

# On the Alternating Direction Method of Multipliers for Nonnegative Inverse Eigenvalue Problems with Partial Eigendata

Zhi Zhao\*      Zhengjian Bai†      Guizhi Chen‡

September 16, 2012

## Abstract

We consider the nonnegative inverse eigenvalue problem with partial eigendata, which aims to find a nonnegative matrix such that it is nearest to a pre-estimated nonnegative matrix and satisfies the prescribed eigendata. In this paper, we propose several iterative schemes based on the alternating direction method of multipliers for solving the nonnegative inverse problem. We also extend our schemes to the symmetric case and the cases of prescribed lower bounds and of prescribed entries. Numerical tests (including a practical engineering application in vibrations) show the efficiency of the proposed iterative schemes.

**Keywords.** Nonnegative matrix, inverse problem, alternating direction method of multipliers, variational inequality.

**AMS subject classifications.** 49J52, 49M15, 65F18, 90C33

## 1 Introduction

An  $n$ -by- $n$  matrix  $C \geq 0$  ( $C > 0$ , respectively) is called nonnegative (positive, respectively) if all its entries are greater than or equal to zero (greater than zero, respectively). Nonnegative matrices arises in various applications including game theory, Markov chain, probabilistic algorithms, numerical analysis, discrete distributions, categorical data, group theory, matrix scaling, and economics, etc. For the applications and mathematical properties of nonnegative matrices, one may refer to [2, 3, 23, 27] and references therein.

The nonnegative inverse eigenvalue problem aims to reconstruct a nonnegative matrix from its spectrum or partial eigendata. The nonnegative inverse eigenvalue problem has got much attention since 1940s (see for instance [10, 13] and references therein) and many authors considered

---

\*School of Mathematical Sciences, Xiamen University, Xiamen 361005, People's Republic of China (zhaozhi231@163.com).

†School of Mathematical Sciences, Xiamen University, Xiamen 361005, People's Republic of China (zjbai@xmu.edu.cn). The research of this author was partially supported by the National Natural Science Foundation of China grant 11271308, the Natural Science Foundation of Fujian Province of China for Distinguished Young Scholars (No. 2010J06002), and NCET.

‡School of Mathematical Sciences, Xiamen University, Xiamen 361005, People's Republic of China (chengz@xmu.edu.cn).

its solvability based on the complete set of eigenvalues. Recently, a few numerical algorithms were developed for computational purpose, including the isospectral flow method [5, 7, 8, 11], the alternating projection method [25], and the nonsmooth Newton-type method [1]. In particular, the isospectral flow method in [5] was extended to the case of prescribed structures and the nonsmooth Newton-type method in [1] was extended to the cases of lower bounds and of prescribed entries.

In this paper, we consider the following nonnegative inverse eigenvalue problem (NIEP): Given a predetermined  $n$ -by- $n$  nonnegative matrix  $C_o$  and a self-conjugate set of partial measured eigendata  $\{(\lambda_j, \mathbf{x}_j)\}_{j=1}^p$  with  $\lambda_j \in \mathbb{C}$ ,  $\mathbf{x}_j \in \mathbb{C}^n$ , and  $p \ll n$ , find an  $n$ -by- $n$  nonnegative matrix  $C$  such that it is nearest to a pre-estimated  $n$ -by- $n$  nonnegative matrix  $C_o$  in the Frobenius norm and has  $\{(\lambda_j, \mathbf{x}_j)\}_{j=1}^p$  as its eigenpairs.

In many applications, the entries of a nonnegative matrix stand for the physical parameters such as mass, length, elasticity, inductance, capacitance, and etc (see for instance [14, 17]). In practice, a predetermined nonnegative matrix  $C_o$  can be obtained from the real structure. However, the predicted dynamical behaviors by  $C_o$ , i.e., the eigenvalues and eigenvectors of  $C_o$ , often disagree with the experimentally measured data [14]. The nonnegative inverse problem aims to reconstruct a nonnegative matrix  $C$  from the measured eigendata.

We note that the NIEP is to find a solution of the following minimization problem.

$$\begin{aligned} \min \quad & \frac{1}{2} \|C - C_o\|^2 \\ \text{subject to (s.t.)} \quad & CX = X\Lambda, \\ & C \geq 0, \end{aligned} \tag{1}$$

where  $\|\cdot\|$  denotes the Euclidean vector norm or the Frobenius matrix norm and  $(\Lambda, X) \in \mathbb{R}^{p \times p} \times \mathbb{R}^{n \times p}$  is the real matrix form of the prescribed self-conjugate eigendata  $\{(\lambda_j, \mathbf{x}_j)\}_{j=1}^p$  as in [1]. For simplicity, we refer to the minimization problem (1) as the NIEP.

It is obvious that the NIEP is a convex quadratic programming problem, which can be solved by typical quadratic programming solvers based on interior point method (see for instance [24, 29]). However, in practice, the problem size  $n$  is very large (say,  $n \geq 1000$ ). Moreover, the number of equation constraints is also very large even when  $p$  is very small (say, if  $p = 30$  but  $n \geq 1000$ , then  $np \geq 30000$  or if  $p = 200$  but  $n \geq 1000$ , then  $np \geq 200000$ ). In this case, Newton-like or interior point algorithms are not so efficient for solving the NIEP since, in each iteration, solving a large-scale Newton equation whose dimensionality is proportional to  $np$  is inevitable.

In this paper, we propose several iterative schemes based on the alternating direction method of multipliers (ADMM) for solving the NIEP. This is motivated by the ADMM-based methods for variational problems [15, 16, 30] and semidefinite programming [19]. In particular, we present the original ADMM [16, 18], an ADMM-based descent method [30], and an relaxed ADMM [4] for the NIEP. The main advantages of ADMM-based methods for the NIEP lie in that: Unlike Newton-like or interior point algorithms, the proposed ADMM-based methods reduce the problem complexity in the sense that no system of linear equations is necessary to solve; The included subproblems are easy to solve: A subproblem has a closed-form solution. The other subproblem is a standard equality-constrained quadratic programming but its solution is explicitly expressed based on the Moore-Penrose inverse of the matrix  $X$  of the measured

eigenvectors, which can be computed with lower computation cost under a certain assumption on the eigendata. We also extend these ADMM-based schemes to the symmetric case and the cases of prescribed lower bounds and of prescribed entries. Finally, we report some numerical tests (including a practical engineering application in vibrations) to illustrate the efficiency of our methods.

Throughout the paper, we use the following notations. The symbols  $A^T, A^H$ , and  $A^\dagger$  denote the transpose and the conjugate transpose, and the Moore-Penrose inverse of a matrix  $A$ , respectively.  $I$  is the identity matrix of appropriate dimension. Denote by  $\|\cdot\|_{\max}$  the entry of largest absolute value of a matrix. Let  $\mathbb{R}^{n \times n}$  and  $\mathbb{SR}^{n \times n}$  be the set of all real matrices of order  $n$  and the set of all real symmetric matrices of order  $n$ , respectively. Let  $\mathbb{R}_+^{n \times n}$  and  $\mathbb{SR}_+^{n \times n}$  stand for the nonnegative orthants of  $\mathbb{R}^{n \times n}$  and  $\mathbb{SR}^{n \times n}$ , respectively. Denote by  $\Pi_{\mathcal{D}}\{\cdot\}$  the metric projection onto  $\mathcal{D} \subseteq \mathbb{R}^{n \times n}$  (or  $\mathbb{SR}^{n \times n}$ ).

The remainder of the paper is organized as follows. In section 2 we present ADMM-based methods for solving the NIEP. In section 3 we discuss some extensions. In section 4 we report some numerical results to demonstrate the efficiency of the proposed methods.

## 2 Alternating direction method of multipliers

In this section, we present some ADMM-based Methods for solving the NIEP.

### 2.1 Reformulation

In this subsection, we give ADMM-oriented reformulation of the NIEP. Let  $\mathbb{R}^{n \times n}$  ( $\mathbb{SR}^{n \times n}$ , respectively) be equipped with the inner product  $\langle A_1, A_2 \rangle = \text{tr}(A_1^T A_2)$  for any  $A_1, A_2 \in \mathbb{R}^{n \times n}$  ( $\mathbb{SR}^{n \times n}$ , respectively) and its induced norm  $\|\cdot\|$ , where ‘‘tr’’ means the trace of a matrix. Then the NIEP (1) can be reformulated as the following problem:

$$\begin{aligned} \min \quad & \frac{1}{2}\|C - C_o\|^2 + \frac{1}{2}\|Y - C_o\|^2 \\ \text{s.t.} \quad & C - Y = 0, \\ & C \in \mathbb{R}_+^{n \times n}, \quad Y \in \mathcal{K}, \end{aligned} \tag{2}$$

where  $\mathcal{K} := \{Y \in \mathbb{R}^{n \times n} : YX = X\Lambda\}$ .

We note that Problem (2) is a convex minimization problem. Therefore,  $(C^*, Y^*)$  is a solution to Problem (2) if and only if there exists a point  $Z^* \in \mathbb{R}^{n \times n}$  such that the following variational inequalities hold [24]

$$\begin{cases} \langle C - C^*, C^* - C_o - Z^* \rangle \geq 0 \quad \forall C \in \mathbb{R}_+^{n \times n}, \\ \langle Y - Y^*, Y^* - C_o + Z^* \rangle \geq 0 \quad \forall Y \in \mathcal{K}, \\ \langle Z - Z^*, C^* - Y^* \rangle \geq 0 \quad \forall Z \in \mathbb{R}^{n \times n}. \end{cases} \tag{3}$$

In the following, we propose ADMM-based methods for solving Problem (2). The augmented Lagrangian function of Problem (2) is given by

$$AL_\beta(C, Y, Z) = \frac{1}{2}\|C - C_o\|^2 + \frac{1}{2}\|Y - C_o\|^2 - \langle C - Y, Z \rangle + \frac{\beta}{2}\|C - Y\|^2,$$

where  $\beta > 0$  is a penalty parameter and  $Z \in \mathbb{R}^{n \times n}$  is the Lagrangian multiplier. By using the classical alternating Lagrangian method (e.g., [21, 26]) to Problem (2), we have the following iterative scheme:

$$\begin{cases} (C^{k+1}, Y^{k+1}) = \arg \min_{C \in \mathbb{R}_+^{n \times n}, Y \in \mathcal{K}} AL_\beta(C, Y, Z^k), \\ Z^{k+1} = Z^k - \beta(C^{k+1} - Y^{k+1}), \end{cases} \quad (4)$$

where  $(C^k, Y^k, Z^k)$  is the current iterate. We observe that the objective function of Problem (2) can be separated into two individual convex functions without crossed variables. As in [19], by splitting the minimizing problem in (4) into two smaller subproblems, we obtain the following ADMM:

$$\begin{cases} C^{k+1} = \arg \min_{C \in \mathbb{R}_+^{n \times n}} AL_\beta(C, Y^k, Z^k), \\ Y^{k+1} = \arg \min_{Y \in \mathcal{K}} AL_\beta(C^{k+1}, Y, Z^k), \\ Z^{k+1} = Z^k - \beta(C^{k+1} - Y^{k+1}). \end{cases} \quad (5)$$

## 2.2 Subproblems

We note that the iterate  $(C^{k+1}, Y^{k+1}, Z^{k+1})$  generated by (5) satisfies the following variational inequalities and equation:

$$\begin{cases} \langle C - C^{k+1}, C^{k+1} - C_o - [Z^k - \beta(C^{k+1} - Y^k)] \rangle \geq 0 \quad \forall C \in \mathbb{R}_+^{n \times n}, \\ \langle Y - Y^{k+1}, Y^{k+1} - C_o + [(Z^k - \beta(C^{k+1} - Y^{k+1}))] \rangle \geq 0 \quad \forall Y \in \mathcal{K}, \\ Z^{k+1} = Z^k - \beta(C^{k+1} - Y^{k+1}). \end{cases} \quad (6)$$

Moreover, we can easily check that the solutions of the two variational inequalities in (6) have the following expressions.

$$\begin{cases} C^{k+1} = \Pi_{\mathbb{R}_+^{n \times n}} \left\{ \frac{1}{1+\beta} (C_o + Z^k + \beta Y^k) \right\}, \\ Y^{k+1} = \Pi_{\mathcal{K}} \left\{ \frac{1}{1+\beta} (C_o - Z^k + \beta C^{k+1}) \right\}. \end{cases} \quad (7)$$

It is easy to know that for any  $W \in \mathbb{R}^{n \times n}$ ,  $\Pi_{\mathbb{R}_+^{n \times n}} \{W\} = (\max(W_{ij}, 0))$ . We also see that for any  $W \in \mathbb{R}^{n \times n}$ ,  $\Pi_{\mathcal{K}} \{W\}$  is the unique solution of the following quadratic optimization problem.

$$\begin{aligned} \min \quad & \|Y - W\|^2 \\ \text{s.t.} \quad & YX = X\Lambda, \\ & Y \in \mathbb{R}^{n \times n}. \end{aligned} \quad (8)$$

One may solve Problem (8) by using existing approaches for quadratic programming (see for instance [22, 24]). Here, we give an explicit solution to Problem (8). To do so, we need the following preliminary lemma.

