

**A SMOOTHING NEWTON-BICGSTAB
METHOD FOR LEAST SQUARES MATRIX
NUCLEAR NORM PROBLEMS**

LUO YANYING

(Bsc., NUS)

**A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF SCIENCE
DEPARTMENT OF MATHEMATICS
NATIONAL UNIVERSITY OF SINGAPORE
2010**

Acknowledgements

I would like to express my deepest thanks and respect to my supervisor Professor Sun Defeng. He has patiently introduced me into the field of optimization and has provided guidance and encouragement throughout my study. My sincere respect to him came from his enthusiasm in the optimization field and his effort in organizing weekly optimization discussion sessions, which had become a fruitful experience and a great learning opportunity for me in this research field.

My sincere thanks also go to all the friends in the department of mathematics: Gao Yan, Liu Yongjin, Zhao Xinyuan, Jiang Kaifeng, Ding Chao and Yang Zhe, for their kindly help and support throughout the project.

Luo Yanying/Jan 2010

Contents

Acknowledgements	ii
Abstract	v
1 Introduction	1
2 Preliminaries	6
2.1 The Lagrangian Dual Problem and Optimality Conditions	6
2.2 The Differential Properties of the Smoothing Functions	11
3 A Smoothing Newton-BiCGStab Method	23
4 Numerical Experiments	27
4.1 Implementation Issues	27
4.2 Numerical Experiments	28
5 Conclusions	37

A Smoothing Newton-BiCGStab Method for Least Squares Matrix Nuclear Norm Problems

Luo Yanying

Department of Mathematics, Faculty of Science
National University of Singapore

Master's thesis

Abstract

In this thesis, we study a smoothing Newton-BiCGStab method for the least squares nonsymmetric matrix nuclear norm problems. For this type of problems, when linear inequality and second-order cone constraints are present, the dual problem is equivalent to a system of nonsmooth equations. Some smoothing functions are introduced to the nonsmooth layers of the system. We will prove that the smoothed system of equations for nonsymmetric matrix problems inherits the strong semismoothness property from the real-valued smoothing functions. As a result, we show that the smoothing Newton-BiCGStab method which was introduced for solving least squares semidefinite programming problems can be extended to solve the least squares nonsymmetric matrix nuclear norm problems.

Chapter

1

Introduction

Let $\Re^{n_1 \times n_2}$ be the space of $n_1 \times n_2$ real valued matrices and $n_1 \leq n_2$. Denote the nuclear norm of $X \in \Re^{n_1 \times n_2}$ by $\|X\|_* = \sum_{i=1}^{n_1} \sigma_i(X)$, where $\sigma_1(X) \geq \sigma_2(X) \geq \dots \geq \sigma_{n_1}(X)$ are singular values of X . Let $\|\cdot\|_2$ stand for the Euclidean norm, and $\|\cdot\|_F$ denote the Frobenius norm which is induced by the standard trace inner product $\langle X, Y \rangle = \text{trace}(Y^T X)$ in $\Re^{n_1 \times n_2}$. Let $\{\mathcal{A}^e, \mathcal{A}^l, \mathcal{A}^q, \mathcal{A}^u\}$ be the linear operators used in four types of constraints respectively: linear equality, linear inequality, second-order cone, and linear vector space constraints. Each of these operators is a linear mapping from $\Re^{n_1 \times n_2}$ to \Re^{m_*} defined respectively by

$$\begin{aligned}
 \mathcal{A}^e(X) &: \Re^{n_1 \times n_2} \rightarrow \Re^{m_e} = [\langle A_1^e, X \rangle, \dots; \langle A_{m_e}^e, X \rangle], \\
 \mathcal{A}^l(X) &: \Re^{n_1 \times n_2} \rightarrow \Re^{m_l} = [\langle A_1^l, X \rangle, \dots; \langle A_{m_l}^l, X \rangle], \\
 \mathcal{A}^q(X) &: \Re^{n_1 \times n_2} \rightarrow \Re^{m_q} = [\langle A_1^q, X \rangle, \dots; \langle A_{m_q}^q, X \rangle], \\
 \mathcal{A}^u(X) &: \Re^{n_1 \times n_2} \rightarrow \Re^{m_u} = [\langle A_1^u, X \rangle, \dots; \langle A_{m_u}^u, X \rangle].
 \end{aligned}$$

The least squares matrix nuclear norm problems discussed in this thesis are of the form:

$$\begin{aligned}
\min \quad & \rho \|X\|_* + \frac{\mu}{2} \|x_u\|_2^2 + \frac{\lambda}{2} \|X - C\|_F^2 \\
\text{s.t.} \quad & \mathcal{A}^e(X) - b^e = 0, \quad b^e \in \mathfrak{R}^{m_e}, \\
& \mathcal{A}^l(X) - b^l \geq 0, \quad b^l \in \mathfrak{R}^{m_l}, \\
& \mathcal{A}^q(X) - b^q \in \mathcal{K}^{m_q}, \quad b^q \in \mathfrak{R}^{m_q}, \\
& \mathcal{A}^u(X) - b^u = x_u, \quad b^u \in \mathfrak{R}^{m_u}, \\
& x_u \in \mathfrak{R}^{m_u}, \quad X \in \mathfrak{R}^{n_1 \times n_2},
\end{aligned} \tag{1.1}$$

where the constants are required to be $\rho \geq 0$, $\mu > 0$, $\lambda > 0$, C is some matrix in $\mathfrak{R}^{n_1 \times n_2}$ and \mathcal{K}^{m_q} denotes a second order cone which is defined by

$$\mathcal{K}^{m_q} := \{y \in \mathfrak{R}^{m_q} \mid y_{m_q} \geq \|y^t\|_2\},$$

where $y = [y_1; y_2; \dots; y_{m_q-1}; y_{m_q}] = [y^t; y_{m_q}]$. Let

$$\begin{aligned}
\mathcal{W}(X) &:= [\mathcal{A}^e; \mathcal{A}^l; \mathcal{A}^q; \mathcal{A}^u](X), \\
T(x_u) &:= [0; 0; 0; x_u], \\
b &:= [b^e; b^l; b^q; b^u] \\
m &:= m_e + m_l + m_q + m_u
\end{aligned}$$

and $Q = \{0\}^{m_e} \times \mathfrak{R}_+^{m_l} \times \mathcal{K}^{m_q} \times \{0\}^{m_u}$. The feasible set \mathcal{F} of the problem (1.1) becomes

$$\mathcal{F} = \{(X, x_u) \in \mathfrak{R}^{n_1 \times n_2} \times \mathfrak{R}^{m_u} \mid \mathcal{W}(X) - T(x_u) \in b + Q\}.$$

Let $f(X, x_u) = \rho \|X\|_* + \frac{\mu}{2} \|x_u\|_2^2 + \frac{\lambda}{2} \|X - C\|_F^2$. Problem (1.1) is a convex problem of the form,

$$\begin{aligned}
\min \quad & f(X, x_u) \\
\text{s.t.} \quad & \mathcal{W}(X) - T(x_u) \in b + Q, \\
& X \in \mathfrak{R}^{n_1 \times n_2}, \quad x_u \in \mathfrak{R}^{m_u}.
\end{aligned} \tag{1.2}$$

The dual cone Q^+ of the closed convex cone Q is given by $Q^+ = \Re^{m_e} \times \Re_+^{m_l} \times \mathcal{K}^{m_q} \times \Re^{m_u}$. The dual problem of (1.1) to be derived in Chapter 2 is of the following form,

$$\begin{aligned} \min \quad & \theta(y) \\ \text{s.t.} \quad & y \in Q^+. \end{aligned} \tag{1.3}$$

When consider problem (1.1), in which X is a symmetric positive semidefinite cone, that is $X \in \mathcal{S}_+^n$, instead of $X \in \Re^{n_1 \times n_2}$, Newton type methods have been used to solve problems with only linear equality and inequality constraints. For example, the inexact Newton-BiCGStab method has been incorporated with some smoothing functions to solve the least squares covariance matrix (*LSCM*) problems with equality and inequality constraints [6],

$$\begin{aligned} \min \quad & \frac{1}{2} \|X - C\|_F^2 \\ (LSCM) \quad \text{s.t.} \quad & \langle A_i, X \rangle = b_i, \quad i = 1, \dots, m_e, \\ & \langle A_i, X \rangle \geq b_i, \quad i = m_e + 1, \dots, m_e + m_l, \\ & X \in \mathcal{S}_+^n. \end{aligned}$$

The dual problem of (*LSCM*) is of the same form as (1.3) and $Q^+ = \Re^{m_e} \times \Re_+^{m_l}$. In absence of the inequality constraints, we have $Q^+ = \Re^{m_e}$, which implies that the dual of (*LSCM*) problem is an unconstrained convex optimization problem. Based on a result of [18], we know that when $\nabla\theta$ is a strongly semismooth function though it is not continuously differentiable. One can still find a quadratically convergent method for solving (*LSCM*) problems [16]. When inequality constraints are present, the dual problem becomes a constrained problem, which can be transformed into a system of equations,

$$F(y) := y - \Pi_{Q^+}(y - \nabla\theta(y)) = \mathbf{0}. \tag{1.4}$$

In this system, the projector $\Pi_{Q^+}(\cdot)$ is a metric projection from $\Re^{m_e+m_l}$ to Q^+ . The function $\nabla\theta$ involves another metric projector onto the symmetric positive

semidefinite cone. The two layers of metric projectors have created obstacles to a direct use of Newton type of algorithms to achieve a quadratic convergence rate. To tackle this problem, Gao and Sun [6] applied some smoothing functions to the two nonsmooth layers of metric projectors in F . A Newton-BiCGStab algorithm is used to solve a smoothed system of (1.4). Their results have shown a promised quadratic convergence rate for the $(LSCM)$ problems with linear inequality constraints.

