Semidefinite Inverse Eigenvalue Problems with Prescribed Entries and Partial Eigendata

Teng-Teng Yao[*] Zheng-Jian Bai[†]

December 30, 2014

Abstract

In this paper, we study the semidefinite inverse eigenvalue problem of reconstructing a real n-by-n matrix C such that it is nearest to the original pre-estimated real n-by-n matrix C_o in the Frobenius norm and satisfies the measured partial eigendata, where the required matrix C should preserve the symmetry, positive semidefiniteness, and the prescribed entries of the pre-estimated matrix C_o. We propose the alternating direction method of multipliers for solving the semidefinite inverse eigenvalue problem, where three related iterative algorithms are presented. We also extend our method to the case of lower bounds. Numerical experiments are reported to illustrate the efficiency of the proposed method for solving semidefinite inverse problems.

Keywords. Inverse eigenvalue problem, positive semidefiniteness, prescribed entries, alternating direction method of multipliers.

AMS subject classifications. 65F10, 65F15, 65F18, 90C25

1 Introduction

We consider the following semidefinite inverse eigenvalue problem with prescribed entries and partial eigendata:

PESDIEP *Given a pre-estimated real symmetric and positive definite matrix $C_o \in \mathbb{R}^{n\times n}$ and a set of measured eigendata $\{(\lambda_k, \mathbf{x}_k) \in \mathbb{R}\times\mathbb{R}^n\}_{k=1}^{p} (p \ll n)$, find a matrix $C \in \mathbb{R}^{n\times n}$ such that it is closest to the original matrix C_o in the Frobenius norm, satisfies the measured eigendata $\{(\lambda_k, \mathbf{x}_k)\}_{k=1}^{p}$, and retains the symmetry, positive semidefiniteness and prescribed entries $\{(C_o)_{ij} \mid (i,j) \in \mathcal{I}_s\}$ of the pre-estimated matrix C_o, where $\mathcal{I}_s \subset \mathcal{N} := \{(i,j) \mid i,j = 1,\ldots,n\}$ is a given index subset such that $(j,i) \in \mathcal{I}_s$ if $(i,j) \in \mathcal{I}_s$.*

The PESDIEP is a kind of structured inverse eigenvalue problems, which arise in many applications such that structural dynamics, vibrations, control design, circuit theory, inverse

[*] School of Mathematical Sciences, Xiamen University, Xiamen 361005, People's Republic of China (yaotengteng718@163.com).

[†] School of Mathematical Sciences and Fujian Provincial Key Laboratory on Mathematical Modeling & High Performance Scientific Computing, Xiamen University, Xiamen 361005, People's Republic of China (zjbai@xmu.edu.cn).

Sturm-Liouville problems, applied physics, finite element model updating, etc. For the applications, mathematical theory, and numerical methods on structured inverse eigenvalue problems, one may refer to, for instance, the survey papers [16, 19] and the books [20, 28, 32, 55] and the references therein.

The PESDIEP plays an important role in many applications such as the finite element model updating in structural dynamics and vibration [7, 8, 20, 28, 32]. In practice, the pre-estimated analytic matrix C_o is a physical matrix whose entries involves some physical parameters such as mass, stiffness, length, elasticity, inductance, capacitance, etc. In general, the physical analytical matrix C_o possesses some specific structural constraints (e.g., symmetry, definiteness, sparsity or bandedness). However, the natural frequencies and mode shapes (i.e., eigenvalues of eigenvectors) predicted by the analytic matrix C_o often do not match with experimentally measured frequencies and mode shapes. To ensure the validity of the original model, one may update or correct the original analytic matrix C_o via the prescribed partial eigendata, which can be experimentally measured from a practical structure. It is desired to update the original matrix C_o with minimal changes. This requires that the updated matrix C should be closest to the original analytic matrix C_o, say, in the Frobenius norm and satisfies the measured eigendata. More importantly, the updated matrix C should preserve various structural constraints of C_o simultaneously. That is, the corrected matrix C should keep the symmetry, definiteness, and sparsity (i.e., prescribed entries) of the analytic matrices C_o unchanged.