**Lemma 2.1** [28, Lemma 1.3] *Let  $A_1 \in \mathbb{C}^{l \times m}$ ,  $A_2 \in \mathbb{C}^{n \times q}$ ,  $A_3 \in \mathbb{C}^{l \times q}$ ,  $E^* \in \mathbb{C}^{m \times n}$  be given. Define*

$$\mathcal{L} := \{E \in \mathbb{C}^{m \times n} : A_1 E A_2 = A_3\}.$$

Then  $\mathcal{L} \neq \emptyset$  if and only if  $A_1, A_2, A_3$  satisfy

$$A_1 A_1^\dagger A_3 A_2^\dagger A_2 = A_3,$$

and in case of  $\mathcal{L} \neq \emptyset$ , any  $E \in \mathcal{L}$  can be expressed as:

$$E = A_1^\dagger A_3 A_2^\dagger + F - A_1^\dagger A_1 F A_2 A_2^\dagger,$$

where  $F \in \mathbb{C}^{m \times n}$  is arbitrary. Moreover, there is a unique matrix  $E^{(0)} \in \mathcal{L}$  given by

$$E^{(0)} = A_1^\dagger A_3 A_2^\dagger + E^* - A_1^\dagger A_1 E^* A_2 A_2^\dagger$$

such that for any unitarily invariant norm  $\|\cdot\|$ ,

$$\|E^{(0)} - E^*\| = \min_{E \in \mathcal{L}} \|E - E^*\|.$$

Based on Lemma 2.1, we can easily find the explicit solution of Problem (8).

**Proposition 2.2** *The set  $\mathcal{K}$  is nonempty if and only if  $X \Lambda X^\dagger X = X \Lambda$ . In this case, every element of  $\mathcal{K}$  has the following form*

$$Y = X \Lambda X^\dagger + F(I - X X^\dagger),$$

where  $F \in \mathbb{R}^{n \times n}$  is arbitrary. Moreover, the unique solution of Problem (8) is given by

$$\Pi_{\mathcal{K}}\{W\} = X \Lambda X^\dagger + W(I - X X^\dagger). \quad (9)$$

**Remark 2.3** *We can assume that the matrix  $X$  is full column rank since  $X$  is the real matrix form of the partial measured eigenvectors  $\{\mathbf{x}_j\}_{j=1}^p$ . In this case, we have  $X^\dagger = (X^T X)^{-1} X^T$  and  $X^\dagger X = I$ . Therefore, by Proposition 2.2, we know that the set  $\mathcal{K}$  is nonempty and the unique solution of Problem (8) is given by (9), where  $X^\dagger = (X^T X)^{-1} X^T$  can be computed efficiently since  $p \ll n$ .*

### 2.3 ADMM-based algorithms

In this subsection, we propose several ADMM-based algorithms for solving Problem (2). For simplicity, given the current iterates  $C^k, Y^k$ , and  $Z^k$ , let  $\tilde{C}^k, \tilde{Y}^k$ , and  $\tilde{Z}^k$  be the iterates generated by solving (5). Then the classical ADMM proposed in [15, 16] is given by

$$\begin{cases} C^{k+1} &= \tilde{C}^k, \\ Y^{k+1} &= \tilde{Y}^k, \\ Z^{k+1} &= \tilde{Z}^k. \end{cases} \quad (10)$$

Also, based on the ADM in [30], we propose the following modified ADMM (MADMM):

$$\begin{cases} C^{k+1} &= \tilde{C}^k, \\ Y^{k+1} &= Y^k - \delta \theta_k (Y^k - \tilde{Y}^k), \\ Z^{k+1} &= Z^k - \delta \theta_k (Z^k - \tilde{Z}^k), \end{cases} \quad (11)$$

where  $\delta \in (0, 2)$  and

$$\theta_k = \frac{1}{\eta_k} (\eta_k - \langle Y^k - \tilde{Y}^k, Z^k - \tilde{Z}^k \rangle) \quad \text{with} \quad \eta_k = \beta \|Y^k - \tilde{Y}^k\|^2 + \frac{1}{\beta} \|Z^k - \tilde{Z}^k\|^2. \quad (12)$$

Finally, motivated by the recent work of Cai, Gu, He, and Yuan [4] for separable convex programming, we present the following relaxed (accelerated) ADMM (RADMM):

$$\begin{cases} \tilde{C}^k = \arg \min_{C \in \mathbb{R}_+^{n \times n}} AL_\beta(C, Y^k, Z^k), \\ \tilde{Z}^k = Z^k - \beta(\tilde{C}^k - Y^k), \\ \tilde{Y}^k = \arg \min_{Y \in \mathcal{K}} AL_\beta(\tilde{C}^k, Y, \tilde{Z}^k), \\ C^{k+1} = \tilde{C}^k, \\ Y^{k+1} = Y^k - \gamma(Y^k - \tilde{Y}^k), \\ Z^{k+1} = Z^k - \gamma(Z^k - \tilde{Z}^k), \end{cases} \quad (13)$$

where  $\gamma \in (0, 2)$  is a relaxation parameter. Compared to the original ADMM (10), the additional computation of the RADMM (13) is just two simple linear combinations of matrices. Hence, the computational cost of this additional step is negligible but the additional step can accelerate the original ADMM numerically [4]. From the latter numerical tests, we can observe the effectiveness of the RADMM over the ADMM for solving the NIEP.

One may refer to [15, 20] for the convergence of the original ADMM (10), [19] for the convergence of the MADMM (11)–(12), and [4] for the convergence of the RADMM (13).

Before ending this section, we provide a possible stopping criterion for the ADMM (10), the MADMM (11)–(12) and the RADMM (13). By (6), we can easily check that the iterates  $\tilde{C}^k$ ,  $\tilde{Y}^k$ , and  $\tilde{Z}^k$  generated by (5) satisfy the following variational inequalities.

$$\begin{cases} \langle C - \tilde{C}^k, \tilde{C}^k - C_o - \tilde{Z}^k - \beta(Y^k - \tilde{Y}^k) \rangle \geq 0 \quad \forall C \in \mathbb{R}_+^{n \times n}, \\ \langle Y - \tilde{Y}^k, \tilde{Y}^k - C_o + \tilde{Z}^k \rangle \geq 0 \quad \forall Y \in \mathcal{K}, \\ \langle Z - \tilde{Z}^k, \tilde{C}^k - \tilde{Y}^k + \frac{1}{\beta}(\tilde{Z}^k - Z^k) \rangle \geq 0 \quad \forall Z \in \mathbb{R}^{n \times n}. \end{cases} \quad (14)$$

It follows from (3) and (14) that  $(\tilde{C}^k, \tilde{Y}^k, \tilde{Z}^k)$  is a solution of (3) if and only if  $Y^k - \tilde{Y}^k = 0$  and  $\tilde{Z}^k - Z^k = 0$ . Hence, the stopping rule for the proposed ADMM-based schemes can be set to be

$$\begin{cases} \|Y^k - Y^{k+1}\|_{\max} / \|Y^0 - Y^1\|_{\max} \leq \epsilon, \\ \|Z^k - Z^{k+1}\|_{\max} / \|Z^0 - Z^1\|_{\max} \leq \epsilon, \end{cases}$$

where  $\epsilon > 0$  is a prescribed tolerance.

### 3 Extensions

In this section, we extend the proposed ADMM-based methods to the symmetric case and the cases of lower bounds and of prescribed entries.

### 3.1 Symmetric nonnegative inverse eigenvalue problems with partial eigendata

The symmetric nonnegative inverse problem with partial eigendata can be stated as follows.

**SNIEP.** Given a predetermined nonnegative matrix  $C_o \in \mathbb{SR}_+^{n \times n}$  and a set of partial measured eigendata  $\{(\lambda_j, \mathbf{x}_j)\}_{j=1}^p$  with  $\lambda_j \in \mathbb{R}$ ,  $\mathbf{x}_j \in \mathbb{R}^n$ , and  $p \ll n$ , find a nonnegative matrix  $C \in \mathbb{SR}_+^{n \times n}$  such that it is nearest to  $C_o$  in the Frobenius norm and has  $\{(\lambda_j, \mathbf{x}_j)\}_{j=1}^p$  as its eigenpairs.

Let  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_p) \in \mathbb{R}^{p \times p}$  and  $X = [x_1, \dots, x_p] \in \mathbb{R}^{n \times p}$ . Then the SNIEP amounts to the solution of the following minimization problem.

$$\begin{aligned} \min \quad & \frac{1}{2} \|C - C_o\|^2 \\ \text{s.t.} \quad & CX = X\Lambda, \\ & C \in \mathbb{SR}_+^{n \times n}, \end{aligned} \tag{15}$$

which can be rewritten as the following form:

$$\begin{aligned} \min \quad & \frac{1}{2} \|C - C_o\|^2 + \frac{1}{2} \|Y - C_o\|^2 \\ \text{s.t.} \quad & C - Y = 0, \\ & C \in \mathbb{SR}_+^{n \times n}, \quad Y \in \mathcal{K}_s, \end{aligned} \tag{16}$$

where  $\mathcal{K}_s := \{Y \in \mathbb{SR}^{n \times n} : YX = X\Lambda\}$ .

We now focus on Problem (16). The augmented Lagrangian function of Problem (16) is given by

$$AL_\beta^s(C, Y, Z) = \frac{1}{2} \|C - C_o\|^2 + \frac{1}{2} \|Y - C_o\|^2 - \langle C - Y, Z \rangle + \frac{\beta}{2} \|C - Y\|^2,$$

where  $\beta > 0$  is a penalty parameter and  $Z \in \mathbb{SR}^{n \times n}$  is the Lagrangian multiplier. Hence, we get the following ADMM:

$$\begin{cases} C^{k+1} = \arg \min_{C \in \mathbb{SR}_+^{n \times n}} AL_\beta^s(C, Y^k, Z^k), \\ Y^{k+1} = \arg \min_{Y \in \mathcal{K}_s} AL_\beta^s(C^{k+1}, Y, Z^k), \\ Z^{k+1} = Z^k - \beta(C^{k+1} - Y^{k+1}), \end{cases} \tag{17}$$

where  $(C^k, Y^k, Z^k)$  is the current iterate. It is easy to verify that the solutions of the two minimization problems in (17) have the following formulas.

$$\begin{cases} C^{k+1} = \Pi_{\mathbb{SR}_+^{n \times n}} \left\{ \frac{1}{1+\beta} (C_o + Z^k + \beta Y^k) \right\}, \\ Y^{k+1} = \Pi_{\mathcal{K}_s} \left\{ \frac{1}{1+\beta} (C_o - Z^k + \beta C^{k+1}) \right\}. \end{cases}$$

Here, for any  $W \in \mathbb{SR}^{n \times n}$ ,  $\Pi_{\mathbb{SR}_+^{n \times n}}\{W\} = (\max(W_{ij}, 0))$ . For any  $W \in \mathbb{SR}^{n \times n}$ ,  $\Pi_{\mathcal{K}_s}\{W\}$  is the unique solution to the following quadratic optimization problem:

$$\begin{aligned} \min \quad & \|Y - W\|^2 \\ \text{s.t.} \quad & YX = X\Lambda, \\ & Y \in \mathbb{SR}^{n \times n}. \end{aligned} \tag{18}$$

This is a classical quadratic programming, which can be solved by existing approaches for quadratic programming (see for instance [22, 24]). Now, we discuss how to solve Problem (18) directly. We first need the following preliminary lemma.