The $(LSCM)$ problem has recently been used by Gao and Sun [7] to iteratively solve the H-Weighted least squares semidefinite programming problems with an additional rank constraint,

$$\begin{aligned}
\min \quad & \frac{1}{2} \|H \circ (X - C)\|_F^2 \\
\text{s.t.} \quad & \langle A_i, X \rangle = b_i, \quad i = 1, \dots, m_e, \\
& \langle A_i, X \rangle \geq b_i, \quad i = m_e + 1, \dots, m_e + m_l, \\
& \text{rank}(X) \leq k, \\
& X \in \mathcal{S}_+^n,
\end{aligned} \tag{1.5}$$

where $H \geq 0$ is a given matrix and " \circ " denotes the Hadamard product of two matrices. Note that $\sum_{i=k+1}^n \sigma_i(X) = 0$ iff $\text{rank}(X) \leq k$. The rank constraint may

be replaced by putting a penalty term $\rho(\sum_{i=1}^n \sigma_i(X) - \sum_{i=1}^k \sigma_i(X))$ to the objective function. The idea of the majorized penalty approach given in [7] is to solve a sequence of $(LSCM)$ problems of the form,

$$\begin{aligned}
\min \quad & \frac{1}{2} \|X - C\|_F^2 + \rho \sum_{i=1}^n \sigma_i(X) - \langle C_\rho, X \rangle \\
\text{s.t.} \quad & \langle A_i, X \rangle = b_i, \quad i = 1, \dots, m_e, \\
& \langle A_i, X \rangle \geq b_i, \quad i = m_e + 1, \dots, m_e + m_l, \\
& X \in \mathcal{S}_+^n,
\end{aligned}$$

where $\langle C_\rho, X \rangle$ is some linearized form of $\rho \sum_{i=1}^k \sigma_i(X)$.

Problem (1.5) is a type of structure preserving low rank problems for symmetric positive semidefinite matrices. On the other hand, there are a lot of applications of the structure preserving low rank approximation problems for nonsymmetric matrices [3], which are of the form,

$$\begin{aligned} \min \quad & \frac{1}{2} \|H \circ (X - C)\|_F^2 \\ \text{s.t.} \quad & X \in \Omega, \\ & \text{rank}(X) \leq k, \\ & X \in \Re^{n_1 \times n_2}, \end{aligned}$$

where Ω is closed convex set containing some structures to be preserved. Once the ideas in [7] are applied to the above structure preserving low rank approximation problems, we will obtain problems of the form (1.1) if Ω is properly chosen. For this, see the last section in [7].

Given its potential importance of problem (1.1) for solving structure preserving low rank approximation problems and beyond, we will focus on solving problem (1.1).

In this thesis, the least squares matrix nuclear norm minimization problems will be shown to have similar properties as the (*LSCM*) problems. The smoothing Newton-BiCGStab method will be applied to solve problem (1.1). Preliminaries such as derivations of the dual problem, optimality conditions, constructions of smoothing functions, the continuous and differentiable properties of nonsymmetric matrix-valued functions that are involved in solving problem (1.1) will be presented in the next chapter. In Chapter 3, the smoothing Newton-BiCGStab method is illustrated with the convergence analysis. Implementation related issues and numerical experiments will be discussed in Chapter 4, and followed by conclusions in Chapter 5.

Chapter 2

Preliminaries

2.1 The Lagrangian Dual Problem and Optimality Conditions

In this chapter, we denote the primal problem (1.2) by (P) .

The Lagrangian function $L(X, x_u, y): \mathfrak{R}^{n_1 \times n_2} \times \mathfrak{R}^{m_u} \times \mathfrak{R}^m \rightarrow \mathfrak{R}$ for (P) is defined by

$$L(X, x_u, y) = f(X, x_u) - \langle \mathcal{W}(X) - T(x_u) - b, y \rangle. \quad (2.1)$$

Let $Q^+ = \mathfrak{R}^{m_e} \times \mathfrak{R}_+^{m_l} \times \mathcal{K}^{m_q} \times \mathfrak{R}^{m_u}$ be the dual cone of Q . The dual objective function $g(y)$ can be derived from the Lagrangian function (2.1) by

$$\begin{aligned}
g(y) &= \inf_{X, x_u} L(X, x_u, y) \\
&= \inf_{X, x_u} \{f(X, x_u) - \langle \mathcal{W}(X) - T(x_u) - b, y \rangle\} \\
&= \inf_{X, x_u} \left\{ \rho \|X\|_* + \frac{\mu}{2} \|x_u\|_2^2 + \frac{\lambda}{2} \|X - C\|_F^2 + \langle b, y \rangle \right. \\
&\quad \left. - \langle \mathcal{A}^e(X), y^e \rangle - \langle \mathcal{A}^l(X), y^l \rangle - \langle \mathcal{A}^q(X), y^q \rangle - \langle \mathcal{A}^u(X) - x_u, y^u \rangle \right\} \\
&= \inf_{X, x_u} \left\{ \rho \|X\|_* + \frac{\lambda}{2} (\|X\|_F^2 - 2 \langle \frac{1}{\lambda} \mathcal{W}^* y + C, X \rangle + \|\frac{1}{\lambda} \mathcal{W}^* y + C\|_F^2) \right. \\
&\quad \left. - \frac{\lambda}{2} \|\frac{1}{\lambda} \mathcal{W}^* y + C\|_F^2 + \frac{\lambda}{2} \|C\|_F^2 + \langle b, y \rangle + \frac{\mu}{2} \|x_u\|_2^2 + \langle x_u, y^u \rangle \right\} \\
&= \inf_{X, x_u} \left\{ \rho \|X\|_* + \frac{\lambda}{2} \|X - C - \frac{1}{\lambda} \mathcal{W}^* y\|_F^2 - \frac{\lambda}{2} \|\frac{1}{\lambda} \mathcal{W}^* y + C\|_F^2 \right. \\
&\quad \left. + \frac{\lambda}{2} \|C\|_F^2 + \langle b, y \rangle + \frac{\mu}{2} \|x_u\|_2^2 + \langle x_u, y^u \rangle \right\} \\
&= \inf_X \left\{ \rho \|X\|_* + \frac{\lambda}{2} \|X - C - \frac{1}{\lambda} \mathcal{W}^* y\|_F^2 \right\} - \frac{\lambda}{2} \|\frac{1}{\lambda} \mathcal{W}^* y + C\|_F^2 \\
&\quad + \frac{\lambda}{2} \|C\|_F^2 + \langle b, y \rangle - \frac{1}{2\mu} \|y^u\|_2^2,
\end{aligned}$$

where $y = [y^e; y^l; y^q; y^u]$, and $\mathcal{W}^* = [\mathcal{A}^{e*} \ \mathcal{A}^{l*} \ \mathcal{A}^{q*} \ \mathcal{A}^{u*}]$ is the adjoint operator of \mathcal{W} .

In order to get the infimum of $\rho \|X\|_* + \frac{\lambda}{2} \|X - C - \frac{1}{\lambda} \mathcal{W}^* y\|_F^2$ in $g(y)$, we need to introduce the singular value shrinkage operator $\mathcal{D}_\tau(\cdot)$. Let $X \in \mathbb{R}^{n_1 \times n_2}$ have the singular value decomposition (SVD) such that

$$X = U \Sigma V_1^T, \quad \Sigma = \text{diag}(\{\sigma_i\}_{1 \leq i \leq n_1}),$$

where $\sigma_1 \geq \dots \geq \sigma_{n_1} \geq 0$ are singular values of X . For any $\tau \geq 0$, $\mathcal{D}_\tau(X)$ is defined by:

$$\mathcal{D}_\tau(X) := U \mathcal{D}_\tau(\Sigma) V_1^T, \quad \mathcal{D}_\tau(\Sigma) = \text{diag}(\{(\sigma_i - \tau)_+\}),$$

where $t_+ := \max(0, t)$. The singular value thresholding operator is a proximity operator associated with nuclear norm. Details of proximity operator can be found

in [9]. The following proposition¹ allows us to obtain the result of $\inf_X \{\rho \|X\|_* + \frac{\lambda}{2} \|X - C - \frac{1}{\lambda} \mathcal{W}^* y\|_F^2\}$. Its proof can be found in [2, 12].

Proposition 2.1.1. For each $\tau \geq 0$ and $Y \in \Re^{n_1 \times n_2}$, the singular value thresholding operator obeys

$$\mathcal{D}_\tau(Y) = \arg \min_X \left\{ \frac{1}{2} \|X - Y\|_F^2 + \tau \|X\|_* \right\}. \quad (2.2)$$

□

Proposition 2.1.1 implies that

$$\begin{aligned} g(y) &= \rho \|\mathcal{D}_{\frac{\rho}{\lambda}}(C + \frac{1}{\lambda} \mathcal{W}^* y)\|_* + \frac{\lambda}{2} \|\mathcal{D}_{\frac{\rho}{\lambda}}(C + \frac{1}{\lambda} \mathcal{W}^* y) - C - \frac{1}{\lambda} \mathcal{W}^* y\|_F^2 \\ &\quad - \frac{\lambda}{2} \|C + \frac{1}{\lambda} \mathcal{W}^* y\|_F^2 + \frac{\lambda}{2} \|C\|_F^2 + \langle b, y \rangle - \frac{1}{2\mu} \|y^u\|_2^2 \\ &= -\frac{\lambda}{2} \|\mathcal{D}_{\frac{\rho}{\lambda}}(C + \frac{1}{\lambda} \mathcal{W}^* y)\|_F^2 + \frac{\lambda}{2} \|C\|_F^2 + \langle b, y \rangle - \frac{1}{2\mu} \|y^u\|_2^2. \end{aligned}$$

Let

$$\theta(y) := -g(y) = \frac{\lambda}{2} \|\mathcal{D}_{\frac{\rho}{\lambda}}(C + \frac{1}{\lambda} \mathcal{W}^* y)\|_F^2 + \frac{1}{2\mu} \|y^u\|_2^2 - \langle b, y \rangle - \frac{\lambda}{2} \|C\|_F^2.$$

Then we obtain the dual problem (D) ,

$$(D) \quad \begin{array}{ll} \min & \theta(y) \\ \text{s.t.} & y \in Q^+ \end{array}$$

The objective function θ in the dual problem (D) is a continuously differentiable convex function. However it is not twice continuously differentiable. Its first order derivative is given by

$$\nabla \theta(y) = \mathcal{W} \mathcal{D}_{\frac{\rho}{\lambda}}(C + \frac{1}{\lambda} \mathcal{W}^* y) + \frac{1}{\mu} T(y^u) - b, \quad (2.3)$$

where $T(y^u) = [\mathbf{0}; \mathbf{0}; \mathbf{0}; y^u]$.

¹Donald Goldfarb first reported the formula (2.2) at the "Foundations of Computational Mathematics Conference'08" held at the City University of Hong Kong, June 2008

The dual problem (D) of problem (P) is a convex constrained vector-valued problem, in contrast to the matrix-valued problem (P) . When it is easier to apply optimization algorithms to solve for solutions for (D) than for (P) , one can use Rockafellar's dual approach [17] to find an optimal solution \bar{y} for (D) first. An optimal solution \bar{X} for (P) can then be obtained by

$$(\bar{X}, \bar{x}_u) = \arg \inf_{X, x_u} L(X, x_u, \bar{y}) = (\mathcal{D}_{\frac{\rho}{\lambda}}(C + \frac{1}{\lambda} \mathcal{W}^* \bar{y}), -\mu^{-1} \bar{y}^u).$$

Before introducing optimality conditions, we assume that the Slater condition holds for the primal problem (P) :

$$\begin{cases} \{A_i\}_{i=1}^{m_e} \text{ are linearly independent,} \\ \exists (X_0, x_u^0) \in \mathcal{F} \text{ such that } \mathcal{W}(X_0) - T(x_u^0) \in b + ri(Q), \end{cases} \quad (2.4)$$

where $ri(Q)$ denotes the relative interior of Q . When the Slater condition is satisfied, the following proposition, which is a straightforward application of Rockafellar's results in [17], holds.

