Toh and L. N. Trefethen, The Chebyshev polynomials of a matrix, *SIAM J. Matrix Anal. Appl., 20 (1999), pp. 400–419.*

[39] J. von Neumann, Some matrix inequalities and metrization of metric spaces, *Tomsk Univ. Rev., 1 (1937), pp. 286–300.*

[40] C. J. Wang, D. F. Sun, and K.-C. Toh, Solving log-determinant optimization problems by a Newton-CG primal proximal point algorithm, *SIAM J. Optim., 20 (2010), pp. 2994–3013.*

[41] G. A. Watson, Characterization of the subdifferential of some matrix norms, *Linear Algebra Appl., 170 (1992), pp. 33–45.*

[42] L. Xiao and S. Boyd, Fast linear iterations for distributed averaging, *Systems Control Lett., 53 (2004), pp. 65–78.*

[43] J. F. Yang and Y. Zhang, Alternating direction algorithms for ℓ_1-problems in compressive sensing, *SIAM J. Sci. Comput., 33 (2011), pp. 250–278.*

[44] K. Yosida, Functional Analysis, *Springer Verlag, Berlin, 1964.*

[45] X.-Y. Zhao, D. F. Sun, and K.-C. Toh, A Newton-CG augmented Lagrangian method for semidefinite programming, *SIAM J. Optim., 20 (2010), pp. 1737–1765.*

[46] K. Zietak, Properties of linear approximations of matrices in the spectral norm, *Linear Algebra Appl., 183 (1993), pp. 41–60.*

[47] K. Zietak, On approximation problems with zero-trace matrices, *Linear Algebra Appl., 247 (1996), pp. 169–183.*

Appendix: Proof of Proposition 4.1

Let $\{Y^i\}_{i\geq 1}$ be a sequence converging to Y such that every element $Y^i \in \mathcal{D}_{\Pi_{\mathcal{B}}}$. This, by Proposition 2.3 (i), implies that $\|Y^i\|_* \neq 1$ for each $i \geq 1$. Let the SVD of Y^i be $Y^i = U^i[\mathrm{Diag}(\sigma^i)\ 0](V^i)^T$. We consider the following three cases.

(i) $\|Y\|_* < 1$. In this case, $\Pi_{\mathcal{B}}(\cdot)$ is continuously differentiable at Y and its generalized Jacobian is a singleton set consisting of the identity operator from $\Re^{m\times n}$ to itself.

(ii) $\|Y\|_* = 1$. Since Y can be approximated by a sequence in the interior of $\mathcal{B}$, it follows that the identity operator $\mathcal{I}$ is always an element of $\partial_{\mathbf{B}}\Pi_{\mathcal{B}}(Y)$. To obtain the remaining elements, we consider the case in which $\{Y^i\}$ has an infinite subsequence outside $\mathcal{B}$. Without loss of generality, we assume that $\|Y^i\|_* > 1$ for all i. By passing to a subsequence if necessary, we know that there exists a positive integer $N \in [r, m]$ such that $N = k_1(\sigma^i)$ for each i. Therefore, one has

$$g_k^i := (\Pi_{\mathbb{B}}(\sigma^i))_k \ = \ \begin{cases} \sigma_k^i - \frac{1}{N}\Big(\sum_{j=1}^N \sigma_j^i - 1\Big), & 1 \leq k \leq N, \\ 0, & \text{otherwise}, \end{cases}$$

and

$$\Pi'_{\mathbb{B}}(\sigma^i) = \begin{bmatrix} I_N & 0 \\ 0 & 0 \end{bmatrix} - \frac{1}{N}\begin{bmatrix} \mathbf{1}_{N\times N} & 0 \\ 0 & 0 \end{bmatrix}.$$