**Lemma 3.1** [28, Lemma 1.4] *Let  $A_1 \in \mathbb{C}^{n \times q}$ ,  $A_2 \in \mathbb{C}^{n \times q}$ ,  $(E^*)^H = E^* \in \mathbb{C}^{n \times n}$  be given. Define*

$$\mathcal{L}_s := \{E \in \mathbb{C}^{n \times n} : E^H = E, EA_1 = A_2\}.$$

*Then  $\mathcal{L}_s \neq \emptyset$  if and only if  $A_1, A_2$  satisfy*

$$A_2 A_1^\dagger A_1 = A_2 \quad \text{and} \quad (A_1 A_1^\dagger A_2 A_1^\dagger)^H = A_1 A_1^\dagger A_2 A_1^\dagger$$

*and in case of  $\mathcal{L}_s \neq \emptyset$ , any  $E \in \mathcal{L}_s$  can be expressed as:*

$$E = A_2 A_1^\dagger + (A_1^\dagger)^H A_2^H - (A_1^\dagger)^H A_2^H A_1 A_1^\dagger + (I - A_1 A_1^\dagger) F (I - A_1 A_1^\dagger),$$

*where  $F^H = F \in \mathbb{C}^{n \times n}$  is arbitrary. Moreover, there is a unique matrix  $E^{(0)} \in \mathcal{L}_s$  given by*

$$E^{(0)} = A_2 A_1^\dagger + (A_1^\dagger)^H A_2^H - (A_1^\dagger)^H A_2^H A_1 A_1^\dagger + (I - A_1 A_1^\dagger) E^* (I - A_1 A_1^\dagger),$$

*such that*

$$\|E^{(0)} - E^*\| = \min_{E \in \mathcal{L}_s} \|E - E^*\|.$$

Based on Lemma 3.1, we can find the explicit solution of Problem (18).

**Proposition 3.2** *The set  $\mathcal{K}_s$  is nonempty if and only if  $X \Lambda X^\dagger X = X \Lambda$  and  $(X X^\dagger X \Lambda X^\dagger)^T = X X^\dagger X \Lambda X^\dagger$ . In this case, every element of  $\mathcal{K}_s$  has the following form*

$$Y = X \Lambda X^\dagger + (X^\dagger)^T (X \Lambda)^T - (X^\dagger)^T (X \Lambda)^T X X^\dagger + (I - X X^\dagger) F (I - X X^\dagger),$$

*where  $F \in \mathbb{S}\mathbb{R}^{n \times n}$  is arbitrary. Moreover, the unique solution of Problem (18) is given by*

$$\Pi_{\mathcal{K}_s}\{W\} = X \Lambda X^\dagger + (X^\dagger)^T (X \Lambda)^T - (X^\dagger)^T (X \Lambda)^T X X^\dagger + (I - X X^\dagger) W (I - X X^\dagger). \quad (19)$$

**Remark 3.3** *As a matrix form of the measured eigenvectors  $\{\mathbf{x}_j\}_{j=1}^p$ ,  $X$  can be assumed to be full column rank. Then we have  $X^\dagger = (X^T X)^{-1} X^T$  and  $X^\dagger X = I$ . From Proposition 3.2, it follows that the set  $\mathcal{K}$  is nonempty if and only if  $(X \Lambda X^\dagger)^T = X \Lambda X^\dagger$ , which should hold in practice since  $(X, \Lambda)$  are the measured eigendata of a symmetric nonnegative matrix. In this case, the unique solution of Problem (18) is given by (19), where  $X^\dagger = (X^T X)^{-1} X^T$  can be calculated with lower computation cost.*

Let  $\tilde{C}^k$ ,  $\tilde{Y}^k$ , and  $\tilde{Z}^k$  denote the iterates generated by solving (17). As in Section 2, for Problem (16), we propose the ADMM:

$$\begin{cases} C^{k+1} &= \tilde{C}^k, \\ Y^{k+1} &= \tilde{Y}^k, \\ Z^{k+1} &= \tilde{Z}^k, \end{cases} \quad (20)$$



the MADMM:

$$\begin{cases} C^{k+1} = \tilde{C}^k, \\ Y^{k+1} = Y^k - \delta\theta_k(Y^k - \tilde{Y}^k), \\ Z^{k+1} = Z^k - \delta\theta_k(Z^k - \tilde{Z}^k), \end{cases} \quad (21)$$

where  $\delta \in (0, 2)$  and

$$\theta_k = \frac{1}{\eta_k}(\eta_k - \langle Y^k - \tilde{Y}^k, Z^k - \tilde{Z}^k \rangle) \quad \text{with} \quad \eta_k = \beta\|Y^k - \tilde{Y}^k\|^2 + \frac{1}{\beta}\|Z^k - \tilde{Z}^k\|^2, \quad (22)$$

and the RADMM:

$$\begin{cases} \tilde{C}^k = \arg \min_{C \in \mathbb{S}\mathbb{R}_+^{n \times n}} AL_\beta^s(C, Y^k, Z^k), \\ \tilde{Z}^k = Z^k - \beta(\tilde{C}^k - Y^k), \\ \tilde{Y}^k = \arg \min_{Y \in \mathcal{K}_s} AL_\beta^s(\tilde{C}^k, Y, \tilde{Z}^k), \\ C^{k+1} = \tilde{C}^k, \\ Y^{k+1} = Y^k - \gamma(Y^k - \tilde{Y}^k), \\ Z^{k+1} = Z^k - \gamma(Z^k - \tilde{Z}^k), \end{cases} \quad (23)$$

where  $\gamma \in (0, 2)$  is a relaxation parameter.

The convergence of the above ADMM-based schemes can be established similarly as in [4, 15, 19, 20].

## 3.2 The case of lower bounds

### 3.2.1 The nonsymmetric case

In many applications, each entry of  $C$  is required to be equal to or greater than a nonnegative number. For given  $C_o \in \mathbb{R}_+^{n \times n}$ , the NIEP with lower bounds is to find a solution to the following minimization problem.

$$\begin{aligned} \min \quad & \frac{1}{2}\|C - C_o\|^2 \\ \text{s.t.} \quad & CX = X\Lambda, \\ & C \geq \underline{C}, \end{aligned} \quad (24)$$

where  $\underline{C} \in \mathbb{R}_+^{n \times n}$  is a prescribed matrix and  $C \geq \underline{C}$  means that  $C - \underline{C} \geq 0$ . By renaming  $C := C - \underline{C}$  and  $C_o := C_o - \underline{C}$ , Problem (24) becomes

$$\begin{aligned} \min \quad & \frac{1}{2}\|C - C_o\|^2 \\ \text{s.t.} \quad & CX = X\Lambda - \underline{C}X, \\ & C \geq 0, \end{aligned} \quad (25)$$

An equivalent form of Problem (25) is given by

$$\begin{aligned} \min \quad & \frac{1}{2}\|C - C_o\|^2 + \frac{1}{2}\|Y - C_o\|^2 \\ \text{s.t.} \quad & C - Y = 0, \\ & C \in \mathbb{R}_+^{n \times n}, \quad Y \in \mathcal{K}_{\underline{C}}, \end{aligned} \tag{26}$$

where  $\mathcal{K}_{\underline{C}} := \{Y \in \mathbb{R}^{n \times n} : YX = X\Lambda - \underline{C}X\}$ .

We note that if  $(\overline{C}, \overline{Y})$  is a solution to Problem (26), then the solution of Problem (24) is given by

$$C^* = \overline{C} + \underline{C}.$$

In the following, we provide some ADMM-based schemes for solving Problem (26). The augmented Lagrangian function of Problem (26) is given by

$$AL_{\beta}^{lb}(C, Y, Z) = \frac{1}{2}\|C - C_o\|^2 + \frac{1}{2}\|Y - C_o\|^2 - \langle C - Y, Z \rangle + \frac{\beta}{2}\|C - Y\|^2,$$

where  $\beta > 0$  is a penalty parameter and  $Z \in \mathbb{R}^{n \times n}$  is the Lagrangian multiplier. Then we get the following ADMM:

$$\begin{cases} C^{k+1} = \arg \min_{C \in \mathbb{R}_+^{n \times n}} AL_{\beta}^{lb}(C, Y^k, Z^k), \\ Y^{k+1} = \arg \min_{Y \in \mathcal{K}_{\underline{C}}} AL_{\beta}^{lb}(C^{k+1}, Y, Z^k), \\ Z^{k+1} = Z^k - \beta(C^{k+1} - Y^{k+1}), \end{cases} \tag{27}$$

where  $(C^k, Y^k, Z^k)$  is the current iterate. By simple calculation, the solutions of the two minimization problems in (27) have the following forms.

$$\begin{cases} C^{k+1} = \Pi_{\mathbb{R}_+^{n \times n}} \left\{ \frac{1}{1+\beta}(C_o + Z^k + \beta Y^k) \right\}, \\ Y^{k+1} = \Pi_{\mathcal{K}_{\underline{C}}} \left\{ \frac{1}{1+\beta}(C_o - Z^k + \beta C^{k+1}) \right\}, \end{cases}$$

where for any  $W \in \mathbb{R}^{n \times n}$ ,  $\Pi_{\mathcal{K}_{\underline{C}}}\{W\}$  can be expressed explicitly by using Lemma 2.1, which can be computed with lower computation cost under the assumption that  $X$  is full column rank (see Proposition 2.2 and Remark 2.3).

Denote by  $\tilde{C}^k$ ,  $\tilde{Y}^k$ , and  $\tilde{Z}^k$  the iterates generated by solving (27). As in Section 2, for Problem (26), we propose the following ADMM:

$$\begin{cases} C^{k+1} = \tilde{C}^k, \\ Y^{k+1} = \tilde{Y}^k, \\ Z^{k+1} = \tilde{Z}^k, \end{cases} \tag{28}$$

the MADM:

$$\begin{cases} C^{k+1} = \tilde{C}^k, \\ Y^{k+1} = Y^k - \delta \theta_k(Y^k - \tilde{Y}^k), \\ Z^{k+1} = Z^k - \delta \theta_k(Z^k - \tilde{Z}^k), \end{cases} \tag{29}$$

where  $\delta \in (0, 2)$  and

$$\theta_k = \frac{1}{\eta_k} (\eta_k - \langle Y^k - \tilde{Y}^k, Z^k - \tilde{Z}^k \rangle) \quad \text{with} \quad \eta_k = \beta \|Y^k - \tilde{Y}^k\|^2 + \frac{1}{\beta} \|Z^k - \tilde{Z}^k\|^2, \quad (30)$$

and the RADMM:

$$\left\{ \begin{array}{l} \tilde{C}^k = \arg \min_{C \in \mathbb{R}_+^{n \times n}} AL_\beta^{lb}(C, Y^k, Z^k), \\ \tilde{Z}^k = Z^k - \beta(\tilde{C}^k - Y^k), \\ \tilde{Y}^k = \arg \min_{Y \in \mathcal{K}_C} AL_\beta^{lb}(\tilde{C}^k, Y, \tilde{Z}^k), \\ C^{k+1} = \tilde{C}^k, \\ Y^{k+1} = Y^k - \gamma(Y^k - \tilde{Y}^k), \\ Z^{k+1} = Z^k - \gamma(Z^k - \tilde{Z}^k), \end{array} \right. \quad (31)$$

where  $\gamma \in (0, 2)$  is a relaxation parameter.

The convergence of the above ADMM-based schemes can be found in [4, 15, 19, 20].

### 3.2.2 The symmetric case

For given  $C_o \in \mathbb{S}\mathbb{R}_+^{n \times n}$ , the SNIEP with lower bounds aims to find a matrix  $C \in \mathbb{S}\mathbb{R}_+^{n \times n}$  such that it solves the following minimization problem.

$$\begin{array}{ll} \min & \frac{1}{2} \|C - C_o\|^2 \\ \text{s.t.} & CX = X\Lambda, \\ & C \geq \underline{C}, \end{array} \quad (32)$$

where  $\underline{C} \in \mathbb{S}\mathbb{R}_+^{n \times n}$  is a prescribed matrix. By renaming  $C := C - \underline{C}$  and  $C_o := C_o - \underline{C}$ , Problem (32) turns into

$$\begin{array}{ll} \min & \frac{1}{2} \|C - C_o\|^2 \\ \text{s.t.} & CX = X\Lambda - \underline{C}X, \\ & C \geq 0, \end{array}$$

which can be reformulated as the following quadratic programming:

$$\begin{array}{ll} \min & \frac{1}{2} \|C - C_o\|^2 + \frac{1}{2} \|Y - C_o\|^2 \\ \text{s.t.} & C - Y = 0, \\ & C \in \mathbb{S}\mathbb{R}_+^{n \times n}, \quad Y \in \mathcal{K}_{\underline{C}}^s, \end{array} \quad (33)$$

where  $\mathcal{K}_{\underline{C}}^s := \{Y \in \mathbb{S}\mathbb{R}^{n \times n} : YX = X\Lambda - \underline{C}X\}$ .