Proposition 2.1.2. Under the Slater condition (2.4), the following results hold:

(i) There exists at least one $\bar{y} \in Q^+$ that solves the dual problem (D) . The unique solution to the primal problem (P) is given by

$$(\bar{X}, \bar{x}_u) = (\mathcal{D}_{\frac{\rho}{\lambda}}(C + \frac{1}{\lambda} \mathcal{W}^* \bar{y}), -\mu^{-1} \bar{y}^u). \quad (2.5)$$

(ii) For every real number ε , the constrained level set $\{y \in Q^+ \mid \theta(y) \leq \varepsilon\}$ is closed, bounded and convex.

□

The convexity in the second part of Proposition 2.1.2 allows us to apply any gradient based optimization method to obtain an optimal solution for the dual problem (D) . When a solution is found for (D) , one can always use (2.5) to obtain a unique optimal solution to the primal problem (P) .

With respect to problem (D) , the Lagrange function may be defined by $L(y, \alpha) = \theta(y) - \langle \alpha, y \rangle$. For some Lagrange multiplier $\bar{\alpha}$, the Karush-Kuhn-Tucker conditions require the optimal solutions \bar{y} of problem (D) to satisfy:

$$\begin{aligned}\nabla L_y(\bar{y}, \bar{\alpha}) &= \nabla \theta(\bar{y}) - \bar{\alpha} = 0, \\ \bar{y} &\in Q^+, \quad -\bar{\alpha} \in \mathcal{N}_{Q^+}(\bar{y}),\end{aligned}$$

where $\mathcal{N}_{Q^+}(\bar{y})$ denotes the normal cone of Q^+ at \bar{y} . It implies that \bar{y} solves problem (D) if and only if it satisfies,

$$\langle y - \bar{y}, \nabla \theta(\bar{y}) \rangle \geq 0, \quad \forall y \in Q^+. \quad (2.6)$$

On the other hand, we define $F: \Re^m \rightarrow \Re^m$ by

$$F(y) := y - \Pi_{Q^+}(y - \nabla \theta(y)), \quad \forall y \in \Re^m. \quad (2.7)$$

It can be verified with the results from [4] that solving the variational inequality (2.6) is equivalent to solving the system of

$$F(y) = \mathbf{0}, \quad y \in \Re^m. \quad (2.8)$$

It is known that F is globally Lipschitz continuous but not everywhere continuously differentiable. One may use Clarke's generalized Jacobian based Newton's methods to solve problem (2.8). However those methods can not be globalized because F does not have any real-valued gradient mapping function. Nevertheless, the smoothing Newton-BiCGStab method has been shown to resolve such difficulty for the least squares semidefinite programming problems [6]. Similarly we may also introduce smoothing functions for the least squares nonsymmetric matrix nuclear problems and design a Newton-BiCGStab method for solving a smoothed system of (2.8).

2.2 The Differential Properties of the Smoothing Functions

Consider a real-valued nonsmooth function

$$f(t) = \max(0, t), \quad t \in \mathbb{R},$$

which we denote by $(t)_+$. $(t)_+$ is not differentiable at $t = 0$. The two smoothing functions used in this thesis for $(t)_+$ are the Huber function $\phi_H: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$,

$$\phi_H(\varepsilon, t) = \begin{cases} t, & \text{if } t \geq \frac{|\varepsilon|}{2} \\ \frac{1}{2|\varepsilon|}(t + \frac{|\varepsilon|}{2})^2, & \text{if } -\frac{|\varepsilon|}{2} < t < \frac{|\varepsilon|}{2} \\ 0, & \text{if } t \leq -\frac{|\varepsilon|}{2} \end{cases}; \quad (2.9)$$

and the Smale smoothing function $\phi_S: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$,

$$\phi_S(\varepsilon, t) = [t + \sqrt{\varepsilon^2 + t^2}]/2, \quad (\varepsilon, t) \in \mathbb{R} \times \mathbb{R}. \quad (2.10)$$

Discussions on the properties of the smoothing functions can be found in [16, 21]. The concept of semismoothness plays an important role in (quadratic) convergence analysis of generalized Newton methods for nonsmooth equations. It was introduced by Mifflin [13], and extended by Qi and Sun [16], for cases when a vector-valued function is not differentiable, but locally Lipschitz continuous.

Definition 2.2.1. Suppose that a vector-valued function $f: \mathbb{R}^{m_1} \rightarrow \mathbb{R}^{m_2}$ is locally Lipschitz continuous at $x \in \mathbb{R}^{m_1}$. f is said to be semismooth at x , if f is directionally differentiable at x ; and for any $V \in \partial f(x + \Delta x)$, the generalized Clarke Jacobian of f at $x + \Delta x$, f satisfies,

$$f(x + \Delta x) - f(x) - V(\Delta x) = o(\|\Delta x\|).$$

Furthermore, f is said to be strongly semismooth at x , if f is semismooth at x and for any $V \in \partial f(x + \Delta x)$, f satisfies,

$$f(x + \Delta x) - f(x) - V(\Delta x) = \mathcal{O}(\|\Delta x\|^2).$$

It has been known that both ϕ_H and ϕ_S are globally Lipschitz continuous, continuously differentiable around (ε, t) whenever $\varepsilon \neq 0$, and are strongly semismooth at $(0, t)$ (see [21] and references therein for details). The outer layer vector-valued functions defined in (2.7), when they are composite functions of $(t)_+$ and a linear function, can be smoothed by using a smoothing function either ϕ_H or ϕ_S . Under certain conditions, the smoothing functions inherit the Lipschitz continuity, differentiability, and semismoothness properties of either ϕ_H or ϕ_S . With respect to the inner layer of F in (2.7), where the singular value thresholding operator is involved, we will also show that the nonsymmetric matrix-valued functions can be smoothed by applying the smoothing function either ϕ_H or ϕ_S to the singular values of the matrix. The resulting matrix-valued function will be shown to inherit the related differential properties from ϕ_H (or ϕ_S). Since ϕ_H and ϕ_S share similar differential properties, in the following, unless we specify we will use ϕ to denote the smoothing function either ϕ_H or ϕ_S .

The function $F(y)$ in (2.7) is given by

$$F(y) = y - \Pi_{Q^+}(y - \mathcal{WD}_{\frac{\rho}{\lambda}}(C + \frac{1}{\lambda}\mathcal{W}^*y) - \frac{1}{\mu}T(y^u) + b), \quad (2.11)$$

where $T(y^u) = [\mathbf{0}; \mathbf{0}; \mathbf{0}; y^u]$. F contains a composition of two nonsmooth functions. In the outer layer, $\Pi_{Q^+}(\cdot)$ is a metric projection operator from \Re^m to Q^+ . $\Pi_{Q^+}(\cdot)$ is given by

$$\Pi_{Q^+}(z) = \begin{pmatrix} z_e \\ (z_l)_+ \\ \Pi_{\mathcal{K}^{m_q}}(z_q) \\ z_u \end{pmatrix}. \quad (2.12)$$

where $z = [z_e; z_l; z_q; z_u]$ and $\Pi_{\mathcal{K}^{m_q}}(z)$ denotes the projection of z onto the second-order cone \mathcal{K}^{m_q} . The properties of second order cone have been well studied. The

following well known proposition gives an analytical solution to $\Pi_{\mathcal{K}^n}(\cdot)$, the metric projection onto a second order cone \mathcal{K}^n of dimension n . See [14] and references therein for more discussions on $\Pi_{\mathcal{K}^{m_q}}(\cdot)$.

Proposition 2.2.1. For any $z \in \Re^n$, let $z = [z^t; z_n]$ where $z^t \in \Re^{n-1}$ and $z_n \in \Re$. Then z has the following spectral decomposition

$$\begin{aligned} z &= \lambda_1(z)c_1(z) + \lambda_2(z)c_2(z), \\ \lambda_i(z) &= z_n + (-1)^i \|z^t\|_2, \\ c_i(z) &= \begin{cases} \frac{1}{2}((-1)^i \frac{z^t}{\|z^t\|_2}, 1)^T, & \text{if } z^t \neq 0 \\ \frac{1}{2}((-1)^i w, 1)^T, & \text{if } z^t = 0 \end{cases}, \end{aligned}$$

where $w \in \Re^{n-1}$ satisfies $\|w\|_2 = 1$. Then $\Pi_{\mathcal{K}^n}(z)$ is given by

$$\Pi_{\mathcal{K}^n}(z) = (\lambda_1(z))_+ c_1(z) + (\lambda_2(z))_+ c_2(z).$$

□

With Proposition 2.2.1, for $\Pi_{\mathcal{K}^n}(\cdot)$, we may introduce a smoothing function $\phi_{\mathcal{K}^n}$ associated with \mathcal{K}^n ,

$$\phi_{\mathcal{K}^n}(\varepsilon, z) = \phi(\varepsilon, \lambda_1(z))c_1(z) + \phi(\varepsilon, \lambda_2(z))c_2(z).$$

It has been shown in [21, Theorem 5.1] that $\phi_{\mathcal{K}^n}(\cdot, \cdot)$ is globally Lipschitz continuous, and strongly semismooth on $\Re_+ \times \Re^n$, if the smoothing function ϕ is globally Lipschitz continuous, and strongly semismooth. Furthermore, a smoothing function $\psi : \Re \times \Re^m \rightarrow \Re^m$ for the outer layer of metric projector (2.12) may now be defined by,

$$\psi(\varepsilon, z) = \begin{pmatrix} z_e \\ \phi(\varepsilon, z_l) \\ \phi_{\mathcal{K}^{m_q}}(\varepsilon, z_q) \\ z_u \end{pmatrix}. \quad (2.13)$$

With the above known results, ψ is a globally Lipschitz continuous, and strongly semismooth function on $\mathbb{R} \times \mathbb{R}^m$.