Let

$$\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_p) \in \mathbb{R}^{p\times p}, \quad X = [\mathbf{x}_1, \ldots, \mathbf{x}_p] \in \mathbb{R}^{n\times p}.$$

Then, the PESDIEP is to solve the following minimization problem.

$$\begin{array}{ll} \min & \frac{1}{2}\|C - C_o\|^2 \\ \text{subject to (s.t.)} & CX = X\Lambda, \\ & C_{ij} = (C_o)_{ij} \quad \forall (i,j) \in \mathcal{I}_s, \\ & C \in \mathcal{S}^n_+, \end{array} \tag{1}$$

where $\|\cdot\|$ denotes the Frobenius matrix norm or the Euclidean vector norm and $\mathcal{S}^n$ and $\mathcal{S}^n_+$ denote the set of all $n\times n$ real symmetric matrices and the set of all $n\times n$ real symmetric and positive semidefinite matices, respectively. Without causing any confusion, we regard the minimization problem (1) as the PESDIEP.

As noted in [43], one may find a solution to the PESDIEP (1) by using classical semidefinite programming (SDP) techniques (see for instance [1, 2, 53]). However, the primal-dual interior-point methods may not be effective for solving large-scale semidefinite programming problems [53]. In many applications, the problem size of the PESDIEP is very large (say, $n \geq 1,000$). In this case, the number np of linear constraints in the PESDIEP (1) is much large even when the number p of given eigenpairs is small (e.g., when $n \geq 1,000$ and $p = 30$, $np \geq 30,000$). By dropping the requirement of partial entries (i.e., $C_{ij} = (C_o)_{ij}\ \forall (i,j) \in \mathcal{I}_s$), one may solve the simplified version of PESDIEP (1) by the semismooth Newton method proposed in [4, 48]. But the requirement of prescribed entries is vital for practical applications, e.g., it is essential to preserve the sparsity of the original physical matrix C_o. In this case, the number $|\mathcal{I}_s|$ of prescribed entries is very large.

Recently, the alternating direction method of multipliers (ADMM), which is proposed by Glowinski and Marrocco [33], has been used in many areas: optimization, image processing and statistical learning, etc. One may refer to the survey paper [12] and references therein for the applications of the ADMM. In this paper, we propose several iterative algorithms based on the ADMM for solving the PESDIEP. This is motivated by the recent papers due to He, Xu, and Yuan [37] and Zhao, Bai, and Chen [57]. In [37], He, Xu, and Yuan introduced an ADMM for solving large-scale semidefinite programming. In [57], the ADMM is successfully applied to nonnegative inverse eigenvalue problems with partial eigendata. We shall present three ADMM-based iterative algorithms for solving the PESDIEP by adding two auxiliary matrix variables so that the resulted two subproblems can be handled easily, where one of the subproblems has closed-form solution and the other is a quadratic minimization problem which can be solved efficiently by solving its dual problem. We also extend the proposed method to the case of lower bounds. We report some numerical tests, including the comparison with the interior-point approach mentioned in [43, 53] for solving the PESDIEP, to illustrate the effectiveness of our method.

Throughout the paper, we use the following notations. The symbol A^T denotes the transpose of a matrix A. I is the identity matrix of an appropriate dimension. Let $\|\cdot\|_{\max}$ denote the entry of largest absolute value of a matrix. Let $\mathcal{D} \subseteq \mathbb{R}^{n\times n}$ (or $\mathcal{S}^n$) be a closed convex set and $\Pi_{\mathcal{D}}\{\cdot\}$ denote the metric projection onto $\mathcal{D}$.

The remainder of the paper is organized as follows. In section 2 we propose several ADMM-based iterative algorithms for solving the PESDIEP. In section 3 we discuss some extensions. In section 4 we report some numerical tests.

2 An Alternating Direction Method of Multipliers

2.1 Problem reformulation

Let $\mathcal{S}^n$ be equipped with the Frobenius inner product

$$\langle A, B\rangle = \mathrm{tr}(AB) \quad \forall A, B \in \mathcal{S}^n,$$

where "tr" means the trace of a matrix. To apply the ADMM to the PESDIEP, one possible way is to rewrite the PESDIEP (1) as the following form

$$\begin{array}{ll} \min & \frac{1}{2}\|C - C_o\|^2 + \frac{1}{2}\|Y - C_o\|^2 \\ \text{s.t.} & C - Y = 0, \\ & C \in \mathcal{S}^n_+, \quad Y \in S_B, \end{array} \tag{2}$$

where $S_B := \{Y \in \mathcal{S}^n \mid YX = X\Lambda,\ Y_{ij} = (C_o)_{ij}\ \forall (i,j) \in \mathcal{I}_s\}$.