We see that if  $(\bar{C}, \bar{Y})$  is a solution to Problem (33), then the solution to Problem (32) is given by

$$C^* = \bar{C} + \underline{C}.$$

Next, we give several ADMM-based schemes for solving Problem (33). The augmented Lagrangian function of Problem (33) is given by

$$AL_{\beta}^{lbs}(C, Y, Z) = \frac{1}{2}\|C - C_o\|^2 + \frac{1}{2}\|Y - C_o\|^2 - \langle C - Y, Z \rangle + \frac{\beta}{2}\|C - Y\|^2,$$

where  $\beta > 0$  is a penalty parameter and  $Z \in \mathbb{SR}^{n \times n}$  is the Lagrangian multiplier. Then we obtain the following ADMM:

$$\begin{cases} C^{k+1} = \arg \min_{C \in \mathbb{SR}_+^{n \times n}} AL_{\beta}^{lbs}(C, Y^k, Z^k), \\ Y^{k+1} = \arg \min_{Y \in \mathcal{K}_{\underline{C}}^s} AL_{\beta}^{lbs}(C^{k+1}, Y, Z^k), \\ Z^{k+1} = Z^k - \beta(C^{k+1} - Y^{k+1}), \end{cases} \quad (34)$$

where  $(C^k, Y^k, Z^k)$  is the current iterate. It is easy to show that the solutions of the two minimization problems in (34) have the following closed forms.

$$\begin{cases} C^{k+1} = \Pi_{\mathbb{SR}_+^{n \times n}} \left\{ \frac{1}{1+\beta}(C_o + Z^k + \beta Y^k) \right\}, \\ Y^{k+1} = \Pi_{\mathcal{K}_{\underline{C}}^s} \left\{ \frac{1}{1+\beta}(C_o - Z^k + \beta C^{k+1}) \right\}, \end{cases}$$

where for any  $W \in \mathbb{SR}^{n \times n}$ ,  $\Pi_{\mathcal{K}_{\underline{C}}^s}\{W\}$  can be calculated with the lower computation cost using Lemma 3.1 under the assumption that  $X$  is full column rank (see Proposition 3.2 and Remark 3.3).

Let  $\tilde{C}^k$ ,  $\tilde{Y}^k$ , and  $\tilde{Z}^k$  stand for the iterates generated by solving (34). As in Section 2, for Problem (33), we propose the ADMM:

$$\begin{cases} C^{k+1} = \tilde{C}^k, \\ Y^{k+1} = \tilde{Y}^k, \\ Z^{k+1} = \tilde{Z}^k, \end{cases} \quad (35)$$

the MADMM:

$$\begin{cases} C^{k+1} = \tilde{C}^k, \\ Y^{k+1} = Y^k - \delta \theta_k (Y^k - \tilde{Y}^k), \\ Z^{k+1} = Z^k - \delta \theta_k (Z^k - \tilde{Z}^k), \end{cases} \quad (36)$$

where  $\delta \in (0, 2)$  and

$$\theta_k = \frac{1}{\eta_k} (\eta_k - \langle Y^k - \tilde{Y}^k, Z^k - \tilde{Z}^k \rangle) \quad \text{with} \quad \eta_k = \beta \|Y^k - \tilde{Y}^k\|^2 + \frac{1}{\beta} \|Z^k - \tilde{Z}^k\|^2, \quad (37)$$

and the RADMM:

$$\begin{cases} \tilde{C}^k = \arg \min_{C \in \mathbb{S}\mathbb{R}_+^{n \times n}} AL_\beta^{lbs}(C, Y^k, Z^k), \\ \tilde{Z}^k = Z^k - \beta(\tilde{C}^k - Y^k), \\ \tilde{Y}^k = \arg \min_{Y \in \mathcal{K}_C^s} AL_\beta^{lbs}(\tilde{C}^k, Y, \tilde{Z}^k), \\ C^{k+1} = \tilde{C}^k, \\ Y^{k+1} = Y^k - \gamma(Y^k - \tilde{Y}^k), \\ Z^{k+1} = Z^k - \gamma(Z^k - \tilde{Z}^k), \end{cases} \quad (38)$$

where  $\gamma \in (0, 2)$  is a relaxation parameter.

The convergence of the above ADMM-based schemes can be deduced as in [4, 15, 19, 20].

### 3.3 The case of prescribed entries

#### 3.3.1 The nonsymmetric case

In many applications, some entries of the nonnegative matrix  $C$  should be fixed (e.g., the symmetric tridiagonal oscillatory matrix in vibrations [17]). As in [1, 5], we consider the NIEP with prescribed entries: Given  $C_o \in \mathbb{R}_+^{n \times n}$ , find a matrix  $C \in \mathbb{R}_+^{n \times n}$  such that it solves the following minimization problem.

$$\begin{aligned} \min \quad & \frac{1}{2} \|C - C_o\|^2 \\ \text{s.t.} \quad & CX = X\Lambda, \\ & C \geq 0, \\ & C_{ij} = (C_o)_{ij} \quad \forall (i, j) \in \mathcal{I}, \end{aligned} \quad (39)$$

where  $\mathcal{I}$  is a prescribed index set such that  $\mathcal{I} \subseteq \mathcal{N} := \{(i, j) \mid i, j = 1, \dots, n\}$ . We know that Problem (39) can be changed into the following form:

$$\begin{aligned} \min \quad & \frac{1}{2} \|C - C_o\|^2 + \frac{1}{2} \|Y - C_o\|^2 \\ \text{s.t.} \quad & C - Y = 0, \\ & C \in \mathcal{C}, \quad Y \in \mathcal{K}, \end{aligned} \quad (40)$$

where  $\mathcal{C} := \{C \in \mathbb{R}_+^{n \times n} : C_{ij} = (C_o)_{ij} \forall (i, j) \in \mathcal{I}\}$  and  $\mathcal{K}$  is defined as in (2).

We now develop some ADMM-based schemes for solving Problem (40). The augmented Lagrangian function of Problem (40) is given by

$$AL_\beta^{pe}(C, Y, Z) = \frac{1}{2} \|C - C_o\|^2 + \frac{1}{2} \|Y - C_o\|^2 - \langle C - Y, Z \rangle + \frac{\beta}{2} \|C - Y\|^2,$$

where  $\beta > 0$  is a penalty parameter and  $Z \in \mathbb{R}^{n \times n}$  is the Lagrangian multiplier. Therefore, we

give the following ADMM for solving Problem (40).

$$\begin{cases} C^{k+1} = \arg \min_{C \in \mathcal{C}} AL_{\beta}^{pe}(C, Y^k, Z^k), \\ Y^{k+1} = \arg \min_{Y \in \mathcal{K}} AL_{\beta}^{pe}(C^{k+1}, Y, Z^k), \\ Z^{k+1} = Z^k - \beta(C^{k+1} - Y^{k+1}), \end{cases} \quad (41)$$

where  $(C^k, Y^k, Z^k)$  is the current iterate. We can deduce that the solutions of the two minimization problems in (41) are given by

$$\begin{cases} C^{k+1} = \Pi_{\mathcal{C}}\left\{\frac{1}{1+\beta}(C_o + Z^k + \beta Y^k)\right\}, \\ Y^{k+1} = \Pi_{\mathcal{K}}\left\{\frac{1}{1+\beta}(C_o - Z^k + \beta C^{k+1})\right\}, \end{cases}$$

where for any  $W \in \mathbb{R}^{n \times n}$ ,  $\Pi_{\mathcal{C}}\{W\}$  is given by

$$(\Pi_{\mathcal{C}}\{W\})_{ij} = \begin{cases} (C_o)_{ij}, & (i, j) \in \mathcal{I}, \\ \max(C_{ij}, 0), & \text{otherwise.} \end{cases}$$

By Proposition 2.2 and Remark 2.3, we know that, for any  $W \in \mathbb{R}^{n \times n}$ ,  $\Pi_{\mathcal{C}}\{W\}$  can be computed with lower computation cost under the assumption that  $X$  is full column rank.

Let  $\tilde{C}^k$ ,  $\tilde{Y}^k$ , and  $\tilde{Z}^k$  be the iterates generated by solving (41). As in Section 2, for Problem (40), we give the following ADMM:

$$\begin{cases} C^{k+1} = \tilde{C}^k, \\ Y^{k+1} = \tilde{Y}^k, \\ Z^{k+1} = \tilde{Z}^k, \end{cases} \quad (42)$$

the MADM:

$$\begin{cases} C^{k+1} = \tilde{C}^k, \\ Y^{k+1} = Y^k - \delta \theta_k (Y^k - \tilde{Y}^k), \\ Z^{k+1} = Z^k - \delta \theta_k (Z^k - \tilde{Z}^k), \end{cases} \quad (43)$$

where  $\delta \in (0, 2)$  and

$$\theta_k = \frac{1}{\eta_k} (\eta_k - \langle Y^k - \tilde{Y}^k, Z^k - \tilde{Z}^k \rangle) \quad \text{with} \quad \eta_k = \beta \|Y^k - \tilde{Y}^k\|^2 + \frac{1}{\beta} \|Z^k - \tilde{Z}^k\|^2, \quad (44)$$

and the RADMM:

$$\begin{cases} \tilde{C}^k = \arg \min_{C \in \mathcal{C}} AL_{\beta}^{pe}(C, Y^k, Z^k), \\ \tilde{Z}^k = Z^k - \beta(\tilde{C}^k - Y^k), \\ \tilde{Y}^k = \arg \min_{Y \in \mathcal{K}} AL_{\beta}^{pe}(\tilde{C}^k, Y, \tilde{Z}^k), \\ C^{k+1} = \tilde{C}^k, \\ Y^{k+1} = Y^k - \gamma(Y^k - \tilde{Y}^k), \\ Z^{k+1} = Z^k - \gamma(Z^k - \tilde{Z}^k), \end{cases} \quad (45)$$

where  $\gamma \in (0, 2)$  is a relaxation parameter.

For the convergence of the above ADMM-based schemes, we can refer to [4, 15, 19, 20].

### 3.3.2 The symmetric case

Let  $C_o \in \mathbb{S}\mathbb{R}_+^{n \times n}$ . The SNIEP with prescribed entries is to find a solution of the following minimization problem.

$$\begin{aligned} \min \quad & \frac{1}{2} \|C - C_o\|^2 \\ \text{s.t.} \quad & CX = X\Lambda, \\ & C \geq 0, \\ & C_{ij} = (C_o)_{ij} \quad \forall (i, j) \in \mathcal{I}_s, \end{aligned} \tag{46}$$

where  $\mathcal{I}_s$  is a prescribed index set such that  $\mathcal{I}_s \subseteq \mathcal{N} := \{(i, j) \mid i, j = 1, \dots, n\}$  and if  $(i, j) \in \mathcal{I}_s$  then  $(j, i) \in \mathcal{I}_s$ . We can rewrite Problem (46) as the following minimization problem.

$$\begin{aligned} \min \quad & \frac{1}{2} \|C - C_o\|^2 + \frac{1}{2} \|Y - C_o\|^2 \\ \text{s.t.} \quad & C - Y = 0, \\ & C \in \mathcal{C}_s, \quad Y \in \mathcal{K}_s, \end{aligned} \tag{47}$$

where  $\mathcal{C}_s := \{C \in \mathbb{S}\mathbb{R}_+^{n \times n} : C_{ij} = (C_o)_{ij} \forall (i, j) \in \mathcal{I}_s\}$  and  $\mathcal{K}_s$  is defined as in (16).