Next we will construct a smoothing function for the inner layer on the nonsymmetric matrix operator $\mathcal{D}_\lambda^\rho(\cdot)$. Let $X \in \mathbb{R}^{n_1 \times n_2}$, and $n_1 \leq n_2$. Suppose that X has the following SVD

$$X = U[\Sigma \ 0]V^T = U \begin{pmatrix} \sigma_1 & \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} & \vdots \\ 0 & \mathbf{0} & \sigma_{n_1} & \mathbf{0} \end{pmatrix} [V_1 \ V_2]^T. \quad (2.14)$$

In order to properly define the smoothing function for nonsymmetric matrix-valued functions, we will transform a nonsymmetric matrix into a symmetric matrix and make use of the known properties of the symmetric matrix-valued functions. Given the SVD of X , we let a symmetric matrix $Y_X \in S^{(n_1+n_2) \times (n_1+n_2)}$ be defined by

$$Y_X = \begin{pmatrix} \mathbf{0} & X \\ X^T & \mathbf{0} \end{pmatrix}.$$

Let

$$P_Y = \frac{1}{\sqrt{2}} \begin{pmatrix} U & U & \mathbf{0} \\ V_1 & -V_1 & \sqrt{2}V_2 \end{pmatrix}, \quad (2.15)$$

where $U \in \mathbb{R}^{n_1 \times n_1}$, $V_1 \in \mathbb{R}^{n_2 \times n_1}$ and $V_2 \in \mathbb{R}^{n_2 \times (n_2-n_1)}$. Then Y_X has the following eigenvalue decomposition:

$$Y_X = P_Y \begin{pmatrix} \Sigma & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\Sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} P_Y^T.$$

For some $\beta > 0$, we define a real-valued function g_β and a corresponding matrix-valued function $G_\beta(Y_X): S^{(n_1+n_2) \times (n_1+n_2)} \rightarrow S^{(n_1+n_2) \times (n_1+n_2)}$ such that

$$g_\beta(t) \quad : \quad = \quad (t - \beta)_+ - (-t - \beta)_+, \quad (2.16)$$

$$G_\beta(Y_X) \quad : \quad = \quad (Y_X - \beta I)_+ - (-Y_X - \beta I)_+. \quad (2.17)$$

Here I denotes an identity matrix of dimension $(n_1 + n_2)$ and the matrix-valued operator $(\cdot)_+$ is the metric projection $\Pi_{S_+^n}(\cdot)$ onto the symmetric positive semidefinite cone. Then one can check [10] that

$$G_\beta(Y_X) = P_Y \begin{pmatrix} g_\beta(\sigma_1) & 0 & 0 & 0 & \dots & 0 & \mathbf{0} \\ 0 & \ddots & 0 & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & g_\beta(\sigma_{n_1}) & 0 & \dots & \vdots & \vdots \\ 0 & \dots & 0 & g_\beta(-\sigma_1) & 0 & \vdots & \vdots \\ \vdots & \vdots & \vdots & 0 & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & \dots & g_\beta(-\sigma_{n_1}) & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \dots & \dots & \mathbf{0} & \mathbf{0} \end{pmatrix} P_Y^T.$$

Applying the transformation functions (2.16) and (2.17) onto the singular value shrinkage operator $\mathcal{D}_{\frac{\rho}{\lambda}}(X) = U[\text{Diag}((\sigma_i - \frac{\rho}{\lambda})_+) \mathbf{0}][V_1 \ V_2]^T$ for $X \in \Re^{n_1 \times n_2}$, we have that

$$G_{\frac{\rho}{\lambda}}(Y_X) = \begin{pmatrix} \mathbf{0} & \mathcal{D}_{\frac{\rho}{\lambda}}(X) \\ \mathcal{D}_{\frac{\rho}{\lambda}}(X)^T & \mathbf{0} \end{pmatrix}. \quad (2.18)$$

As a result of (2.18), the smoothing functions ϕ_g for g_β and Φ_G for G_β may be defined, respectively by

$$\phi_g(\varepsilon, t) \quad : \quad = \quad \phi(\varepsilon, t - \beta) - \phi(\varepsilon, -t - \beta) \quad (2.19)$$

and

$$\Phi_G(\varepsilon, Y_X) := P_Y \begin{pmatrix} \phi_g(\varepsilon, \sigma_1) & 0 & 0 & 0 & \dots & 0 & \mathbf{0} \\ 0 & \ddots & 0 & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \phi_g(\varepsilon, \sigma_{n_1}) & 0 & \dots & \vdots & \vdots \\ 0 & \dots & 0 & \phi_g(\varepsilon, -\sigma_1) & 0 & \vdots & \vdots \\ \vdots & \vdots & \vdots & 0 & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & \dots & \phi_g(\varepsilon, -\sigma_{n_1}) & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \dots & \dots & 0 & \mathbf{0} \end{pmatrix} P_Y^T.$$

One can easily derive that Φ_G has the following form

$$\Phi_G(\varepsilon, Y_X) = \begin{pmatrix} \mathbf{0} & \Phi_{\mathcal{D}_{\frac{\rho}{\lambda}}}(\varepsilon, X) \\ (\Phi_{\mathcal{D}_{\frac{\rho}{\lambda}}}(\varepsilon, X))^T & \mathbf{0} \end{pmatrix},$$

where

$$\Phi_{\mathcal{D}_{\frac{\rho}{\lambda}}}(\varepsilon, X) := U[\text{Diag}(\phi_g(\varepsilon, \sigma_i)) \ \mathbf{0}][V_1 \ V_2]^T. \quad (2.20)$$

We have known that the smoothing function (2.13) for the outer layer of F in (2.7) is strongly semismooth at $(0, y)$. Next we will show the strong semismoothness of $\Phi_{\mathcal{D}_{\beta}}$, which is a smoothing function for the inner layer of F .

Let $\Delta X \in \Re^{m \times n}$ and

$$H = \begin{pmatrix} \mathbf{0} & \Delta X \\ \Delta X^T & \mathbf{0} \end{pmatrix}. \quad (2.21)$$

For any $Y \in S^n$, $\lambda(Y) \in \Re^n$ denotes the vector of eigenvalues of Y . Let $Y = P \text{diag}(\lambda(Y)) P^T$ be the eigenvalue decomposition of Y . A Löwner function $F: S^n \rightarrow S^n$ is then defined with respect to a real-valued function $f(\cdot)$,

$$F(Y) := P \text{diag}[f(\lambda_1(Y)), f(\lambda_2(Y)), \dots, f(\lambda_n(Y))] P^T. \quad (2.22)$$

When f is differentiable at μ , a first divided difference function $F^{[1]}$ at $\mu \in \mathfrak{R}^n$ is defined by

$$(F^{[1]}(\mu))_{i,j} = \begin{cases} \frac{f(\mu_i) - f(\mu_j)}{\mu_i - \mu_j}, & \text{if } \mu_i \neq \mu_j \\ f'(\mu_i), & \text{if } \mu_i = \mu_j \end{cases}. \quad (2.23)$$

With the results of Löwner (see [1] for details), we have the following lemma.

Lemma 2.2.1. If a real-valued function $f(\cdot)$ is continuously differentiable in an open interval (a_1, a_2) containing all the eigenvalues $\{\lambda_i(Y)\}$ of Y , then the Löwner function $F(\cdot)$ is differentiable at Y . For any $H \in S^n$, the derivative of $F(\cdot)$ is given by

$$F'(Y)H = P(F^{[1]}(\lambda(Y)) \circ (P^T H P))P^T.$$

□

With Lemma 2.2.1, we have that Φ_G is differentiable at (ε, Y_X) for any $\varepsilon > 0$, and its derivative is given by

$$\begin{aligned} (\Phi_G)'_Y(\varepsilon, Y_X)H &= P_Y(\Omega(\varepsilon, \lambda(Y_X)) \circ (P_Y^T H P_Y))P_Y^T, \\ (\Phi_G)'_\varepsilon(\varepsilon, Y_X) &= P_Y \text{diag}(\phi'_\varepsilon(\varepsilon, \lambda_1(Y_X)), \dots, \phi'_\varepsilon(\varepsilon, \lambda_{n_1+n_2}(Y_X)))P_Y^T, \end{aligned} \quad (2.24)$$

where P_Y is the same as in (2.15). $\Omega(\varepsilon, \lambda(Y_X))$ is the first divided difference matrix of Φ_G at $\lambda(\Phi_G(\varepsilon, Y_X))$ such that

$$\lambda(\Phi_G(\varepsilon, Y_X)) = [\phi_g(\varepsilon, \sigma_1); \dots \phi_g(\varepsilon, \sigma_{n_1}); \phi_g(\varepsilon, -\sigma_1); \dots \phi_g(\varepsilon, -\sigma_{n_1}); 0; \dots; 0],$$

and $\sigma = [\sigma_1; \dots; \sigma_{n_1}]$ are singular values of X . Since

$$P_Y = \frac{1}{\sqrt{2}} \begin{pmatrix} U & U & \mathbf{0} \\ V_1 & -V_1 & \sqrt{2}V_2 \\ n_1 & n_1 & n_2 \end{pmatrix} \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}, \quad (2.25)$$

we divide $\Omega(\varepsilon, \lambda(Y_X))$ into nine parts,

$$\Omega(\varepsilon, \lambda(Y_X)) = \Omega_{\mathcal{D}}(\varepsilon, \sigma) = \begin{pmatrix} \Omega_{11} & \Omega_{12} & \Omega_{13} \\ \Omega_{21} & \Omega_{22} & \Omega_{23} \\ \Omega_{31} & \Omega_{32} & \Omega_{33} \end{pmatrix} \begin{matrix} n_1 \\ n_1 \\ n_2 - n_1 \end{matrix} \quad \begin{matrix} n_1 \\ n_1 \\ n_2 - n_1 \end{matrix}.$$

It can be verified that $\Omega_{11} = \Omega_{22}$, $\Omega_{13} = \Omega_{23}$. Explicitly for (2.24), we have that

$$\begin{aligned} & (\Phi_G)'_Y(\varepsilon, Y_X)H \\ &= P_Y(\Omega_{\mathcal{D}}(\varepsilon, \sigma) \circ \frac{1}{2} \begin{pmatrix} U^T & V_1^T \\ U^T & -V_1^T \\ \mathbf{0} & \sqrt{2}V_2^T \end{pmatrix} \begin{pmatrix} \mathbf{0} & \Delta X \\ \Delta X^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} U & U & \mathbf{0} \\ V_1 & -V_1 & \sqrt{2}V_2 \end{pmatrix})P_Y^T \\ &= \frac{1}{2}P_Y \begin{pmatrix} (A^T + A) \circ \Omega_{11} & (A^T - A) \circ \Omega_{12} & \sqrt{2}B \circ \Omega_{13} \\ (A - A^T) \circ \Omega_{21} & -(A^T + A) \circ \Omega_{22} & \sqrt{2}B \circ \Omega_{23} \\ \sqrt{2}B^T \circ \Omega_{31} & \sqrt{2}B^T \circ \Omega_{32} & \mathbf{0} \end{pmatrix} P_Y^T \\ &= \begin{pmatrix} \mathbf{0} & P_{12} \\ P_{12}^T & \mathbf{0} \end{pmatrix}, \end{aligned}$$

where A stands for $U^T \Delta X V_1$, B stands for $U^T \Delta X V_2$, and

$$\begin{aligned} P_{12} &= (\Phi_{\mathcal{D}_\beta})'_X(\varepsilon, X) \Delta X \\ &= \frac{1}{2}U((A + A^T) \circ \Omega_{11} + (A - A^T) \circ \Omega_{12})V_1^T + U(B \circ \Omega_{13})V_2^T. \end{aligned}$$