The augmented Lagrangian function for Problem (2) is given by [39, 47]

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

where $\beta > 0$ is a penalty parameter and $Z \in \mathcal{S}^n$ is the Lagrange multiplier. Then, the classical method of multipliers for Problem (2) can be written as [10, 39, 47]

$$\begin{cases} (C^{k+1}, Y^{k+1}) = \underset{C \in \mathcal{S}^n_+,\, Y \in S_B}{\operatorname{argmin}} \; AL_\beta(C, Y, Z^k), \\ Z^{k+1} = Z^k - \beta(C^{k+1} - Y^{k+1}), \end{cases}$$

where (C^k, Y^k, Z^k) is the current iterate. We observe that Problem (2) involves an objective function, which can be separated into two individual convex functions with the matrix variable C and the matrix variable Y, respectively. Hence, we get the following ADMM [30, 33]

$$\begin{cases} C^{k+1} = \underset{C \in \mathcal{S}^n_+}{\operatorname{argmin}} \, AL_\beta(C, Y^k, Z^k), \\ Y^{k+1} = \underset{Y \in S_B}{\operatorname{argmin}} \, AL_\beta(C^{k+1}, Y, Z^k), \\ Z^{k+1} = Z^k - \beta(C^{k+1} - Y^{k+1}), \end{cases} \tag{3}$$

where (C^k, Y^k, Z^k) is the current iterate.

2.2 Subproblems

We note that there are two minimization subproblems in Problem (3). By simple calculation, we know that these subproblems have the following closed-form analytical solutions.

$$\begin{cases} C^{k+1} &= \Pi_{\mathcal{S}^n_+}\big\{\frac{1}{1+\beta}(C_o + Z^k + \beta Y^k)\big\}, \\ Y^{k+1} &= \Pi_{S_B}\big\{\frac{1}{1+\beta}(C_o - Z^k + \beta C^{k+1})\big\}. \end{cases}$$

For any given $V \in \mathcal{S}^n$, $\Pi_{\mathcal{S}^n_+}(V)$ has explicit formula if the spectral decomposition of V are computed [4, 40]. In particular, let the spectral decomposition of V be given by [36]

$$V = Q\Theta Q^T, \quad \Theta = \operatorname{diag}(\theta_1, \ldots, \theta_n),$$

where $\theta_1, \ldots, \theta_n$ are eigenvalues of V and $Q \in \mathbb{R}^{n \times n}$ is an orthogonal matrix, whose columns are orthonormal eigenvectors of V. Then, $\Pi_{\mathcal{S}^n_+}(V)$ admits the following explicit analytic formula (e.g., [4, 40])

$$\Pi_{\mathcal{S}^n_+}(V) = Q\Theta_+ Q^T, \quad \Theta_+ = \operatorname{diag}(\max\{\theta_1, 0\}, \ldots, \max\{\theta_n, 0\}).$$

In our numerical experiments, we use the efficient Matlab Mex interface[1] for spectral decomposition.

Next, we consider how to compute $\Pi_{S_B}(V)$ for a given $V \in \mathcal{S}^n$. In fact, $\Pi_{S_B}(V)$ is the unique solution of the following quadratic programming:

$$\begin{array}{ll} \min\limits_{Y \in \mathcal{S}^n} & \dfrac{1}{2}\|Y - V\|^2 \\ \text{s.t.} & YX = X\Lambda, \\ & Y_{ij} = (C_o)_{ij} \quad \forall (i,j) \in \mathcal{I}_s. \end{array} \tag{4}$$

[1] which computes the spectral decomposition via a divide-and-conquer routine (dgesdd) implemented in LAPACK, and its code can be downdloaded at: `http://videoprocessing.ucsd.edu/~karl/software.html`.