We now present several ADMM-based schemes for solving Problem (47). The augmented Lagrangian function of Problem (47) is given by

$$AL_\beta^{pes}(C, Y, Z) = \frac{1}{2} \|C - C_o\|^2 + \frac{1}{2} \|Y - C_o\|^2 - \langle C - Y, Z \rangle + \frac{\beta}{2} \|C - Y\|^2,$$

where  $\beta > 0$  is a penalty parameter and  $Z \in \mathbb{S}\mathbb{R}^{n \times n}$  is the Lagrangian multiplier. Then we have the following ADMM:

$$\begin{cases} C^{k+1} = \arg \min_{C \in \mathcal{C}_s} AL_\beta^{pes}(C, Y^k, Z^k), \\ Y^{k+1} = \arg \min_{Y \in \mathcal{K}_s} AL_\beta^{pes}(C^{k+1}, Y, Z^k), \\ Z^{k+1} = Z^k - \beta(C^{k+1} - Y^{k+1}), \end{cases} \tag{48}$$

where  $(C^k, Y^k, Z^k)$  is the current iterate. It is easy to check that the solutions of the two minimization problems in (48) have the closed forms:

$$\begin{cases} C^{k+1} &= \Pi_{\mathcal{C}} \left\{ \frac{1}{1+\beta} (C_o + Z^k + \beta Y^k) \right\}, \\ Y^{k+1} &= \Pi_{\mathcal{K}} \left\{ \frac{1}{1+\beta} (C_o - Z^k + \beta C^{k+1}) \right\}, \end{cases}$$

where for any  $W \in \mathbb{S}\mathbb{R}^{n \times n}$ ,  $\Pi_{\mathcal{C}}\{W\}$  is given by

$$(\Pi_{\mathcal{C}}\{W\})_{ij} = \begin{cases} (C_o)_{ij}, & (i, j) \in \mathcal{I}_s, \\ \max(C_{ij}, 0), & \text{otherwise.} \end{cases}$$

By Proposition 3.2 and Remark 3.3, for any  $W \in \mathbb{S}\mathbb{R}^{n \times n}$ ,  $\Pi_{\mathcal{K}}\{W\}$  can be computed efficiently under the assumption that  $X$  is full column rank.

Let  $\tilde{C}^k$ ,  $\tilde{Y}^k$ , and  $\tilde{Z}^k$  denote the iterates generated by solving (48). As in Section 2, for Problem (47), we provide the following ADMM:

$$\begin{cases} C^{k+1} &= \tilde{C}^k, \\ Y^{k+1} &= \tilde{Y}^k, \\ Z^{k+1} &= \tilde{Z}^k, \end{cases} \quad (49)$$

the MADM:

$$\begin{cases} C^{k+1} = \tilde{C}^k, \\ Y^{k+1} = Y^k - \delta\theta_k(Y^k - \tilde{Y}^k), \\ Z^{k+1} = Z^k - \delta\theta_k(Z^k - \tilde{Z}^k), \end{cases} \quad (50)$$

where  $\delta \in (0, 2)$  and

$$\theta_k = \frac{1}{\eta_k} (\eta_k - \langle Y^k - \tilde{Y}^k, Z^k - \tilde{Z}^k \rangle) \quad \text{with} \quad \eta_k = \beta \|Y^k - \tilde{Y}^k\|^2 + \frac{1}{\beta} \|Z^k - \tilde{Z}^k\|^2, \quad (51)$$

and the RADMM:

$$\begin{cases} \tilde{C}^k = \arg \min_{C \in \mathcal{C}_s} AL_{\beta}^{pes}(C, Y^k, Z^k), \\ \tilde{Z}^k = Z^k - \beta(\tilde{C}^k - Y^k), \\ \tilde{Y}^k = \arg \min_{Y \in \mathcal{K}_s} AL_{\beta}^{pes}(\tilde{C}^k, Y, \tilde{Z}^k), \\ C^{k+1} = \tilde{C}^k, \\ Y^{k+1} = Y^k - \gamma(Y^k - \tilde{Y}^k), \\ Z^{k+1} = Z^k - \gamma(Z^k - \tilde{Z}^k), \end{cases} \quad (52)$$

where  $\gamma \in (0, 2)$  is a relaxation parameter.

The convergence of the above ADMM-based schemes can be derived as in [4, 15, 19, 20].

## 4 Numerical tests

In this section, we report the numerical performances of the proposed ADMM-based iterative schemes for solving the NIEP. All the numerical tests were done using `MATLAB 7.10` on a personal computer Intel(R) Core(TM)2 Duo of 2.50 GHz CPU. In our numerical experiments, we choose the initial guess  $C^0 = Y^0 = C_o$  and  $Z^0 = 0$  for all the ADMM-based iterative schemes. For the NIEP, the stopping criterion is set to be

$$\text{Res.} := (\|C^k X - X \Lambda\|^2 + \sum_{i,j=1}^n \min(C_{ij}^k, 0)^2)^{\frac{1}{2}} \leq \epsilon;$$



For the SNIEP, the stopping criterion is set to be

$$\text{Res}^s := (\|C^k X - X \Lambda\|^2 + \sum_{i,j=1}^n \min(C_{ij}^k, 0)^2 + \|(C^k)^T - C^k\|^2)^{\frac{1}{2}} \leq \epsilon;$$

For the NIEP with prescribed entries, the stopping criterion is set to be

$$\text{Res}^{\text{pe}} := (\|C^k X - X \Lambda\|^2 + \sum_{i,j=1}^n \min(C_{ij}^k, 0)^2 + \sum_{(i,j) \in \mathcal{I}} (C_{ij}^k - (C_o)_{ij})^2)^{\frac{1}{2}} \leq \epsilon;$$

For the SNIEP with prescribed entries, the stopping criterion is set to be

$$\text{Res}^{\text{pes}} := (\|C^k X - X \Lambda\|^2 + \sum_{i,j=1}^n \min(C_{ij}^k, 0)^2 + \|(C^k)^T - C^k\|^2 + \sum_{(i,j) \in \mathcal{I}_s} (C_{ij}^k - (C_o)_{ij})^2)^{\frac{1}{2}} \leq \epsilon,$$

where  $\epsilon > 0$  is a given tolerance.

#### 4.1 The small-scale problem

In this section, we report some numerical tests for the small-scale NIEP. For demonstration purpose, we set  $\epsilon = 1.0 \times 10^{-7}$  for Examples 4.1–4.3 below.

**Example 4.1** *We consider the NIEP with  $n = 6$ . We first randomly generate an  $n$ -by- $n$  non-negative matrix  $\widehat{C}$  as follows:*

$$\widehat{C} = \begin{bmatrix} 0.5951 & 0.3668 & 0.6937 & 1.5196 & 1.0926 & 1.5873 \\ 0.9716 & 1.5850 & 0.8806 & 1.7252 & 0.8842 & 0.6675 \\ 2.0551 & 1.8138 & 0.4745 & 1.8580 & 1.2166 & 0.5957 \\ 1.8199 & 0.5857 & 1.8915 & 1.9115 & 1.3636 & 0.9424 \\ 0.7013 & 0.4898 & 1.6336 & 0.2773 & 0.8217 & 0.3996 \\ 0.6318 & 0.8815 & 1.2479 & 0.7110 & 0.7271 & 1.9140 \end{bmatrix}.$$

Then we set

$$C_o = \begin{bmatrix} 0.5416 & 0.3548 & 0.6943 & 1.4283 & 1.0900 & 1.4309 \\ 0.8936 & 1.4437 & 0.8466 & 1.7966 & 0.9005 & 0.6483 \\ 1.8683 & 1.6815 & 0.4960 & 1.7996 & 1.2431 & 0.5600 \\ 1.7134 & 0.5724 & 1.7736 & 1.8342 & 1.2701 & 0.9994 \\ 0.6620 & 0.5097 & 1.5563 & 0.2769 & 0.7646 & 0.3971 \\ 0.5788 & 0.8299 & 1.2428 & 0.7401 & 0.7257 & 1.9939 \end{bmatrix}.$$

We use the three eigenvalues  $\{6.6929, -0.7795 \pm 0.7092i\}$  of  $\widehat{C}$  and associated eigenvectors as prescribed eigendata, where  $i := \sqrt{-1}$ .

By applying the ADMM (10), the MADMM (11)–(12), and the RADMM (13) with  $\beta = 1$  and  $\delta = \gamma = 1.9$  to Example 4.1, we get the following physical solution:

$$\bar{C} = \begin{bmatrix} 0.6129 & 0.4134 & 0.7016 & 1.5021 & 1.1469 & 1.4930 \\ 0.9986 & 1.4850 & 0.8825 & 1.7987 & 0.8585 & 0.6669 \\ 2.0693 & 1.7675 & 0.4909 & 1.8401 & 1.2336 & 0.6270 \\ 1.7637 & 0.6320 & 1.8642 & 1.9018 & 1.2904 & 1.0450 \\ 0.6704 & 0.5320 & 1.6075 & 0.3049 & 0.7707 & 0.4122 \\ 0.6224 & 0.8362 & 1.2459 & 0.7206 & 0.6912 & 1.9893 \end{bmatrix}.$$

The numerical results are shown in Table 1, where **IT.** and **Err.** stand for the numbers of iterations and the residual  $\|C^k X - X\Lambda\|$  at the final iterate of our algorithms, respectively.

**Example 4.2** We focus on the SNIEP with  $n = 6$ . Let  $\hat{C}$  be a random  $n$ -by- $n$  symmetric nonnegative matrix given by

$$\hat{C} = \begin{bmatrix} 0.1425 & 1.3844 & 1.0963 & 1.8436 & 1.1848 & 0.8354 \\ 1.3844 & 0.5011 & 0.8572 & 1.7028 & 1.0592 & 1.0396 \\ 1.0963 & 0.8572 & 1.9323 & 0.3309 & 0.6526 & 0.8411 \\ 1.8436 & 1.7028 & 0.3309 & 2.0668 & 0.8491 & 0.8407 \\ 1.1848 & 1.0592 & 0.6526 & 0.8491 & 0.8609 & 1.7486 \\ 0.8354 & 1.0396 & 0.8411 & 0.8407 & 1.7486 & 1.0606 \end{bmatrix}.$$

Then we set

$$C_o = \begin{bmatrix} 0.1350 & 1.3621 & 1.1681 & 1.9196 & 1.1443 & 0.9010 \\ 1.3621 & 0.4683 & 0.8171 & 1.7786 & 1.0252 & 1.0567 \\ 1.1681 & 0.8171 & 1.8299 & 0.3531 & 0.6188 & 0.9060 \\ 1.9196 & 1.7786 & 0.3531 & 2.0432 & 0.8422 & 0.8798 \\ 1.1443 & 1.0252 & 0.6188 & 0.8422 & 0.8607 & 1.7236 \\ 0.9010 & 1.0567 & 0.9060 & 0.8798 & 1.7236 & 1.1000 \end{bmatrix}.$$

We use the three largest eigenvalues  $\{6.5825, 1.7633, 1.0568\}$  of  $\hat{C}$  and associated eigenvectors as prescribed eigendata.

By using the ADMM (20), the MADMM (21)–(22), and the RADMM (23) with  $\beta = 1$  and  $\delta = \gamma = 1.9$  to Example 4.2, we obtain the following physical solution:

$$\bar{C} = \begin{bmatrix} 0.0913 & 1.3481 & 1.1146 & 1.8885 & 1.1738 & 0.8662 \\ 1.3481 & 0.4711 & 0.8715 & 1.7356 & 1.0677 & 1.0473 \\ 1.1146 & 0.8715 & 1.9254 & 0.3144 & 0.6547 & 0.8315 \\ 1.8885 & 1.7356 & 0.3144 & 2.0277 & 0.8485 & 0.8230 \\ 1.1738 & 1.0677 & 0.6547 & 0.8485 & 0.8863 & 1.7245 \\ 0.8662 & 1.0473 & 0.8315 & 0.8230 & 1.7245 & 1.0748 \end{bmatrix}.$$

The numerical results are reported in Table 1.