For each i, it holds that

$$\big(\Omega(\sigma^i)\big)_{kj} = \begin{cases} 1, & \text{if } k,j \in \alpha_1 \cup \alpha_2, \\ \frac{g^i_k}{\sigma^i_k - \sigma^i_j}, & \text{if } k \in \alpha_1,\, j \in \alpha_3, \\ 0, & \text{if } k \in \alpha_3,\, j \in \alpha_3, \end{cases}, \big(\Gamma(\sigma^i)\big)_{kj} = \begin{cases} \frac{g^i_k + g^j_k}{\sigma^i_k + \sigma^i_j}, & \text{if } k \in \alpha_1,\, j \in \alpha_1 \cup \alpha_2, \\ \frac{g^i_k}{\sigma^i_k + \sigma^i_{j+N}}, & \text{if } k \in \alpha_1,\, j \in \alpha_3, \\ 0, & \text{if } k \in \alpha_3,\, j \in \alpha_3, \end{cases}$$

and

$$\big(\mathcal{F}(\sigma^i)\big)_{kj} = \begin{cases} -\frac{1}{N}, & \text{if } k,j \in \alpha_1 \cup \alpha_2, \\ 0, & \text{otherwise}, \end{cases}, \big(\Upsilon(\sigma^i)\big)_k = \begin{cases} \frac{g^i_k}{\sigma^i_k}, & \text{if } k \in \alpha_1, \\ 0, & \text{if } k \in \alpha_3. \end{cases}$$

Now from Proposition 2.3 (i), we know that for any given $H \in \Re^{m\times n}$,

$$\Pi'_{\mathcal{B}}(Y^i)H = U^i \left[W^i - \frac{\mathrm{Tr}(\widetilde{H}^i_{11})}{N}\begin{bmatrix} I_N & 0 \\ 0 & 0 \end{bmatrix},\ \mathrm{Diag}\big(\Upsilon(\sigma^i)\big)\widetilde{H}^i_2 \right](V^i)^T, \tag{61}$$

where the matrix $W^i \in \Re^{m\times m}$ is defined by

$$W^i = \Omega(\sigma^i) \circ S(\widetilde{H}^i_1) + \Gamma(\sigma^i) \circ T(\widetilde{H}^i_1) - \frac{\mathrm{Tr}(\widetilde{H}^i_{11})}{N}\begin{bmatrix} I_N & 0 \\ 0 & 0 \end{bmatrix}$$

with $\widetilde{H}^i_1 \in \Re^{m\times m}, \widetilde{H}^i_2 \in \Re^{m\times(n-m)}$, $[\widetilde{H}^i_1\ \widetilde{H}^i_2] = (U^i)^T H V^i$ and $\widetilde{H}^i_{11}$ being the matrix extracted from the first N columns and rows of $\widetilde{H}^i_1$. By simple algebraic computations, we have

$$\begin{aligned} &\lim_{i\to\infty}\big(\Omega(\sigma^i)\big)_{\alpha_1\alpha_3} = \mathbf{1}_{r\times(m-N)}, \quad \lim_{i\to\infty}\big(\Gamma(\sigma^i)\big)_{\alpha_1(\alpha_1\cup\alpha_2)} = \mathbf{1}_{r\times N}, \\ &\lim_{i\to\infty}\big(\Gamma(\sigma^i)\big)_{\alpha_1\alpha_3} = \mathbf{1}_{r\times(m-N)}, \quad \lim_{i\to\infty}\big(\Upsilon(\sigma^i)\big)_{\alpha_1} = \mathbf{1}_r. \end{aligned}$$

Note that $\Big\{\Big(\big(\Omega(\sigma^i)\big)_{\alpha_2\alpha_3}, \big(\Gamma(\sigma^i)\big)_{\alpha_2\alpha_2}, \big(\Gamma(\sigma^i)\big)^i_{\alpha_2\alpha_3}, \big(\Gamma(\sigma^i)\big)_{\alpha_2}\Big)\Big\}_{i\ge1}$ is bounded and $\mathcal{S}_N$ is the set of cluster points associated with the sequence. By taking limits on both sides of (61), we are able to establish the conclusion that for any $\mathcal{V}(\neq\mathcal{I}) \in \partial_{\mathbf{B}}\Pi_{\mathcal{B}}(Y)$, there exist an integer $N \in [r, m]$, $(\Omega^\infty_{\alpha_2\alpha_3}, \Gamma^\infty_{\alpha_2\alpha_2}, \Gamma^\infty_{\alpha_2\alpha_3}, \Upsilon^\infty_{\alpha_2}) \in \mathcal{S}_N$ and singular vector matrices U^∞, V^∞ of Y such that for any $H \in \Re^{m\times n}$, (28) is valid.