There are many existing methods in the literature for solving Problem (4) (see for instance [45]). Here, we solve Problem (4) as follows. Let $\mathcal{J}_s := \mathcal{N}\backslash\mathcal{I}_s$. Define $\widehat{\mathcal{I}}_s := \{(i,j) \mid (i,j) \in \mathcal{I}_s,\ i \leq j\}$ and $\widehat{\mathcal{J}}_s := \{(i,j) \mid (i,j) \in \mathcal{J}_s,\ i \leq j\}$. For a matrix $A \in \mathcal{S}^n$, let $A_{\widehat{\mathcal{I}}_s}$ denote the column vector with entries A_{ij} for all $(i,j) \in \widehat{\mathcal{I}}_s$. Define the linear operator $P : \mathbb{R}^{|\widehat{\mathcal{I}}_s|} \to \mathcal{S}^n$ by

$$P_{ij}(A_{\widehat{\mathcal{I}}_s}) := \begin{cases} A_{ij} & \text{if } (i,j) \in \widehat{\mathcal{I}}_s, \\ A_{ji} & \text{if } (j,i) \in \widehat{\mathcal{I}}_s, \\ 0 & \text{otherwise.} \end{cases}$$

It follows easily that $\overline{Y} \in \mathcal{S}^n$ is a solution to Problem (4) if and only if $\overline{Y}_{\widehat{\mathcal{I}}_s} = (C_o)_{\widehat{\mathcal{I}}_s}$ and $\overline{Y}_{\widehat{\mathcal{J}}_s} \in \mathbb{R}^{|\widehat{\mathcal{J}}_s|}$ solve the following minimization problem:

$$\begin{array}{ll} \min\limits_{Y_{\widehat{\mathcal{J}}_s} \in \mathbb{R}^{|\widehat{\mathcal{J}}_s|}} & \frac{1}{2}\big\|Y_{\widehat{\mathcal{J}}_s} - V_{\widehat{\mathcal{J}}_s}\big\|^2 \\ \text{s.t.} & \mathcal{H}(Y_{\widehat{\mathcal{J}}_s}) := P(Y_{\widehat{\mathcal{J}}_s})X = X\Lambda - P\big((C_o)_{\widehat{\mathcal{I}}_s}\big)X \equiv G. \end{array} \tag{5}$$

Now, we present a dual method for solving Problem (5) (see for instance [45]). Let

$$\mathcal{R}(\mathcal{H}) := \big\{\mathcal{H}(Y_{\widehat{\mathcal{J}}_s}) \mid Y_{\widehat{\mathcal{J}}_s} \in \mathbb{R}^{|\widehat{\mathcal{J}}_s|}\big\}.$$

Then the linear operator $\mathcal{H} : \mathbb{R}^{|\widehat{\mathcal{J}}_s|} \to \mathcal{R}(\mathcal{H})$ is surjective. The Lagrangian function $L : \mathbb{R}^{|\widehat{\mathcal{J}}_s|} \times \mathcal{R}(\mathcal{H}) \to \mathbb{R}$ for Problem (5) is given by

$$L(Y_{\widehat{\mathcal{J}}_s}, W) = \frac{1}{2}\big\|Y_{\widehat{\mathcal{J}}_s} - V_{\widehat{\mathcal{J}}_s}\big\|^2 - \langle \mathcal{H}(Y_{\widehat{\mathcal{J}}_s}) - G, W\rangle.$$

The Lagrangian dual function $\Phi : \mathcal{R}(\mathcal{H}) \to \mathbb{R}$ is defined as

$$\Phi(W) := \inf_{Y_{\widehat{\mathcal{J}}_s} \in \mathbb{R}^{|\widehat{\mathcal{J}}_s|}} L(Y_{\widehat{\mathcal{J}}_s}, W).$$

Since $L(Y_{\widehat{\mathcal{J}}_s}, W)$ is a convex quadratic function of $Y_{\widehat{\mathcal{J}}_s}$, one may get the minimum value of L in terms of $Y_{\widehat{\mathcal{J}}_s}$ by using the optimality condition

$$\nabla_{Y_{\widehat{\mathcal{J}}_s}} L(Y_{\widehat{\mathcal{J}}_s}, W) = Y_{\widehat{\mathcal{J}}_s} - V_{\widehat{\mathcal{J}}_s} - \mathcal{H}^*(W) = 0, \tag{6}$$

where the adjoint $\mathcal{H}^* : \mathcal{R}(\mathcal{H}) \to \mathbb{R}^{|\widehat{\mathcal{J}}_s|}$ of $\mathcal{H}$ is given by

$$\mathcal{H}^*(W) := \frac{1}{2}(WX^T + XW^T)_{\widehat{\mathcal{J}}_s} \quad \forall W \in \mathcal{R}(\mathcal{H}).$$