**Example 4.3** We consider the NIEP with prescribed entries and  $n = 5$ . Let  $\widehat{C}$  be a random  $n \times n$  nonnegative matrix given by

$$\widehat{C} = \begin{bmatrix} 0.5951 & 0.6318 & 0.4898 & 1.8915 & 1.8580 \\ 0.9716 & 0.3668 & 0.8815 & 1.6336 & 1.9115 \\ 2.0551 & 1.5850 & 0.6937 & 1.2479 & 0.2773 \\ 1.8199 & 1.8138 & 0.8806 & 1.5196 & 0.7110 \\ 0.7013 & 0.5857 & 0.4745 & 1.7252 & 1.0926 \end{bmatrix}.$$

Then we set

$$C_o = \begin{bmatrix} 0.5416 & 0.5788 & 0.5097 & \mathbf{1.8915} & \mathbf{1.8580} \\ 0.8936 & 0.3548 & 0.8299 & 1.5563 & \mathbf{1.9115} \\ \mathbf{2.0551} & 1.4437 & 0.6943 & 1.2428 & 0.2769 \\ \mathbf{1.8199} & \mathbf{1.8138} & 0.8466 & 1.4283 & 0.7401 \\ 0.6620 & 0.5724 & 0.4960 & 1.7966 & 1.0900 \end{bmatrix},$$

where  $\mathcal{I} = \{(3, 1), (4, 1), (4, 2), (1, 4), (1, 5), (2, 5)\}$ . We use the three eigenvalues  $\{5.7450, -0.5347 \pm 0.4345i\}$  of  $\widehat{C}$  and associated eigenvectors as prescribed eigendata.

By applying the ADMM (42), the MADMM (43)–(44), and the RADMM (45) with  $\beta = 6$  and  $\delta = \gamma = 1.9$  to Example 4.3, we get the following physical solution:

$$\overline{C} = \begin{bmatrix} 0.5951 & 0.6318 & 0.4898 & \mathbf{1.8915} & \mathbf{1.8580} \\ 0.9725 & 0.3666 & 0.8837 & 1.6311 & \mathbf{1.9115} \\ \mathbf{2.0551} & 1.5720 & 0.6683 & 1.2766 & 0.2836 \\ \mathbf{1.8199} & \mathbf{1.8138} & 0.8806 & 1.5196 & 0.7110 \\ 0.6853 & 0.5920 & 0.4448 & 1.7599 & 1.0908 \end{bmatrix}.$$

The numerical results are displayed in Table 1.

**Example 4.4** We consider the SNIEP with prescribed entries and  $n = 5$ . Let  $\widehat{C}$  be a random  $n \times n$  symmetric nonnegative matrix given by

$$\widehat{C} = \begin{bmatrix} 0.1425 & 0.8007 & 0.5561 & 1.1179 & 0.2910 \\ 0.8007 & 1.8740 & 0.7745 & 0.7520 & 2.0560 \\ 0.5561 & 0.7745 & 1.6073 & 0.7319 & 1.5498 \\ 1.1179 & 0.7520 & 0.7319 & 1.9084 & 1.4194 \\ 0.2910 & 2.0560 & 1.5498 & 1.4194 & 1.9925 \end{bmatrix}.$$

Then we set

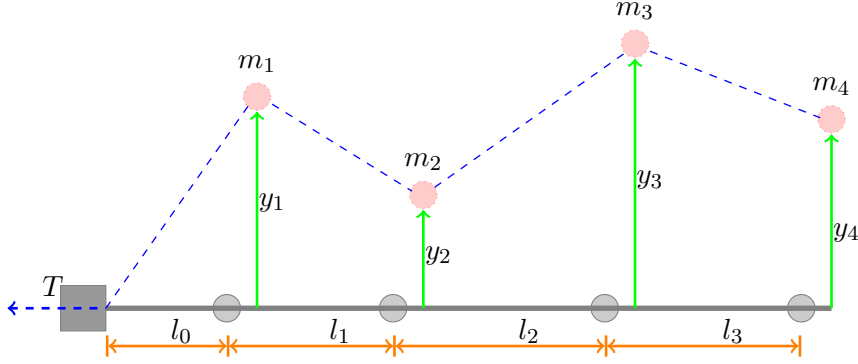
$$C_o = \begin{bmatrix} 0.1350 & 0.7971 & 0.5768 & 1.1708 & 0.2912 \\ 0.7971 & \mathbf{1.8740} & 0.7381 & 0.7411 & \mathbf{2.0560} \\ 0.5768 & 0.7381 & 1.6791 & 0.7260 & 1.4690 \\ 1.1708 & 0.7411 & 0.7260 & \mathbf{1.9084} & 1.5241 \\ 0.2912 & \mathbf{2.0560} & 1.4690 & 1.5241 & \mathbf{1.9925} \end{bmatrix},$$

where  $\mathcal{I}_s = \{(2, 2), (5, 2), (4, 4), (2, 5), (5, 5)\}$ . We use the two eigenvalues  $\{5.9410, 1.3768\}$  of  $\widehat{C}$  and associated eigenvectors as prescribed eigendata.





Figure 1: A fixed-free string with  $n = 4$  beads in the state of rest (solid gray) and in the deformation state (dotted).



$$(\hat{l}_0, \hat{l}_1, \hat{l}_2, \hat{l}_3) = (0.257175, 0.203200, 0.203200, 0.203200) \text{ (meter)}, \quad T = 154.2 \text{ (Newton)},$$

and let the matrix  $\hat{C}$  be in the form of (56) with  $m_j = \hat{m}_j$  and  $l_j = \hat{l}_j$ . Define the length  $\{l_j^o\}_{j=0}^{n-1}$  by

$$l_j^o := \hat{l}_j + r_j \hat{l}_j, \quad j = 0, 1, \dots, n-1,$$

where  $r_j$  is a random number in  $[-0.1, 0.1]$  and the matrix  $C_o$  takes the form of (56) with  $m_j = \hat{m}_j$  and  $l_j = l_j^o$ . Let  $\mathcal{I}_s$  denote the index set corresponding to the entries of  $\hat{C}$  whose values are equal to zeros. We choose the first  $p$  eigenvalues of  $\hat{C}$  with largest absolute values and associated eigenvectors as prescribed eigendata.

**Example 4.6** This is an inverse problem for the beaded string with  $n = 6$  beads. Let

$$(\hat{m}_1, \hat{m}_2, \hat{m}_3, \hat{m}_4, \hat{m}_5, \hat{m}_6) = (0.017804, 0.030783, 0.017804, 0.017804, 0.030783, 0.017804) \text{ (kg)},$$

$$(\hat{l}_0, \hat{l}_1, \hat{l}_2, \hat{l}_3, \hat{l}_4, \hat{l}_5) = (0.1047750, 0.203200, 0.127000, 0.254000, 0.127000, 0.203200) \text{ (meter)},$$

$$T = 161.4 \text{ (Newton)},$$

and let the matrix  $\hat{C}$  be in the form of (56) with  $m_j = \hat{m}_j$  and  $l_j = \hat{l}_j$ . Define the length  $\{l_j^o\}_{j=0}^{n-1}$  by

$$l_j^o := \hat{l}_j + r_j \hat{l}_j, \quad j = 0, 1, \dots, n-1,$$

where  $r_j$  is a random number in  $[-0.1, 0.1]$  and the matrix  $C_o$  has the form of (56) with  $m_j = \hat{m}_j$  and  $l_j = l_j^o$ . Let  $\mathcal{I}_s$  denote the index set corresponding to the entries of  $\hat{C}$  whose values are equal to zeros. We choose the first  $p$  eigenvalues of  $\hat{C}$  with largest absolute values and associated eigenvectors as prescribed eigendata.

We apply the proposed ADMM-based schemes to Examples 4.5–4.6, where  $\epsilon = 10^{-7}$ . The numerical results are listed in Tables 2–3, where Table 3 displays the computed masses and lengths for the beaded string. Table 2 shows that the proposed ADMM-based schemes works very well for Examples 4.5–4.6. We also observe from Table 3 that, as expected, the desired masses are recovered when  $p = 2$ .

Table 2: Convergence results for Examples 4.5–4.6

		Ex. 4.5		Ex. 4.6	
		$\beta = 6$ and $\delta = \gamma = 1.6$		$\beta = 4$ and $\delta = \gamma = 1.6$	
p		IT.	Err.	IT.	Err.
1	ADMM	51	$8.1 \times 10^{-8}$	95	$7.3 \times 10^{-8}$
	MADMM	51	$7.7 \times 10^{-8}$	57	$8.1 \times 10^{-8}$
	RADMM	49	$7.8 \times 10^{-8}$	53	$2.3 \times 10^{-8}$
		$\beta = 6$ and $\delta = \gamma = 1.6$		$\beta = 49$ and $\delta = \gamma = 1.6$	
2	ADMM	118	$9.5 \times 10^{-8}$	847	$7.8 \times 10^{-8}$
	MADMM	71	$3.3 \times 10^{-8}$	623	$9.9 \times 10^{-8}$
	RADMM	75	$8.0 \times 10^{-8}$	630	$9.8 \times 10^{-8}$

### 4.3 Comparison with an interior-point approach

To further illustrate the efficiency of the proposed ADMM-based schemes, we compare our methods numerically with the interior point algorithm in [29] for solving the NIEP. To use the package SDPT3 [29], we can reformulate the NIEP (1) as a standard form as follows:

$$\begin{aligned}
 \min \quad & \frac{1}{2}z \\
 \text{s.t.} \quad & CX = X\Lambda, \\
 & \sqrt{z} \geq \|C - C_o\|, \\
 & C \geq 0.
 \end{aligned} \tag{57}$$

Similarly, the SNIEP (15), the NIEP with lower bounds (24), the SNIEP with lower bounds (32), the NIEP with prescribed entries (39), and the SNIEP with prescribed entries (46) can be rewritten as the standard forms, which can be solved by using the package SDPT3.

**Example 4.7** We consider the NIEP with varying  $n$  and  $p$ . Let  $\widehat{C}$  be a random  $n \times n$  nonnegative matrix with each entry generated from the uniform distribution on the interval  $[0, 10]$ . Then we define  $C_o$  by

$$(C_o)_{ij} := \widehat{C}_{ij} + r_{ij}\widehat{C}_{ij} \quad i, j = 1, \dots, n.$$

where  $r_{ij}$  is a random number in  $[-0.1, 0.1]$ . We choose the first  $p$  eigenvalues of  $\widehat{C}$  with largest moduli and associated eigenvectors as prescribed eigendata.

**Example 4.8** We focus on the NIEP with prescribed entries for different  $n$  and  $p$ . Let  $\widehat{C}$  be a random  $n \times n$  nonnegative matrix with each entry generated from the uniform distribution on the interval  $[0, 10]$ . Define

$$(C_o)_{ij} := \begin{cases} \widehat{C}_{ij}, & (i, j) \in \mathcal{I}, \\ \widehat{C}_{ij} + r_{ij}\widehat{C}_{ij}, & \text{otherwise,} \end{cases}$$

where  $r_{ij}$  is a random number in  $[-0.1, 0.1]$  and  $\mathcal{I}$  is the index set corresponding to the entries of  $\widehat{C}$  whose values are equal to or greater than 8.5. We choose the first  $p$  eigenvalues of  $\widehat{C}$  with largest moduli and associated eigenvectors as prescribed eigendata.

Table 3: Recovered masses and lengths for Examples 4.5–4.6

Ex. 4.5							
p		$m_1$	$m_2$	$m_3$	$m_4$		
1	true	0.017804	0.009097	0.009097	0.017804		
	recovered	0.020316	0.008721	0.008204	0.016561		
	2	recovered	0.017804	0.009097	0.009097	0.017804	
p		$l_0$	$l_1$	$l_2$	$l_3$		
1	true	0.257175	0.203200	0.203200	0.203200		
	recovered	0.191507	0.196329	0.223961	0.221314		
	2	recovered	0.257175	0.203200	0.203200	0.203200	
Ex. 4.6							
p		$m_1$	$m_2$	$m_3$	$m_4$	$m_5$	$m_6$
1	true	0.017804	0.030783	0.017804	0.017804	0.030783	0.017804
	recovered	0.006835	0.016745	0.013255	0.022074	0.044911	0.028963
	2	recovered	0.017804	0.030783	0.017804	0.017804	0.030783
p		$l_0$	$l_1$	$l_2$	$l_3$	$l_4$	$l_5$
1	true	0.104775	0.203200	0.127000	0.254000	0.127000	0.203200
	recovered	0.292503	0.457304	0.201994	0.260547	0.093552	0.130123
	2	recovered	0.104775	0.203200	0.127000	0.254000	0.127000

**Example 4.9** We focus on the SNIEP with various  $n$  and  $p$ . Let  $\widehat{C}$  be a random  $n \times n$  symmetric nonnegative matrix with each entry generated from the uniform distribution on the interval  $[0, 10]$ . Define  $C_o$  by

$$(C_o)_{ij} := \widehat{C}_{ij} + r_{ij}\widehat{C}_{ij}, \quad i, j = 1, \dots, n,$$

where  $r_{ij} = r_{ji}$  is a random number in  $[-0.1, 0.1]$ . We choose the first  $p$  eigenvalues of  $\widehat{C}$  with largest absolute values and associated eigenvectors as prescribed eigendata.