(iii) $\|Y\|_* > 1$. Taking a subsequence if necessary, we know that there exists a positive integer $N \in [k_1(\sigma), k_2(\sigma)]$ such that $N = k_1(\sigma^i)$ for each i. Therefore,

$$g^i_k := (\Pi_{\mathbb{B}}(\sigma^i)_k = \begin{cases} \sigma^i_k - \frac{1}{N}\Big(\sum_{j=1}^N \sigma^i_j - 1\Big), & 1 \le k \le N, \\ 0, & \text{otherwise}, \end{cases}$$

and

$$\Pi'_{\mathbb{B}}(\sigma^i) = \begin{bmatrix} I_N & 0 \\ 0 & 0 \end{bmatrix} - \frac{1}{N} \begin{bmatrix} \mathbf{1}_{N\times N} & 0 \\ 0 & 0 \end{bmatrix}.$$

For each i, it holds that

$$\big(\Omega(\sigma^i)\big)_{kj} = \begin{cases} 1, & \text{if } k,j \in \gamma_1, \\ \frac{g_k^i}{\sigma_k^i - \sigma_j^i}, & \text{if } k \in \beta_1,\, j \in \beta_3, \\ \frac{g_k^i}{\sigma_k^i - \sigma_j^i}, & \text{if } k \in \gamma_1,\, j \in \beta_4 \cup \beta_5, \\ 0 & \text{if } k \in \gamma_2,\, j \in \gamma_2, \end{cases} \quad \big(\Gamma(\sigma^i)\big)_{kj} = \begin{cases} \frac{g_k^i + g_k^j}{\sigma_k^i + \sigma_j^i}, & \text{if } k,j \in \gamma_1, \\ \frac{g_k^i}{\sigma_k^i + \sigma_j^i}, & \text{if } k \in \gamma_1,\, j \in \gamma_2, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\big(\mathcal{F}(\sigma^i)\big)_{kj} = \begin{cases} -\frac{1}{N}, & \text{if } k,j \in \gamma_1, \\ 0, & \text{otherwise,} \end{cases} \quad \big(\Upsilon(\sigma^i)\big)_k = \begin{cases} \frac{g_k^i}{\sigma_k^i}, & \text{if } k \in \gamma_1, \\ 0, & \text{otherwise.} \end{cases}$$

Then the equality (61) is also valid. Simple calculations show that

$$\begin{aligned} &\lim_{i\to\infty} \big(\Omega(\sigma^i)\big)_{\beta_1\beta_3} = \Omega_{\beta_1\beta_3}, \quad \lim_{i\to\infty} \big(\Omega(\sigma^i)\big)_{\gamma_1(\beta_4\cup\beta_5)} = \Omega_{\gamma_1(\beta_4\cup\beta_5)}, \\ &\lim_{i\to\infty} \big(\Gamma(\sigma^i)\big)_{\gamma_1\gamma_1} = \Gamma_{\gamma_1\gamma_1}, \quad \lim_{i\to\infty} \big(\Gamma(\sigma^i)\big)_{\gamma_1\gamma_2} = \Omega_{\gamma_1\gamma_2}, \\ &\lim_{i\to\infty} \big(\Upsilon(\sigma^i)\big)_{\gamma_1} = \Upsilon_{\gamma_1}. \end{aligned}$$

Note that $\left\{\big(\Omega(\sigma^i)\big)_{\beta_2\beta_3}\right\}$ is bounded and $\mathcal{T}_N$ is the set of cluster points associated with the sequence. By taking limits on both sides of (61), we have the conclusion that for any $\mathcal{V} \in \partial_{\mathbf{B}}\Pi_{\mathcal{B}}(Y)$, there exist an integer $N \in [k_1(\sigma),\, k_2(\sigma)]$, $\Omega^{\infty}_{\beta_2\beta_3} \in \mathcal{S}_N$ and singular vector matrices U^{∞}, V^{∞} of Y such that for any $H \in \Re^{m\times n}$, (29) holds. This completes the proof.