Similarly to (2.24), we have that $\Phi_{\mathcal{D}_\beta}$ is differentiable at (ε, X) when $\varepsilon > 0$, and its derivative is given by

$$\begin{aligned} (\Phi_{\mathcal{D}_\beta})'_X(\varepsilon, X) \Delta X &= \frac{1}{2}U((A + A^T) \circ \Omega_{11} + (A - A^T) \circ \Omega_{12})V_1^T + U(B \circ \Omega_{13})V_2^T, \\ (\Phi_{\mathcal{D}_\beta})'_\varepsilon(\varepsilon, X) &= U[\text{diag}(\phi'_\varepsilon(\varepsilon, \sigma_i(X) - \beta)) \mathbf{0}][V_1 \ V_2]^T, \end{aligned} \tag{2.26}$$

The function Φ_G in (2.17) is a symmetric matrix-valued function where eigenvalues of the matrix are given by ϕ_g , which is a sum of two strongly semismooth

functions. The sum of two strongly semismooth functions is also strongly semismooth. From the results of [21], we know that the smoothing matrix-valued function Φ_G inherits the globally Lipschitz continuous and strong semismoothness of ϕ_g . We have seen from above that the derivative of $\Phi_{\mathcal{D}_\beta}$ has an analogous transformation form to the derivative of Φ_G as from X to Y_X . Thus $\Phi_{\mathcal{D}_\beta}$ analogously inherit the globally Lipschitz continuous and strongly semismooth properties at any $(0, X) \in \Re \times \Re^{n_1 \times n_2}$. In particular, for any $\Delta X \rightarrow \mathbf{0}$ and $\varepsilon \rightarrow 0$ and $V \in \partial\Phi_{\mathcal{D}_\beta}(\varepsilon, X + \Delta X)$,

$$\Phi_{\mathcal{D}_\beta}(\varepsilon, X + \Delta X) - \Phi_{\mathcal{D}_\beta}(0, X) - V(\varepsilon, \Delta X) = \mathcal{O}(\|(\varepsilon, \Delta X)\|^2). \quad (2.27)$$

Now we are ready to introduce a smoothing function $\Upsilon: \Re \times \Re^m \rightarrow \Re^m$ for F defined in (2.7) with (2.13) and (2.20),

$$\Upsilon(\varepsilon, y) := y - \psi(\varepsilon, y - \mathcal{W}\Phi_{\mathcal{D}_\beta}(\varepsilon, C + \frac{1}{\lambda}\mathcal{W}^*y) - \frac{1}{\mu}T(y^u) + b), \quad (2.28)$$

where $T(y^u) = [\mathbf{0}; \mathbf{0}; \mathbf{0}; y^u]$.

The differential properties of Υ , which will be used for the convergence analysis of our algorithm, are summarized in the following proposition.

Proposition 2.2.2. Let $\Upsilon: \Re \times \Re^m$ be defined by (2.28). Let $y \in \Re^m$. Then it holds that

- (i) Υ is globally Lipschitz continuous on $\Re \times \Re^m$.
- (ii) Υ is continuously differentiable around (ε, y) where $\varepsilon \neq 0$. If $m_q = 0$, then any fixed $\varepsilon \in \Re$, $\Upsilon(\varepsilon, \cdot)$ is P_0 -function, i.e. for any $y, h \in \Re^m$ with $y \neq h$, it holds that

$$\max_{y_i \neq h_i} (y_i - h_i)(\Upsilon_i(\varepsilon, y) - \Upsilon_i(\varepsilon, h)) \geq 0. \quad (2.29)$$

- (iii) Υ is strongly semismooth at $(0, y)$. In particular, for any $\varepsilon \downarrow 0$, and $h \in \Re^m$,

$h \rightarrow 0$ we have that

$$\Upsilon(\varepsilon, y + h) - \Upsilon(0, y) - \Upsilon'(\varepsilon, y + h) \begin{pmatrix} \varepsilon \\ h \end{pmatrix} = \mathcal{O}(\|(\varepsilon, h)\|^2).$$

(iv) For any $h \in \mathfrak{R}^m$,

$$\partial_B \Upsilon(0, y)(0, h) \subseteq h - \partial_B \psi(0, y - \nabla \theta(y))(0, z), \quad (2.30)$$

where $z = h - \frac{1}{\lambda} \mathcal{W} \partial_B \Phi_{\mathcal{D}_\beta}(0, C + \frac{1}{\lambda} \mathcal{W}^* y)(0, \mathcal{W}^* h) - \frac{1}{\mu} T(h^u)$ and $T(h^u) = [\mathbf{0}; \mathbf{0}; \mathbf{0}; h^u]$.

Proof. (i) Since both ψ and $\Phi_{\mathcal{D}_\beta}$ are globally Lipschitz continuous, Υ is also globally Lipschitz continuous.

(ii) From the definitions of ψ (2.13), Φ (2.20) and Υ (2.28), we know that ψ is continuously differentiable for any $(\varepsilon, y) \in \mathfrak{R} \times \mathfrak{R}^m$ when $\varepsilon \neq 0$. For any $\varepsilon \neq 0$. Define $g_\varepsilon: \mathfrak{R}^m \rightarrow \mathfrak{R}^m$ such that

$$g_\varepsilon(y) = \mathcal{W} \Phi_{\mathcal{D}_\beta}(\varepsilon, C + \frac{1}{\lambda} \mathcal{W}^* y) + \frac{1}{\mu} T(y^u) - b, \quad y \in \mathfrak{R}^m,$$

where $T(y^u) = [\mathbf{0}; \mathbf{0}; \mathbf{0}; y^u]$, g_ε is continuously differentiable on \mathfrak{R}^m . Furthermore, we have that

$$\begin{aligned} \langle h, (g_\varepsilon)'(y)h \rangle &= \langle h, \mathcal{W}(\Phi_{\mathcal{D}_\beta})'_X(\varepsilon, X) \left(\frac{1}{\lambda} (\mathcal{W}^* h) \right) + \frac{1}{\mu} T(h^u) \rangle \\ &= \frac{1}{\lambda} \langle \mathcal{W}^* h, (\Phi_{\mathcal{D}_\beta})'_X(\varepsilon, X) (\mathcal{W}^* h) \rangle + \frac{1}{\mu} (T(h^u) \cdot T(h^u)) \\ &= \frac{1}{2\lambda} \langle Z, P_Y [\Omega(\varepsilon, \sigma) \circ (P_Y^T Z P_Y)] P_Y^T \rangle + \frac{1}{\mu} \langle h^u, h^u \rangle \\ &\geq 0, \end{aligned}$$

where $X = C + \frac{1}{\lambda} \mathcal{W}^* y$, $T(h^u) = [\mathbf{0}; \mathbf{0}; \mathbf{0}; h^u]$ and

$$Z := \begin{bmatrix} 0 & \mathcal{W}^* h \\ (\mathcal{W}^* h)^T & 0 \end{bmatrix}.$$

This implies that g_ε is a P_0 function on \mathfrak{R}^m . Let $y, h \in \mathfrak{R}^m$ with $y \neq h$. Then there exists $i \in \{1, \dots, m\}$ with $y_i \neq h_i$ such that

$$(y_i - h_i)((g_\varepsilon)_i(y) - (g_\varepsilon)_i(h)) \geq 0.$$

Noted that $m_q = 0$, then for any $z \in \mathfrak{R}^m$,

$$\psi'_{z_i}(\varepsilon, z_i) \in [0, 1], i = 1, \dots, m.$$

As a result,

$$(y_i - h_i)(\Upsilon_i(\varepsilon, y) - \Upsilon_i(\varepsilon, h)) \geq 0.$$

Thus Υ is a P_0 -function and (2.29) holds for any $y, h \in \mathfrak{R}^m$ such that $y \neq h$.

(iii) We have shown that the smoothing functions ψ defined in (2.13) is strongly semismooth at any $(0, y) \in \mathfrak{R} \times \mathfrak{R}^m$; and $\Phi_{\mathcal{D}_{\frac{\rho}{\lambda}}}$ defined in (2.13) is strongly semismooth at any $(0, X) \in \mathfrak{R} \times \mathfrak{R}^{n_1 \times n_2}$. With the known result that a composite function of strongly semismooth function is also strongly semismooth [5], we can conclude that Υ is strongly semismooth at $(0, y)$.

(iv) Both ψ and $\Phi_{\mathcal{D}_\beta}$ are directionally differentiable. For any $(\varepsilon, y') \in \mathfrak{R} \times \mathfrak{R}^m$ such that Υ is Fréchet differentiable at (ε, y') , the directional derivative gives that

$$\Upsilon'((\varepsilon, y'); (0, h)) = h - \psi'((\varepsilon, z'); (0, h - \frac{1}{\lambda} \mathcal{W} \Phi'_{\mathcal{D}_\beta}((\varepsilon, \frac{1}{\lambda} \mathcal{W}^* y'); (0, \mathcal{W}^* y)) - \frac{1}{\mu} T(h^u)),$$

where $T(h^u) = [\mathbf{0}; \mathbf{0}; \mathbf{0}; h^u]$ and $z' = y' - \nabla \theta(y')$. With the semismoothness of ψ and $\Phi_{\mathcal{D}_\beta}$, it implies that

$$\Upsilon'((\varepsilon, y'); (0, h)) \in h - \partial_B \psi((\varepsilon, z'); (0, h - \frac{1}{\lambda} \mathcal{W} \partial_B \Phi_{\mathcal{D}_\beta}((\varepsilon, \frac{1}{\lambda} \mathcal{W}^* y'); (0, \mathcal{W}^* y)) - \frac{1}{\mu} T(h^u)).$$

By taking $(\varepsilon, y') \rightarrow (0, y)$, we obtain (2.30).