**Example 4.10** We focus on the SNIEP with prescribed entries for different  $n$  and  $p$ . Let  $\widehat{C}$  be a random  $n \times n$  symmetric nonnegative matrix with each entry generated from the uniform distribution on the interval  $[0, 10]$ . Define

$$(C_o)_{ij} := \begin{cases} \widehat{C}_{ij}, & (i, j) \in \mathcal{I}, \\ \widehat{C}_{ij} + r_{ij}\widehat{C}_{ij}, & \text{otherwise,} \end{cases}$$

where  $r_{ij} = r_{ji}$  is a random number in  $[-0.1, 0.1]$  and  $\mathcal{I}_s$  is the index set corresponding to the entries of  $\widehat{C}$  whose values are equal to or greater than 8.5. We choose the first  $p$  eigenvalues of  $\widehat{C}$  with largest absolute values and associated eigenvectors as prescribed eigendata.

For simplicity, we set  $\epsilon = 10^{-7}$ . For the interior point algorithm in [29], we use the default tolerance. We report the numerical results for Examples 4.7–4.10 in Tables 4–7, where CT. denotes the total computing time (in seconds). Here, LS failed means that the step-length is too short to proceed during the computation.

We see from Tables 4–7 that the proposed ADMM-based schemes are more effective than the interior point algorithm in [29] for solving the NIEP.



Table 4: Comparison of ADMM and an interior-point approach for Example 4.7

		$\beta = 1$ and $\delta = \gamma = 1.9$											
		IPM			ADMM			MDMM			RDMM		
$n$	$p$	IT.	CT.	Err.	IT.	CT.	Err.	IT.	CT.	Err.	IT.	CT.	Err.
30	6	29	6.93	$1.7 \times 10^{-6}$	40	0.02	$7.8 \times 10^{-8}$	21	0.02	$7.9 \times 10^{-8}$	18	0.02	$5.1 \times 10^{-8}$
40	8	31	3.71	$2.1 \times 10^{-7}$	36	0.05	$8.0 \times 10^{-8}$	16	0.06	$6.7 \times 10^{-8}$	15	0.06	$8.2 \times 10^{-8}$
50	10	31	5.34	$2.2 \times 10^{-7}$	45	0.06	$7.3 \times 10^{-8}$	25	0.14	$6.9 \times 10^{-8}$	20	0.02	$7.2 \times 10^{-8}$
60	10	33	7.10	$3.5 \times 10^{-7}$	34	0.05	$9.4 \times 10^{-8}$	18	0.13	$3.2 \times 10^{-8}$	15	0.06	$4.0 \times 10^{-8}$
100	10	LS failed			28	0.06	$8.7 \times 10^{-8}$	12	0.09	$6.0 \times 10^{-8}$	11	0.06	$5.7 \times 10^{-8}$
200	10	LS failed			25	0.23	$8.5 \times 10^{-8}$	13	0.12	$7.7 \times 10^{-8}$	9	0.08	$6.6 \times 10^{-8}$
250	10	out of memory			24	0.28	$8.5 \times 10^{-8}$	14	0.19	$1.5 \times 10^{-8}$	9	0.11	$3.6 \times 10^{-8}$

Table 5: Comparison of ADMM and an interior-point approach for Example 4.8

		$\beta = 6$ and $\delta = \gamma = 1.6$											
		IPM			ADMM			MDMM			RDMM		
$n$	$p$	IT.	CT.	Err.	IT.	CT.	Err.	IT.	CT.	Err.	IT.	CT.	Err.
30	6	32	7.39	$3.5 \times 10^{-8}$	50	0.02	$7.1 \times 10^{-8}$	39	0.02	$8.9 \times 10^{-8}$	37	0.00	$9.3 \times 10^{-8}$
40	8	34	3.93	$1.0 \times 10^{-7}$	44	0.22	$6.1 \times 10^{-8}$	38	0.33	$9.8 \times 10^{-8}$	37	0.23	$9.7 \times 10^{-8}$
50	10	34	5.76	$1.3 \times 10^{-7}$	44	0.06	$5.3 \times 10^{-8}$	38	0.20	$8.2 \times 10^{-8}$	36	0.19	$7.3 \times 10^{-8}$
60	10	35	7.63	$9.6 \times 10^{-8}$	45	0.13	$2.6 \times 10^{-8}$	36	0.20	$8.7 \times 10^{-8}$	36	0.19	$8.1 \times 10^{-8}$
100	10	37	22.84	$2.1 \times 10^{-7}$	34	0.13	$4.0 \times 10^{-8}$	33	0.23	$7.3 \times 10^{-8}$	32	0.20	$5.5 \times 10^{-8}$
200	10	LS failed			28	0.27	$5.0 \times 10^{-8}$	31	0.33	$7.8 \times 10^{-8}$	28	0.28	$9.8 \times 10^{-8}$
250	10	out of memory			26	0.28	$8.3 \times 10^{-8}$	31	0.56	$7.9 \times 10^{-8}$	27	0.31	$9.8 \times 10^{-8}$

#### 4.4 The large-scale problem

In this section, we give some numerical experiments to show the efficiency of the proposed ADMM-based schemes for solving the large-scale NIEP. For demonstration purpose, in Tables 8–11, we report the numerical results for Examples 4.7–4.10 with different  $n$  and  $p$ .

Tables 8–11 show that the proposed algorithms are very efficient for solving the large-scale NIEP. We also observe that the proposed algorithms converge to the desired accuracy with only a small number of iterations. Finally, we see that the number of iterations is independent of the problem parameters  $n$  and  $p$  and thus the proposed algorithms are very robust.

## Acknowledgments

We would like to thank the editor and the anonymous referees for their helpful and valuable comments.

## References

- [1] Z. J. Bai, S. Serra-Capizzano, and Z. Zhao, *Nonnegative inverse eigenvalue problems with partial eigendata*, Numer. Math., 120 (2012), pp. 387–431.

Table 6: Comparison of ADMM and an interior-point approach for Example 4.9

		$\beta = 1$ and $\delta = \gamma = 1.9$											
		IPM			ADMM			MDMM			RDMM		
$n$	$p$	IT.	CT.	Err.	IT.	CT.	Err.	IT.	CT.	Err.	IT.	CT.	Err.
30	6	30	4.04	$1.8 \times 10^{-7}$	1	0.02	$4.6 \times 10^{-14}$	6	0.02	$4.6 \times 10^{-8}$	6	0.00	$4.6 \times 10^{-8}$
40	8	33	7.15	$4.6 \times 10^{-7}$	25	0.05	$9.2 \times 10^{-8}$	11	0.08	$8.5 \times 10^{-8}$	10	0.05	$7.5 \times 10^{-8}$
50	10	34	9.33	$3.8 \times 10^{-7}$	1	0.02	$1.2 \times 10^{-13}$	6	0.03	$7.8 \times 10^{-8}$	6	0.03	$7.8 \times 10^{-8}$
60	10	34	15.49	$4.2 \times 10^{-7}$	23	0.08	$9.0 \times 10^{-8}$	10	0.08	$3.9 \times 10^{-8}$	9	0.05	$4.8 \times 10^{-8}$
100	10	LS failed			1	0.02	$2.5 \times 10^{-13}$	7	0.05	$5.5 \times 10^{-9}$	7	0.05	$5.5 \times 10^{-9}$
200	10	out of memory			1	0.03	$6.5 \times 10^{-13}$	7	0.09	$8.0 \times 10^{-9}$	7	0.09	$8.0 \times 10^{-9}$

Table 7: Comparison of ADMM and an interior-point approach for Example 4.10

		$\beta = 5$ and $\delta = \gamma = 1.6$											
		IPM			ADMM			MDMM			RDMM		
$n$	$p$	IT.	CT.	Err.	IT.	CT.	Err.	IT.	CT.	Err.	IT.	CT.	Err.
30	6	32	4.18	$3.9 \times 10^{-7}$	49	0.02	$9.6 \times 10^{-8}$	34	0.02	$8.8 \times 10^{-8}$	34	0.02	$3.4 \times 10^{-8}$
40	8	33	7.15	$2.9 \times 10^{-7}$	55	0.19	$5.3 \times 10^{-8}$	36	0.20	$9.9 \times 10^{-8}$	37	0.20	$7.2 \times 10^{-8}$
50	10	34	9.41	$9.2 \times 10^{-7}$	38	0.17	$9.7 \times 10^{-8}$	33	0.19	$5.9 \times 10^{-8}$	31	0.20	$8.7 \times 10^{-8}$
60	10	34	15.37	$1.7 \times 10^{-6}$	41	0.19	$5.1 \times 10^{-8}$	33	0.19	$7.9 \times 10^{-8}$	32	0.25	$7.5 \times 10^{-8}$
100	10	38	131.3	$1.5 \times 10^{-7}$	33	0.20	$9.3 \times 10^{-8}$	30	0.23	$6.9 \times 10^{-8}$	27	0.19	$8.6 \times 10^{-8}$
200	10	out of memory			24	0.28	$9.9 \times 10^{-8}$	25	0.33	$6.9 \times 10^{-8}$	23	0.28	$9.6 \times 10^{-8}$

- [2] R. B. Bapat and T. E. S. Raghavan, *Nonnegative Matrices and Applications*, Encyclopedia of Mathematics and its Applications 64, Cambridge University Press, 1997.
- [3] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics in Applied Mathematics, 9, SIAM, Philadelphia, 1994.
- [4] X. J. Cai, G. Y. Gu, B. S. He, and X. M. Yuan, *A relaxed customized proximal point algorithm for separable convex programming*, Preprint, Department of Mathematics, Hong Kong Baptist University, Hong Kong, China, 2011.
- [5] X. Chen and D. L. Liu, *Isospectral flow method for nonnegative inverse eigenvalue problem with prescribed structure*, J. Comput. Appl. Math., 235 (2011), pp. 3990–4002.
- [6] M. T. Chu, *Inverse eigenvalue problems*, SIAM Rev., 40 (1998), pp. 1–39.
- [7] M. T. Chu, F. Diele, and I. Sgura, *Gradient flow method for matrix completion with prescribed eigenvalues*, Linear Algebra Appl., 379 (2004), pp. 85–112.
- [8] M. T. Chu and K. R. Driessel, *Constructing symmetric nonnegative matrices with prescribed eigenvalues by differential equations*, SIAM J. Math. Anal., 22 (1991), pp. 1372–1387.
- [9] M. T. Chu and G. H. Golub, *Structured inverse eigenvalue problems*, Acta Numer., 11 (2002), pp. 1–71.