Thus, the Lagrangian dual function is given by

$$\begin{aligned} \Phi(W) &= \frac{1}{2}\left\|\mathcal{H}^*(W)\right\|^2 - \langle \mathcal{H}(V_{\widehat{\mathcal{J}}_s} + \mathcal{H}^*(W)) - G, W\rangle \\ &= \frac{1}{2}\left\|\mathcal{H}^*(W)\right\|^2 - \langle V_{\widehat{\mathcal{J}}_s} + \mathcal{H}^*(W), H^*(W)\rangle + \langle G, W\rangle \\ &= -\frac{1}{2}\left\|\mathcal{H}^*(W)\right\|^2 - \langle V_{\widehat{\mathcal{J}}_s}, H^*(W)\rangle + \langle G, W\rangle \\ &= -\frac{1}{2}\big\|V_{\widehat{\mathcal{J}}_s} + \mathcal{H}^*(W)\big\|^2 + \langle G, W\rangle + \frac{1}{2}\big\|V_{\widehat{\mathcal{J}}_s}\big\|^2. \end{aligned}$$

Hence, the Lagrangian dual problem is given by

$$\max_{W\in\mathcal{R}(\mathcal{H})} \quad \Phi(W) \tag{7}$$

or

$$\min_{W\in\mathcal{R}(\mathcal{H})} \quad \Psi(W) := \frac{1}{2}\big\|V_{\widehat{\mathcal{J}}_s} + \mathcal{H}^*(W)\big\|^2 - \langle G, W\rangle - \frac{1}{2}\big\|V_{\widehat{\mathcal{J}}_s}\big\|^2. \tag{8}$$

We note that $0 \in \mathcal{R}(\mathcal{H})$, Slater's condition (see for instance [13, §5.2.3]) holds for Problem (5). By using [49, Theorems 17 and 18], there exists a solution $\overline{W} \in \mathcal{R}(\mathcal{H})$ to the dual problem (7) (or Problem (8)) such that $\Phi(\overline{W})$ is equal to the optimal value of the dual problem (7) and is also equal to the optimal value of the original problem (5). That is, the optimal duality gap is zero. Once a solution $\overline{W} \in \mathcal{R}(\mathcal{H})$ to the dual problem (7) is computed, by (6), the unique solution to the original problem (5) is given by

$$\overline{Y}_{\widehat{\mathcal{J}}_s} = V_{\widehat{\mathcal{J}}_s} + \mathcal{H}^*(\overline{W}).$$

Therefore, we only need to solve the dual problem (7), i.e., Problem (8). A solution $\overline{W} \in \mathcal{R}(\mathcal{H})$ of the dual problem (8) solves

$$\nabla\Psi(W) = \mathcal{H}\big(V_{\widehat{\mathcal{J}}_s} + \mathcal{H}^*(W)\big) - G = 0 \tag{9}$$

or equivalently,

$$G - \mathcal{H}(V_{\widehat{\mathcal{J}}_s}) = \mathcal{H}(\mathcal{H}^*(W)) = \frac{1}{2}P\big((WX^T + XW^T)_{\widehat{\mathcal{J}}_s}\big)X.$$

We note that the linear operator $\mathcal{H} : \mathbb{R}^{|\widehat{\mathcal{J}}_s|} \to \mathcal{R}(\mathcal{H})$ defined in Problem (5) is onto, $\mathcal{H}\mathcal{H}^*$ is self-adjoint and positive definite. Therefore, one may apply the conjugate gradient (CG) method [36, Algorithm 10.2.1] to solving the linear system (9).

Remark 2.1 *The total cost of computing a solution $\overline{W} \in \mathcal{R}(\mathcal{H})$ to the linear system* (9) *is $O(n^3p^3)$ flops. This cost is practically acceptable since the number of prescribed eigenpairs is very small. The numerical tests in section* 4 *demonstrate that all the ADMM-based algorithms to be proposed perform much efficiently for the large PESDIEP over the semidefinite programming techniques (e.g., the interior point method) mentioned in [43, 54].*

2.3 ADMM-based algorithms

In this section, we propose three ADMM-based iterative algorithms for solving Problem (2). For convenience, for the current iterate (C^k, Y^k, Z^k), let $(\widetilde{C}^k, \widetilde{Y}^k, \widetilde{Z}^k)$ be the solution to (3). The first algorithm is the classical ADMM as in (3), i.e.,

$$\begin{cases} C^{k+1} &=& \widetilde{C}^k, \\ Y^{k+1} &=& \widetilde{Y}^k, \\ Z^{k+1} &=& \widetilde{Z}^k. \end{cases}$$