□

Note that in (ii) of the Proposition 2.2.2, we assume $m_q = 0$ in order to prove that for any $\varepsilon \neq 0$, $\Upsilon(\varepsilon, \cdot)$ is P_0 -function. When $m_q > 0$, this conclusion may not hold. However, it is possible to show that for any $\varepsilon \neq 0$, $\Upsilon(\varepsilon, \cdot)$ is a generalized P_0 -function and that for any $y \in \Re^m$, $\Upsilon'_y(\varepsilon, y)$ is a quasi P_0 -matrix, using the techniques introduced in [19]. For simplicity, we omit the details here.

A Smoothing Newton-BiCGStab Method

Recall that in the system of equations $F(y) = \mathbf{0}$ in (2.7), F is defined by

$$F(y) = y - \Pi_{Q^+}(y - \nabla\theta(y)), \quad y \in \Re^m. \quad (3.1)$$

The function F is globally Lipschitz continuous but not everywhere continuously differentiable. A smoothing function Υ has been defined in the last chapter by,

$$\Upsilon(\varepsilon, y) := y - \psi(\varepsilon, y - \mathcal{W}\Phi_{\mathcal{D}_{\mathcal{K}}}(\varepsilon, C + \frac{1}{\lambda}\mathcal{W}^*y) - \frac{1}{\mu}T(y^u) + b), \quad (3.2)$$

where $T(y^u) = [\mathbf{0}; \mathbf{0}; \mathbf{0}; y^u]$.

Let $\kappa \in (0, \infty)$ be a constant. Define $G: \Re \times \Re^m \rightarrow \Re^m$ by

$$G(\varepsilon, y) := \Upsilon(\varepsilon, y) + \kappa|\varepsilon|y, \quad (\varepsilon, y) \in \Re \times \Re^m. \quad (3.3)$$

From Proposition 2.2.2, we know that for any $(\varepsilon, y) \in \Re \times \Re^m$ with $\varepsilon \neq 0$, $\Upsilon(\varepsilon, \cdot)$ is P_0 -function. It implies that $\Upsilon'_y(\varepsilon, y)$ is only a P_0 -matrix, which may be singular. The term $\kappa|\varepsilon|y$ is used to avoid the singularity of $\Upsilon(\varepsilon, \cdot)$. In order to solve $F(y) = 0$, a system of smoothing equations,

$$E(\varepsilon, y) = \mathbf{0}, \quad (\varepsilon, y) \in \Re \times \Re^m \quad (3.4)$$

is constructed, where $E : \Re \times \Re^m \rightarrow \Re \times \Re^m$, is defined by

$$E(\varepsilon, y) := \begin{bmatrix} \varepsilon \\ G(\varepsilon, y) \end{bmatrix} = \begin{bmatrix} \varepsilon \\ \Upsilon(\varepsilon, y) + \kappa|\varepsilon|y \end{bmatrix}. \quad (3.5)$$

For any $(0, \bar{y})$ satisfying the system of smoothing equations $E(\varepsilon, y) = \mathbf{0}$, \bar{y} is also a solution of the system of $F(y) = \mathbf{0}$. Because of the differential properties which have been summarized in Proposition 2.2.2, the following algorithm introduced in Gao and Sun [6] can be used to solve the system (3.4).

Define a merit function $\varphi : \Re \times \Re^m \rightarrow \Re_+$ such that

$$\varphi(\varepsilon, y) := \|E(\varepsilon, y)\|^2. \quad (3.6)$$

Algorithm 3.1: A Smoothing Newton-BiCGStab Method

1. Set $k = 0$. A scaler r is chosen to be $r \in (0, 1)$. Let $\eta \in (0, 1)$ be such that

$$\delta := \sqrt{2} \max\{r\hat{\varepsilon}, \eta\} < 1.$$

Select constants $\rho \in (0, \frac{1}{2})$, $\sigma \in (0, \frac{1}{2})$, $\tau \in (0, 1)$, and $\hat{\tau} \in [1, \infty)$. Let $\varepsilon^0 := \hat{\varepsilon}$ and $y^0 \in \Re^m$ be an arbitrary starting point.

2. If $E(\varepsilon^k, y^k) = 0$, then stop. Otherwise, compute

$$\zeta_k := r \min\{1, \varphi(\varepsilon^k, y^k)\} \quad \text{and} \quad \eta_k := \min\{\tau, \hat{\tau}\|E(\varepsilon^k, y^k)\|\}.$$

3. For an inexact Newton's direction, the BiCGStab iterative solver by Van der Vorst [20] is used to solve the following equation

$$E(\varepsilon^k, y^k) + E'(\varepsilon^k, y^k) \begin{bmatrix} \Delta\varepsilon^k \\ \Delta y^k \end{bmatrix} = \begin{bmatrix} \zeta_k \hat{\varepsilon} \\ 0 \end{bmatrix}, \quad (3.7)$$

approximately such that

$$\|R_k\| \leq \min\{\eta_k \|G(\varepsilon^k, y^k) + G'(\varepsilon^k, y^k)\Delta\varepsilon^k\|, \eta \|E(\varepsilon^k, y^k)\|\},$$

where

$$\Delta\varepsilon^k := -\varepsilon^k + \zeta_k \hat{\varepsilon},$$

and

$$R_k := G(\varepsilon^k, y^k) + G'(\varepsilon^k, y^k) \begin{bmatrix} \Delta\varepsilon^k \\ \Delta y^k \end{bmatrix}.$$

4. Line Search: Let l_k be the smallest nonnegative integer l satisfying

$$\varphi(\varepsilon^k + \rho^l \Delta\varepsilon^k, y^k + \rho^l \Delta y^k) \leq [1 - 2\rho(1 - \delta)\rho^l] \varphi(\varepsilon^k, y^k).$$

Then update the search point by,

$$(\varepsilon^{k+1}, y^{k+1}) = (\varepsilon^k + \rho^{l_k} \Delta\varepsilon^k, y^k + \rho^{l_k} \Delta y^k).$$

5. Let $k = k + 1$, and go to step 2.

□

Let \mathcal{N} be

$$\mathcal{N} := \{(\varepsilon, y) \mid \varepsilon \geq \eta(\varepsilon, y) \hat{\varepsilon}\}. \quad (3.8)$$

We have the following global and local convergence results for solving the system of smoothing equations (3.4).

Theorem 3.0.1. Let $E(\varepsilon, y)$ be defined by (3.5). Suppose that the Slater condition (2.4) holds. For the system (3.4), Algorithm 3.1 is well defined and generates a bounded infinite sequence $\{(\varepsilon^k, y^k)\} \in \mathcal{N}$ such that any accumulation point $(\bar{\varepsilon}, \bar{y})$ of $\{(\varepsilon^k, y^k)\}$ is a solution of $E(\varepsilon, y) = 0$.

Proof. This follows from Proposition 2.2.2, Theorem 4.1 in Gao and Sun [6], and the fact that the solution set to the dual problem (D) is bounded under the Slater condition.

□

Theorem 3.0.2. Let $E(\varepsilon, y)$ be defined by (3.5). Let $(\bar{\varepsilon}, \bar{y})$ be an accumulation point generated by Algorithm 3.1. If V is nonsingular for any $V \in \partial E(0, \bar{y})$, then the sequence $\{(\varepsilon^k, y^k)\}$ generated by Algorithm 3.1 converges to $(\bar{\varepsilon}, \bar{y})$ quadratically, i.e.,

$$\|(\varepsilon^{k+1} - \bar{\varepsilon}, y^{k+1} - \bar{y})\| = \mathcal{O}(\|(\varepsilon^k - \bar{\varepsilon}, y^k - \bar{y})\|^2). \quad (3.9)$$

Proof. This follows from that fact that Υ is strongly semismooth and [6, Theorem 4.5].

□

In the above Theorem 3.0.2, for the quadratic convergence of Algorithm 3.1, we need the nonsingularity of V for any $V \in \partial E(0, \bar{y})$. It is possible to verify that this assumption holds as in Theorem 4.5 in [6], if the constraint non-degenerate condition holds at \bar{X} . Again, we omit the details here.

Numerical Experiments

4.1 Implementation Issues

The least squares nonsymmetric matrix nuclear problem

$$\begin{aligned}
\min \quad & \rho \|X\|_* + \frac{\mu}{2} \|x_u\|_2^2 + \frac{\lambda}{2} \|X - C\|_F^2 \\
(NS) \quad \text{s.t.} \quad & \mathcal{W}(X) - T(x_u) \in b + Q, \\
& X \in \Re^{n_1 \times n_2}
\end{aligned} \tag{4.1}$$

has an analogous form of the least squares semidefinite programming problem

$$\begin{aligned}
\min \quad & \rho \langle X, I \rangle + \frac{\mu}{2} \|x_u\|_2^2 + \frac{\lambda}{2} \|X - C\|_F^2 \\
(S) \quad \text{s.t.} \quad & \mathcal{W}(X) - T(x_u) \in b + Q, \\
& X \in S_+^{n_1}.
\end{aligned} \tag{4.2}$$

In Chapter 2, we have seen that the dual objective function g_{NS} of (NS) is given by

$$g_{NS}(y) = -\frac{\lambda}{2} \|\mathcal{D}_{\frac{\rho}{\lambda}}(C + \frac{1}{\lambda} \mathcal{W}^* y)\|_F^2 - \frac{1}{2\mu} \|y^u\|_2^2 + \langle b, y \rangle + \frac{\lambda}{2} \|C\|_F^2. \tag{4.3}$$

One can verify that the dual objective function g_S for (S) is of the form,

$$g_S(y) = -\frac{\lambda}{2} \|\Pi_{S_+^n}(C + \frac{1}{\lambda} \mathcal{W}^* y - \frac{\rho}{\lambda} I)\|_F^2 - \frac{1}{2\mu} \|y^u\|_2^2 + \langle b, y \rangle + \frac{\lambda}{2} \|C\|_F^2. \tag{4.4}$$

Observe that g_{NS} and g_S have only one different term. Thus we may define a general operator $\mathcal{P}_\beta: \mathfrak{R}^{n_1 \times n_2} \rightarrow \mathfrak{R}^{n_1 \times n_2}$ with some $\beta \in \mathfrak{R}$ for both (NS) and (S) such that

$$\mathcal{P}_\beta(X) = \begin{cases} \mathcal{D}_\beta(X) & \text{for } (NS) \\ \Pi_{S_+^n}(X - \beta I) & \text{for } (S) \end{cases}. \quad (4.5)$$

Let

$$\theta(y): = -g_\star(y) = \frac{\lambda}{2} \|\mathcal{P}_{\frac{\rho}{\lambda}}(C + \frac{1}{\lambda} \mathcal{W}^* y)\|_F^2 + \frac{1}{2\mu} \|y^u\|_2^2 - \langle b, y \rangle - \frac{\lambda}{2} \|C\|_F^2.$$

The dual problem to be solved for both (NS) and (S) is of the form

$$\begin{aligned} \min \quad & \theta(y) \\ \text{s.t.} \quad & y \in Q^+. \end{aligned}$$

The $(LSCM)$ problems is a special case of (S) , where $\rho = 0$, $\mu = 0$, and only equality and inequality constraints are present. In [6], Gao and Sun's implementation has been shown to efficiently solve a special type of $(LSCM)$ problems, in which $\{\mathcal{A}_s^e, \mathcal{A}_s^l\}$ are of simple sparse forms. The analogy between (NS) and (S) indicates that both types of problems can be solved by the same algorithmic framework. For this thesis, an implementation *Smh-NewtonBICG.m* for solving both (NS) and (S) with general forms of $\{\mathcal{A}_*^e, \mathcal{A}_*^l, \mathcal{A}_*^q, \mathcal{A}_*^u\}$ has been rewritten in *Matlab*.