Table 8: Numerical results for Example 4.7

$p = 30, \beta = 1.0, \text{ and } \delta = \gamma = 1.9$									
	ADMM			MADMM			RADMM		
n	IT.	CT.	Err.	IT.	CT.	Err.	IT.	CT.	Err.
100	44	0.34	$8.6 \times 10^{-8}$	25	0.23	$9.7 \times 10^{-8}$	20	0.17	$6.9 \times 10^{-8}$
200	31	0.34	$9.9 \times 10^{-8}$	13	0.20	$8.2 \times 10^{-8}$	13	0.14	$4.4 \times 10^{-8}$
500	27	0.72	$5.8 \times 10^{-8}$	14	0.78	$2.7 \times 10^{-8}$	10	0.33	$2.5 \times 10^{-8}$
1,000	25	2.26	$5.9 \times 10^{-8}$	15	2.96	$3.5 \times 10^{-8}$	8	0.89	$7.7 \times 10^{-8}$
1,500	24	5.01	$7.1 \times 10^{-8}$	15	6.82	$5.3 \times 10^{-9}$	8	2.08	$2.8 \times 10^{-8}$
2,000	24	8.67	$6.1 \times 10^{-8}$	15	11.72	$7.7 \times 10^{-8}$	8	3.64	$1.8 \times 10^{-8}$
$n = 1000, \beta = 1.0, \text{ and } \delta = \gamma = 1.9$									
	ADMM			MADMM			RADMM		
p	IT.	CT.	Err.	IT.	CT.	Err.	IT.	CT.	Err.
10	22	2.00	$7.4 \times 10^{-8}$	14	2.90	$1.9 \times 10^{-8}$	7	0.86	$6.9 \times 10^{-8}$
20	24	2.09	$6.1 \times 10^{-8}$	13	2.70	$7.0 \times 10^{-8}$	8	0.98	$3.2 \times 10^{-8}$
50	26	2.51	$9.2 \times 10^{-8}$	14	2.92	$7.3 \times 10^{-8}$	9	1.05	$6.0 \times 10^{-8}$
80	29	3.15	$5.9 \times 10^{-8}$	14	3.11	$8.4 \times 10^{-9}$	11	1.39	$1.8 \times 10^{-8}$
100	30	3.26	$7.7 \times 10^{-8}$	16	3.60	$7.8 \times 10^{-8}$	11	1.45	$6.9 \times 10^{-8}$
150	34	4.12	$6.1 \times 10^{-8}$	15	3.50	$9.1 \times 10^{-8}$	13	2.01	$7.1 \times 10^{-8}$
200	37	5.13	$9.9 \times 10^{-8}$	17	4.35	$7.3 \times 10^{-8}$	16	2.55	$3.3 \times 10^{-8}$

- [10] M. T. Chu and G. H. Golub, *Inverse Eigenvalue Problems: Theory, Algorithms, and Applications*, Oxford University Press, Oxford, 2005.
- [11] M. T. Chu and Q. Guo, *A numerical method for the inverse stochastic spectrum problem*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 1027–1039.
- [12] S. J. Cox, M. Embree, and J. M. Hokanson, *One can hear the composition of a string: experiments with an inverse eigenvalue problem*, SIAM Rev., 54 (2012), pp. 157–178.
- [13] P. D. Egleston, T. D. Lenker, and S. K. Narayan, *The nonnegative inverse eigenvalue problem*, Linear Algebra Appl., 379 (2004), pp. 475–490.
- [14] M. I. Friswell and J. E. Mottershead, *Finite Element Model Updating in Structural Dynamics*, Kluwer Academic Publishers, 1995.
- [15] D. Gabay, *Application of the method of multipliers to variational inequalities*, in *Augmented Lagrangian Methods: Application to the Numerical Solution of Boundary-value Problems*, M. Fortin and R. Glowinski, eds., North-Holland, Amsterdam, The Netherlands, 1983, pp. 299–331.
- [16] D. Gabay, and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximations*, Comput. Math. Appl., 2 (1976), pp. 17–40.
- [17] G. M. L. Gladwell, *Inverse Problems in Vibration*, Kluwer Academic Publishers, 2004.

Table 9: Numerical results for Example 4.8

$p = 30, \beta = 5, \text{ and } \delta = \gamma = 1.6$									
	ADMM			MADMM			RADMM		
n	IT.	CT.	Err.	IT.	CT.	Err.	IT.	CT.	Err.
100	52	0.27	$6.9 \times 10^{-8}$	39	0.30	$9.2 \times 10^{-8}$	38	0.25	$6.2 \times 10^{-8}$
200	37	0.31	$7.7 \times 10^{-8}$	33	0.33	$7.7 \times 10^{-8}$	31	0.28	$9.4 \times 10^{-8}$
500	29	0.95	$4.4 \times 10^{-8}$	30	1.81	$7.1 \times 10^{-8}$	27	0.91	$7.1 \times 10^{-8}$
1,000	26	2.95	$7.9 \times 10^{-8}$	31	6.94	$9.1 \times 10^{-8}$	25	3.22	$9.6 \times 10^{-8}$
1,500	25	6.58	$5.5 \times 10^{-8}$	32	15.86	$7.4 \times 10^{-8}$	25	7.43	$7.0 \times 10^{-8}$
2,000	27	12.45	$5.1 \times 10^{-8}$	32	28.33	$9.4 \times 10^{-8}$	25	13.24	$5.3 \times 10^{-8}$
$n = 1000, \beta = 5, \text{ and } \delta = \gamma = 1.6$									
	ADMM			MADMM			RADMM		
p	IT.	CT.	Err.	IT.	CT.	Err.	IT.	CT.	Err.
10	23	2.54	$5.4 \times 10^{-8}$	31	6.84	$8.1 \times 10^{-8}$	23	2.89	$9.6 \times 10^{-8}$
20	24	2.68	$7.5 \times 10^{-8}$	31	6.96	$8.9 \times 10^{-8}$	25	3.14	$5.0 \times 10^{-8}$
50	28	3.34	$5.1 \times 10^{-8}$	31	6.85	$7.6 \times 10^{-8}$	27	3.56	$7.6 \times 10^{-8}$
80	31	3.84	$4.2 \times 10^{-8}$	31	7.41	$8.3 \times 10^{-8}$	29	4.04	$8.1 \times 10^{-8}$
100	32	4.13	$7.5 \times 10^{-8}$	32	7.68	$7.0 \times 10^{-8}$	30	4.45	$9.4 \times 10^{-8}$
150	38	5.52	$7.6 \times 10^{-8}$	35	8.97	$5.7 \times 10^{-8}$	33	5.35	$8.9 \times 10^{-8}$
200	43	6.74	$6.8 \times 10^{-8}$	37	9.83	$8.9 \times 10^{-8}$	36	6.24	$7.5 \times 10^{-8}$

- [18] R. Glowinski and A. Marrocco, *Approximation par éléments finis d'ordre un et résolution par pénalisation-dualité d'une classe de problèmes non linéaires*, R.A.I.R.O., 1975, pp. 41-76.
- [19] 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.
- [20] B. S. He and H. Yang, *Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities*, Oper. Res. Lett., 23 (1998), pp. 151–161.
- [21] M. R. Hestenes, *Multiplier and gradient methods*, J. Optim. Theory Appl., 4 (1969), pp. 303–320.
- [22] J. B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, 1993.
- [23] H. Minc, *Nonnegative Matrices*, John Wiley&Sons, New York, 1988.
- [24] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 1999.
- [25] R. Orsi, *Numerical methods for solving inverse eigenvalue problems for nonnegative matrices*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 190–212.
- [26] R. T. Rockafellar, *Augmented Lagrangains and applications of the proximal point algorithm in convex programming*, Math. Oper. Res., 1 (1976), pp. 97–116.

Table 10: Numerical results for Example 4.9

$p = 30, \beta = 1.0, \text{ and } \delta = \gamma = 1.9$									
	ADMM			MADMM			RADMM		
n	IT.	CT.	Err.	IT.	CT.	Err.	IT.	CT.	Err.
100	43	0.28	$7.9 \times 10^{-8}$	25	0.19	$6.6 \times 10^{-8}$	19	0.13	$8.8 \times 10^{-8}$
200	27	0.30	$8.2 \times 10^{-8}$	16	0.22	$3.4 \times 10^{-8}$	12	0.14	$5.8 \times 10^{-8}$
500	26	1.58	$7.9 \times 10^{-8}$	14	1.36	$8.0 \times 10^{-9}$	10	0.67	$6.2 \times 10^{-8}$
1,000	23	5.57	$7.1 \times 10^{-8}$	13	4.59	$2.5 \times 10^{-8}$	8	2.09	$5.3 \times 10^{-8}$
1,500	21	11.06	$6.6 \times 10^{-8}$	13	10.36	$9.6 \times 10^{-8}$	7	4.26	$7.1 \times 10^{-8}$
2,000	20	18.97	$7.0 \times 10^{-8}$	11	15.82	$7.2 \times 10^{-8}$	7	7.72	$4.9 \times 10^{-8}$
$n = 1000, \beta = 1.0, \text{ and } \delta = \gamma = 1.9$									
	ADMM			MADMM			RADMM		
p	IT.	CT.	Err.	IT.	CT.	Err.	IT.	CT.	Err.
10	18	4.29	$9.0 \times 10^{-8}$	11	3.93	$2.9 \times 10^{-8}$	7	1.87	$1.9 \times 10^{-8}$
20	20	4.68	$7.5 \times 10^{-8}$	11	3.93	$3.6 \times 10^{-8}$	7	1.84	$6.0 \times 10^{-8}$
50	24	6.01	$9.9 \times 10^{-8}$	15	5.24	$3.3 \times 10^{-8}$	9	2.42	$4.9 \times 10^{-8}$
80	27	7.26	$7.9 \times 10^{-8}$	14	5.55	$2.4 \times 10^{-8}$	11	3.24	$2.1 \times 10^{-8}$
100	30	8.71	$9.2 \times 10^{-8}$	12	4.81	$6.7 \times 10^{-8}$	12	3.68	$6.9 \times 10^{-8}$
150	35	11.67	$9.6 \times 10^{-8}$	22	9.55	$9.6 \times 10^{-8}$	15	5.10	$6.8 \times 10^{-8}$
200	44	15.38	$7.5 \times 10^{-8}$	27	12.70	$5.9 \times 10^{-8}$	20	7.60	$4.6 \times 10^{-8}$

- [27] E. Senata, *Non-negative Matrices and Markov Chains*, 2nd rev. ed., Springer-Verlag, New York, 2006.
- [28] J. G. Sun, *Backward perturbation analysis of certain characteristic subspaces*, Numer. Math., 65 (1993), 357–382.
- [29] R. H. Tütüncü, K. C. Toh, and M. J. Todd, *Solving semidefinite-quadratic-linear programs using SDPT3*. Math. Program., 95 (2003), pp. 189–217. <http://www.math.nus.edu.sg/mat-tohkc/sdpt3.html>
- [30] C. H. Ye and X. M. Yuan, *A descent method for structured monotone variational inequalities*, Optim. Methods Softw., 22 (2007), pp. 329–338.

Table 11: Numerical results for Example 4.10

$p = 30, \beta = 5, \text{ and } \delta = \gamma = 1.6$									
	ADMM			MADMM			RADMM		
n	IT.	CT.	Err.	IT.	CT.	Err.	IT.	CT.	Err.
100	58	0.41	$9.2 \times 10^{-8}$	41	0.30	$8.7 \times 10^{-8}$	40	0.27	$8.6 \times 10^{-8}$
200	39	0.44	$6.5 \times 10^{-8}$	33	0.47	$7.0 \times 10^{-8}$	32	0.41	$6.2 \times 10^{-8}$
500	25	1.62	$5.2 \times 10^{-8}$	27	2.45	$5.7 \times 10^{-8}$	25	1.81	$4.8 \times 10^{-8}$
1,000	21	5.23	$7.1 \times 10^{-8}$	27	9.49	$5.9 \times 10^{-8}$	22	5.74	$8.4 \times 10^{-8}$
1,500	20	10.87	$5.5 \times 10^{-8}$	26	20.55	$9.5 \times 10^{-8}$	22	12.87	$4.6 \times 10^{-8}$
2,000	18	17.54	$7.2 \times 10^{-8}$	27	36.68	$6.6 \times 10^{-8}$	21	21.37	$8.4 \times 10^{-8}$
$n = 1000, \beta = 5, \text{ and } \delta = \gamma = 1.6$									
	ADMM			MADMM			RADMM		
p	IT.	CT.	Err.	IT.	CT.	Err.	IT.	CT.	Err.
10	17	3.88	$2.6 \times 10^{-8}$	25	8.28	$6.8 \times 10^{-8}$	20	4.90	$6.7 \times 10^{-8}$
20	19	4.46	$9.6 \times 10^{-8}$	26	8.64	$7.3 \times 10^{-8}$	21	5.13	$8.6 \times 10^{-8}$
50	24	6.05	$5.1 \times 10^{-8}$	27	9.78	$8.1 \times 10^{-8}$	24	6.32	$9.1 \times 10^{-8}$
80	28	7.71	$6.3 \times 10^{-8}$	28	10.70	$7.4 \times 10^{-8}$	27	7.75	$5.8 \times 10^{-8}$
100	30	8.53	$6.0 \times 10^{-8}$	29	11.39	$7.5 \times 10^{-8}$	28	8.53	$8.0 \times 10^{-8}$
150	36	11.58	$7.3 \times 10^{-8}$	33	14.50	$8.4 \times 10^{-8}$	32	10.78	$8.1 \times 10^{-8}$
200	43	15.23	$6.6 \times 10^{-8}$	37	17.36	$6.9 \times 10^{-8}$	35	13.34	$9.5 \times 10^{-8}$