Next, by using the descent method for structured monotone variational inequalities [56], we propose the following modified ADMM (MADMM) for solving Problem (2):

$$\begin{cases} C^{k+1} = \widetilde{C}^k, \\ Y^{k+1} = Y^k - \delta\eta_k(Y^k - \widetilde{Y}^k), \\ Z^{k+1} = Z^k - \delta\eta_k(Z^k - \widetilde{Z}^k), \end{cases}$$

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

$$\eta_k = \frac{1}{\xi_k}\big(\xi_k - \langle Y^k - \widetilde{Y}^k, Z^k - \widetilde{Z}^k\rangle\big), \quad \xi_k = \beta\|Y^k - \widetilde{Y}^k\|^2 + \frac{1}{\beta}\|Z^k - \widetilde{Z}^k\|^2.$$

Finally, by applying the acceleration technique in [14], we obtain the following relaxed (acceleration) ADMM (RADMM) for solving Problem (2):

$$\begin{cases} \widetilde{C}^k = \operatorname*{argmin}_{C \in \mathcal{S}^n_+} AL_\beta(C, Y^k, Z^k), \\ \widetilde{Z}^k = Z^k - \beta(\widetilde{C}^k - Y^k), \\ \widetilde{Y}^k = \operatorname*{argmin}_{Y \in S_B} AL_\beta(\widetilde{C}^k, Y, \widetilde{Z}^k), \\ C^{k+1} = \widetilde{C}^k, \\ Y^{k+1} = Y^k - \mu(Y^k - \widetilde{Y}^k), \\ Z^{k+1} = Z^k - \mu(Z^k - \widetilde{Z}^k), \end{cases}$$

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

For the convergence of the above ADMM-based algorithms, one may see [14, 29, 30, 33, 38, 37, 56].

Finally, as in [37], we provide a feasible stopping criterion for the proposed ADMM-based iterative algorithms. We first 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 $Z^* \in \mathcal{S}^n$ such that the following variational inequalities hold [45]

$$\begin{cases} \langle C - C^*, C^* - C_o - Z^*\rangle \geq 0 \quad \forall C \in \mathcal{S}^n_+, \\ \langle Y - Y^*, Y^* - C_o + Z^*\rangle \geq 0 \quad \forall Y \in S_B, \\ \langle Z - Z^*, C^* - Y^*\rangle \geq 0 \quad \forall Z \in \mathcal{S}^n. \end{cases} \tag{10}$$

It is easy to check that the iterate $(\widetilde{C}^k, \widetilde{Y}^k, \widetilde{Z}^k)$ generated by (3) satisfies the following variational inequalities and equation

$$\begin{cases} \langle C - \widetilde{C}^k, \widetilde{C}^k - C_o - \widetilde{Z}^k - \beta(Y^k - \widetilde{Y}^k))\rangle \geq 0 \quad \forall C \in \mathcal{S}^n_+, \\ \langle Y - \widetilde{Y}^k, \widetilde{Y}^k - C_o + \widetilde{Z}^k\rangle \geq 0 \quad \forall Y \in S_B, \\ \langle Z - \widetilde{Z}^k, \widetilde{C}^k - \widetilde{Y}^k + \frac{1}{\beta}(\widetilde{Z}^k - Z^k\rangle \geq 0 \quad \forall Z \in \mathcal{S}^n. \end{cases} \tag{11}$$

By (10) and (11), $(\widetilde{C}^k, \widetilde{Y}^k, \widetilde{Z}^k)$ is a solution to (10) if and only if $Y^k - \widetilde{Y}^k = 0$ and $Z^k - \widetilde{Z}^k = 0$. Therefore, the proposed ADMM-based algorithms can be stopped when

$$\max\left\{ \frac{\|Y^k - Y^{k+1}\|_{\max}}{\|Y^0 - Y^1\|_{\max}}, \frac{\|Z^k - Z^{k+1}\|_{\max}}{\|Z^0 - Z^1\|_{\max}} \right\} \leq \epsilon, \tag{12}$$

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

3 Extensions

In this section, we extend the proposed ADMM-based algorithms to the case of lower bounds. In many applications, the updated matrix $C \in \mathcal{S}^n$ should be positive definite [28]. This can be guaranteed if the minimal eigenvalue of C is greater than a prescribed scalar $\gamma > 0$. As in [48], the PESDIEP with lower bounds can be defined as follows.