4.2 Numerical Experiments

The algorithm is implemented in *MATLAB R2009a*, with experiments running on Intel Core 2 Duo at 2.00 GHz CPU with RAM of 2GB. The code *Smh-NewtonBICG.m* reads in six inputs:

$$\{\rho, \mu, \lambda, C, W, \text{options}\},$$

where ρ , μ and λ are the scalar parameters to define the objective function of the problem. The matrix C has to be a structure variable with an element 'type' given by n or s to indicate the type of Problem (NS) or (S) respectively, another element 'dim' indicating the dimension and an element 'val' to store the matrix value of C . W is a structure array to define the four types of constraints. Each member in the array of W is a structure variable with four structure elements {'type', 'dim', 'A' and 'b'} which are used to define the form of some type of constraints. The 'type' with options in $\{e, q, l, u\}$ is used to indicate the type of a constraint. The 'dim' is used to define the dimension of the constraints. The 'A' is used to input the matrix form of \mathcal{W} . And 'b' is used for the vector form of b in Problem (NS) or (S).

The implementation *CaliMat.m* in [6] for solving covariance matrix problem has made use of the simple sparse forms of $\{\mathcal{A}_s^e, \mathcal{A}_s^l\}$. The elements in the operators $\{\mathcal{A}_s^e(\cdot), \mathcal{A}_s^l(\cdot)\}$ are referred by values at the nonzero components and their respective matrix indices. In this thesis, the code *Smh-NewtonBICG.m* extended the implementation to solve for problems with general forms of $\{\mathcal{A}_*^e, \mathcal{A}_*^l, \mathcal{A}_*^q, \mathcal{A}_*^u\}$. The constraint operator \mathcal{W} for Problem (NS) or (S) is defined by,

$$\begin{aligned}\mathcal{W}(X) &= [\langle A_1, X \rangle; \cdots; \langle A_m, X \rangle] \\ &= [\text{svec}(A_1)^T \text{svec}(X); \cdots; \text{svec}(A_m)^T \text{svec}(X)] \\ &= [\text{svec}(A_1) \cdots \text{svec}(A_m)]^T \text{svec}(X),\end{aligned}$$

where $\text{svec}(\cdot)$ is operator to transfer matrix X to a vector in which the elements are formed by stacking up the columns $\{x_1, x_2, \cdots, x_n\}$ of the matrix X . The matrix form of $\mathcal{W}(\cdot) = [\text{svec}(A_1) \cdots \text{svec}(A_m)]^T \text{svec}(\cdot)$ is given in the structure element 'A' in input W to *Smh-NewtonBICG.m*. When the dimensions (n_1, n_2) of the underlying matrix or the number of constraints are large, the difference in implementation on the forms of $\{\mathcal{A}_*^e, \mathcal{A}_*^l, \mathcal{A}_*^q, \mathcal{A}_*^u\}$ would affect the computational

cost of $\mathcal{W}(\cdot)$. The first three examples compare the computational time difference between *CaliMat.m* and *Smh-NewtonBICG.m*. The results are reported on an average of five experiments for each example. The Huber smoothing function, which was shown to be more efficient for symmetric matrix problems in [6], is used for the first four examples. In the last example, we will compare the performance between the use of Huber smoothing function (2.9) and the use of Smale smoothing function (2.10).

Example 4.2.1. Given a symmetric matrix C which is the 1-day correlation matrix of dimension (387) from the lagged data sets of RiskMetrics. We let $\rho = 1.0$, $\mu = 1.0$ and $\lambda = 0.0$. The index sets of the constraints are given by

$$\begin{aligned}\mathcal{B}^e &= \{(i, i) \mid i = 1, \dots, n_1\}, \\ \mathcal{B}^l &= \mathcal{B}^g = \mathcal{B}^u = \emptyset,\end{aligned}$$

and the constraint operator \mathcal{W} is given by

$$\mathcal{W}_k: \mathcal{E}^{i_k, i_k} = 1, \text{ for } (i_k, j_k) \in \mathcal{B}^e, \quad k \in 1, \dots, m_e.$$

Example 4.2.2. Let C be randomly generated with entries in the range of $[-1, 1]$ with uniform probability distribution. $\rho = 1.0$, $\lambda = 1.0$ and $\mu = 0.0$. The index set is the same as in Example 4.2.1. The constraint operator \mathcal{W} is given by

$$\mathcal{W}_k: \mathcal{E}^{i_k, i_k} = r_k, \text{ for } (i_k, j_k) \in \mathcal{B}^e, \quad k \in 1, \dots, m_e,$$

where $r_k \in [0, 1]$ is randomly generated. Table 4.2 is a comparison in cases of different dimensions (1) $n_1 = 500$, (2) $n_1 = 1000$ and (3) $n_1 = 2000$.

Example 4.2.3. Similar to Example 4.2.2 where C is randomly generated, let $\rho = 1.0$, $\lambda = 1.0$ and $\mu = 0.0$. The index sets for equality and linear inequality

constraints associated with $n_1 \times n_1$ matrices are given by

$$\begin{aligned}\mathcal{B}^{e_1} &= \{(i, i) \mid i = 1, \dots, n_1\}, \\ \mathcal{B}^{e_2} &= \{(i, j) \mid 1 \leq i < j \leq n_1\}, \\ \mathcal{B}^{l_u} &= \{(i, j) \mid 1 \leq i < j \leq n_1\}, \\ \mathcal{B}^{l_l} &= \{(i, j) \mid 1 \leq i < j \leq n_1\},\end{aligned}$$

where \mathcal{B}^{e_2} is the index set for fixed off-diagonal elements, \mathcal{B}^{l_u} and \mathcal{B}^{l_l} are index sets for off-diagonal elements to which an upper or lower bound are imposed respectively. They are randomly generated at each row of the matrix. The number of elements in \mathcal{B}^{e_2} , \mathcal{B}^{l_u} and \mathcal{B}^{l_l} are determined by parameters \hat{n}_{e_2} , \hat{n}_u and \hat{n}_l , which are an average number of elements to be constrained on each row. The constraint operator \mathcal{W} is given by

$$\mathcal{W}_k: \begin{cases} \mathcal{E}^{i_k, i_k} = r_k, & \text{for } (i_k, j_k) \in \mathcal{B}^{e_1}, \quad k \in 1, \dots, m_{e_1}, \\ \mathcal{E}^{i_k, i_k} = r_k, & \text{for } (i_k, j_k) \in \mathcal{B}^{e_2}, \quad k \in m_{e_1} + 1, \dots, m_{e_1} + m_{e_2}, \\ \mathcal{E}^{i_k, i_k} \leq r_k, & \text{for } (i_k, j_k) \in \mathcal{B}^{e_2}, \quad k \in m_e + 1, \dots, m_e + m_{l_1}, \\ \mathcal{E}^{i_k, i_k} \geq r_k, & \text{for } (i_k, j_k) \in \mathcal{B}^{e_2}, \quad k \in m_e + m_{l_1}, \dots, m_{e_1} + m_{e_2} + m_{l_1} + m_{l_2}, \end{cases}$$

and each element in $r = [r_1; r_2; \dots; r_m]$ is randomly generated in the range of $[0, 1]$. In this example, we let $\hat{n}_{e_2} = \hat{n}_u = \hat{n}_l = \hat{n}$ and $n_1 = 1000$. Comparisons for three cases are reported in Table 4.3 where 1) $\hat{n} = 1$ and $m_{e_2} = m_{l_1} = m_{l_2} = 999$; 2) $\hat{n} = 5$ and $m_{e_2} = m_{l_1} = m_{l_2} = 4985$; 3) $\hat{n} = 10$ and $m_{e_2} = m_{l_1} = m_{l_2} = 9945$.

The above three examples compare the computational performance of two different implementations. We can see that the direct access by index referencing to the nonzero components of the constrained matrices, which was used in *CaliMat.m* saves computational time by a scalar factor (< 3 for the three examples here), while the local convergence rate for *Smh-NewtonBICG.m* retains the same as *CaliMat.m*. Now we look at some examples for solving problem (1.1) with *Smh-NewtonBICG.m*.

Example 4.2.4 is a generalized subproblem of solving rank minimization problems [11] with only equality constraints for both square and nonsquare matrices. In Example 4.2.5, the other three types of constraints in problem (1.1) are added in to demonstrate the computational flexibility of the *Smh-NewtonBICG.m*.

Example 4.2.4. Let (n_1, n_2) be the dimensions of matrices in (NS) , r be a predetermined rank, and m be the number of sample entries. $\rho = 1.0$, $\lambda = 1.0$ and $\mu = 0.0$. We generated $M = M_L M_R^T$, where M_L and M_R are $n \times r$ matrices with i.i.d. standard Gaussian entries. M is used as the matrix with some predetermined rank. we let $\rho = 1.0$, $\lambda = 1.0$, $\mu = 0.0$ and $C = \text{zeros}(n_1, n_2)$. The index sets for constraints are given by

$$\begin{aligned}\mathcal{B}^e &= \{(i_k, j_k) \mid k = 1, \dots, m_e\}, \\ \mathcal{B}^l &= \mathcal{B}^q = \mathcal{B}^u = \emptyset,\end{aligned}$$

and the constraint operator \mathcal{W} is given by

$$\mathcal{W}_k: \mathcal{E}^{i_k, j_k} = M(i_k, j_k) \text{ for } (i_k, j_k) \in \mathcal{B}^e, \quad k \in 1, \dots, m_e.$$

In the table 4.4, the computational results (average of five cases) are reported for cases with respect to the ratio (m/d_r) between the number of sampled entries (m) and the degree of freedom ($d_r: = r(n_1 + n_2 - r)$) of a $n_1 \times n_2$ matrix of rank r . So here $m_e = m$. The computational results for square matrix problem and nonsquare matrix problem are also compared. For square matrix problems, let $n_1 = n_2 = 1000$ and 1) $m/d_r = 4, m_e = 390000$; (2) $m/d_r = 5, m_e = 487500$. For nonsquare matrix problems, let $n_1 = 1000, n_2 = 1003$ and (3) $m/d_r = 4, m_e = 487500$; (4) $m/d_r = 5, m_e = 488250$.

As seen from the table 4.4, solving the nonsquare matrix problems is comparably more difficult in achieving the same level of residue as square matrix problems at

a similar number of iterations. Slow convergence shows when the residue of merit function reaches to $1.0E - 3$, thus for nonsquare cases the statistics are obtained when the residue falls below $1.0E - 3$. In the next example, we add in some inequality constraints, as well as the second order constraints. Comparisons are also shown between the uses of Huber smoothing function and Smale smoothing function introduced. Base on the results, we can see that for nonsymmetric matrix problems, the Smale function seems to be more superior than the Huber function.

Example 4.2.5. Let M be generated as in Example 4.2.4. Let $\rho = 1.0$, $\lambda = 1.0$ and $\mu = 1.0$. The index sets for constraints are randomly generated as in the previous examples. The constraint operator \mathcal{W} is given by

$$\mathcal{W}_k: \begin{cases} \mathcal{E}^{i_k, j_k} - M(i_k, j_k) = 0, \\ \mathcal{E}^{i_k, j_k} - M(i_k, j_k) \leq b_{l_1}, \\ \mathcal{E}^{i_k, j_k} - M(i_k, j_k) \geq b_{l_2}, \\ \mathcal{E}^{i_k, j_k} - M(i_k, j_k) \in \mathcal{K}_{k+1}, \\ \mathcal{E}^{i_k, j_k} - M(i_k, j_k) = b_u, \end{cases}$$

In table 4.5, we have the results for two cases of different dimensions: (1) $n_1 = n_2 = 1000$, $m/d_r = 5$, $m_e = 487500$, $m_{l_1} = m_{l_2} = 100$, $m_u = 100$, and $m_q = 10$; (2) $n_1 = n_2 = 2000$, $m/d_r = 5$, $m_e = 987500$, $m_{l_1} = m_{l_2} = 500$, $m_u = 500$, and $m_q = 50$.

	<i>CaliMat.m</i>	<i>SmhNewton.m</i>
Iterations	10	8
Func. Evaluation	11	8
BiCG/CG steps	27	14
Residule	4.80e-09	6.78e-07
Time (Precond.)	0.5	0.5
Time (BiCG/CG)	0.6	2.5
Time (SVD/EIG)	2.4	2.1
Total time (seconds)	4.2	7.7

Table 4.1: Example 4.2.1

	<i>CaliMat.m</i>			<i>SmhNewton.m</i>		
	n=500	n=1000	n=2000	n=500	n=1000	n=2000
Iterations	9	9	10	9	9	9
Func. Evaluation	10	10	11	9	9	9
BiCG/CG steps	23	23	26	15	15	15
Residule	3.2E-08	4.5E-07	1.4E-08	1.7e-07	1.4E-08	2.8E-08
Time (Precond.)	0.9	4.6	23.6	1.2	7.7	55.2
Time (BiCG/CG)	1.3	5.9	32.6	7.3	48.7	365.7
Time (SVD/EIG)	5.5	54.2	551.2	5.7	59.6	533.0
Total time (seconds)	8.6	67.4	616.2	18.2	133.8	1062.5

Table 4.2: Example 4.2.2

	<i>CaliMat.m</i>			<i>SmhNewton.m</i>		
	$\hat{n} = 1$	$\hat{n} = 5$	$\hat{n} = 10$	$\hat{n} = 1$	$\hat{n} = 5$	$\hat{n} = 10$
Iterations	11	12	14	11	12	15
Func. Evaluation	13	15	20	12	14	22
BiCG/CG steps	19	28	38	19	28	40
Residule	1.5E-07	3.5E-07	1.3E-07	1.8e-07	3.5E-07	1.3E-07
Time (Precond.)	2.6	3.0	3.5	10.6	12.6	16.2
Time (BiCG/CG)	9.9	14.8	21.4	65.7	95.4	137.1
Time (SVD/EIG)	68.0	80.1	105.8	73.1	87.8	127.8
Total time (seconds)	84.1	102.1	135.9	189.9	240.0	339.72

Table 4.3: Example 4.2.3

r=50	$n_1 = n_2 = 1000$		$n_1 = 1000, n_2 = 1003$	
	$m/d_r = 4$	$m/d_r = 5$	$m/d_r = 4$	$m/d_r = 5$
Iterations	6	6	10	10
Func. Evaluation	6	6	13	14
BiCG/CG steps	7	7.6	17	17
Residule	6.06E-08	1.06E-07	7.24E-04	6.46E-4
Time (BiCG/CG)	30.2	32.3	85.7	95.6
Time (SVD/EIG)	119.6	108.9	234.0	236.6
Total time (seconds)	212.4	181.3	413.4	453.9

Table 4.4: Example 4.2.4

	$n_1 = n_2 = 1000$ $m_e = 487500, m_u = 100$ $m_{l_1} = m_{l_2} = 100, m_q = 10$ 1) Huber 2) Smale		$n_1 = n_2 = 2000$ $m_e = 987500, m_u = 500$ $m_{l_1} = m_{l_2} = 500, m_q = 50$ 1) Huber 2) Smale	
Iterations	15	7	10	7
Func. Evaluation	15	7	10	7
BiCG/CG steps	27	12	17	9
Residule	6.68E-07	6.28E-07	7.10E-07	3.03E-07
Time (BiCG/CG)	135.8	56.9	487.1	359.4
Time (SVD/EIG)	247.1	118.0	1404.1	1239.7
Total time	525.5	245.3	2526.6	2041.1

Table 4.5: Example 4.2.5

Conclusions

In this thesis, we applied a smoothing Newton-BiCGStab method to solve the least squares nonsymmetric matrix nuclear norm problem (1.1). When the inequality and second order cone constraints are present, the corresponding dual problem is no longer an unconstrained convex problem. Solving the constrained dual problem is equivalent to solving for zeros of some system of nonsmooth equations. Smoothing functions are applied to the system of nonsmooth equations. The differential properties such as the global Lipschitz continuity and the strong semismoothness of the smoothed-nonsmooth functions have been presented in Chapter 2. The smoothing Newton-BiCGStab method illustrated in Chapter 3 can be globalized for solving problem (1.1) and a quadratic local convergence rate can be achieved under certain assumptions. Numerical experiments in the last chapter has demonstrated that Algorithm 3.1 can be used to efficiently solve problems (1.1).

Bibliography

- [1] R. Bhatia *Matrix Analysis*, Springer-Verlag, New York, 1997.
- [2] J. -F. Cai, E. J. Candés, and Z. W. Shen, A Singular Value Thresholding Algorithm for Matrix Completion, 2008.
- [3] M. T. Chu, R. E. Funderlic, R. J. Plemmons, *Structured Low Rank Approximation*, Linear Algebra and Its Applications 366(2003) 157-172.
- [4] B.C. Eaves, *On the Basic Theorem for Complementarity*, Mathematical Programming 1 (1971), pp. 68-75.
- [5] A. Fischer, *Solution of Monotone Complementarity Problems with Locally Lipschitzian Functions*, Mathematical Programming 76 (1997), pp 513-532.
- [6] Y. Gao and D. F. Sun, *Calibrating Least Squares Covariance Matrix Problems with Equality and Inequality Constraints*, SIAM Journal on Matrix Analysis and Applications 31 (2009), pp. 1432-1457.
- [7] Y. Gao and D. F. Sun, *A Majorized Penalty Approach for Calibrating Rank Constrained Correlation Matrix Problems*, manuscript, 2010.

-
- [8] G. H. Golub and C. F. Van Loan *Matrix Computations*, Johns Hopkins University Press, 1996.
 - [9] J. B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms* , I, volume 305 of Grundlehren der Mathematischen Wissenschaften, Fundamental Principles of Mathematical Sciences. SpringerVerlag, Berlin, 1993.
 - [10] K. -F. Jiang, D. F. Sun, and K. C. Toh, *A Proximal Point Method for Matrix Least Squares Problem with Nuclear Norm Regularization*, manuscript.
 - [11] Y. -J. Liu, D. F. Sun, and K. C. Toh, *An Implementable Proximal Point Algorithmic Framework for Nuclear Norm Minimization*, July 2009.
 - [12] S. Q. Ma, D. Goldfarb and L.F. Chen, *Fixed Point and Bregman Iterative Methods for Matrix Rank Minimization*, 2008.
 - [13] R. Mifflin, *Semismooth and Semiconvex Functions in Constrained Optimization* SIAM Journal, Control Optimization 15 957-972.
 - [14] J. V. Outrata, and D. F. Sun, *On the Coderivative of the Projection Operator onto the Second-order Cone*, June 19, 2008.
 - [15] H. D. Qi and D. F. Sun, *A Quadratically Convergent Newton Method for Computing the Nearest Correlation Matrix*, SIAM Journal on Matrix Analysis and Applications 28 (2006) 360-385.
 - [16] L. Qi and D. F. Sun, *Nonsmooth and Smoothing Methods for NCP and VI*. C. Floudas, P. Pardalos, eds. *Encyclopedia of Optimization*, Kluwer Academic Publishers, Norwell, MA, 100C104. 2001.
 - [17] R. T. Rockfellar, *Conjugate Duality and Optimization*, SIAM, Philadelphia, 1974.

-
- [18] D. F. Sun and J. Sun, *Semismooth Matrix-valued Functions*, Mathematical of Operations Research, Vol 27, No. 1 February 2002, pp. 150-169.
 - [19] D. F. Sun and L. Qi, *Solving Variational Inequality Problems via Smoothing-nonsmooth Reformulations*, Journal of Computational and Applied Mathematics, 129 (2001) 37–62.
 - [20] H. A. Van Der Vorst, *BI-CGSTAB: A Fast and Smoothly Converging Variant of BI-CG for the Solution of Nonsymmetric Linear Systems*, SIAM Journal on Scientific and Statistical Computing 13 (1992), pp. 631-644.
 - [21] J. Y. Zhao *The Smoothing Function of the Nonsmooth Matrix Valued Function*, Master Thesis, National University of Singapore, 2004.