$$\begin{aligned} \min \quad & \frac{1}{2}\|C - C_o\|^2 \\ \text{s.t.} \quad & CX = X\Lambda, \\ & C_{ij} = (C_o)_{ij} \quad \forall (i,j) \in \mathcal{I}_s, \\ & C \succeq \gamma I, \end{aligned} \tag{13}$$

where $\gamma > 0$ and $C \succeq \gamma I$ means that $C - \gamma I \in \mathcal{S}^n_+$. We consider the following minimization problem:

$$\begin{aligned} \min \quad & \frac{1}{2}\|C - (C_o - \gamma I)\|^2 \\ \text{s.t.} \quad & CX = X\Lambda - \gamma X, \\ & (C)_{ij} = (C_o - \gamma I)_{ij} \quad \forall (i,j) \in \mathcal{I}_s, \\ & C \in \mathcal{S}^n_+. \end{aligned} \tag{14}$$

We note that if $\overline{C}$ is the solution of Problem (14), then the unique solution of the PESDIEP with lower bounds (13) is give by

$$C^* = \overline{C} + \gamma I.$$

Thus we only need to focus on Problem (14). An ADMM-based reformulation of Problem (14) is given by

$$\begin{aligned} \min \quad & \frac{1}{2}\|C - C_o\|^2 + \frac{1}{2}\|Y - C_o\|^2 \\ \text{s.t.} \quad & C - Y = 0, \\ & C \in \mathcal{S}^n_+, \quad Y \in S^{lb}_B, \end{aligned} \tag{15}$$

where $S^{lb}_B := \{Y \in \mathcal{S}^n \mid YX = X\Lambda - \gamma X,\ Y_{ij} = (C_o - \gamma I)_{ij}\ \forall (i,j) \in \mathcal{I}_s\}$.

As in section 2, we can easily develop the ADMM-based iterative algorithms for solving Problem (15).

4 Numerical Tests

In this section, we report some numerical results to show the efficiency of the proposed ADMM-based iterative algorithms for solving the PESDIEP. All the numerical tests are carried out in `MATLAB 8.1` running on a personal computer of 2.30 GHz CPU and 12.0 GB of RAM. In our numerical tests, we choose the starting point $C^0 = Y^0 = C_o$ and $Z^0 = 0$ for all ADMM-based iterative algorithms. We set the stopping criterion to be (12). We solve the linear equation (9) by the CG method, which is stopped if the maximum number of iterations reaches 1000 or the tolerance is less than 10^{-8}. In what follows, `IT.`, `CT.`, and `Res.` denote the numbers of iterations, the total computing time (in seconds), and the residual $(\|C^kX - X\Lambda\|^2 + \sum_{(i,j)\in\mathcal{I}_s}((C^k)_{ij} - (C_o)_{ij})^2)^{1/2}$ at the final iterate of the proposed algorithms, respectively.

4.1 Small problems

In this section, we report some numerical results for the small PESDIEP with $\epsilon = 10^{-8}$.

Example 4.1 *We consider the PESDIEP with $n = 6$ and $p = 2$. Let $\widehat{C}$ be a random $n \times n$ correlation matrix given by*

$$\widehat{C} = \begin{bmatrix} 1.0000 & 0.0764 & 0.2063 & -0.0419 & -0.3358 & 0.1113 \\ 0.0764 & 1.0000 & 0.2057 & -0.5707 & -0.2011 & -0.3215 \\ 0.2063 & 0.2057 & 1.0000 & 0.1272 & 0.2629 & -0.1643 \\ -0.0419 & -0.5707 & 0.1272 & 1.0000 & -0.1665 & 0.1262 \\ -0.3358 & -0.2011 & 0.2629 & -0.1665 & 1.0000 & 0.2569 \\ 0.1113 & -0.3215 & -0.1643 & 0.1262 & 0.2569 & 1.0000 \end{bmatrix}.$$

Then we set

$$C_o = \begin{bmatrix} \mathbf{1.0000} & 0.0841 & 0.2372 & -0.0473 & -0.3266 & 0.1261 \\ 0.0841 & \mathbf{1.0000} & 0.2369 & -\mathbf{0.5707} & -0.1791 & -0.3365 \\ 0.2372 & 0.2369 & \mathbf{1.0000} & 0.1216 & 0.2695 & -0.1934 \\ -0.0473 & -\mathbf{0.5707} & 0.1216 & \mathbf{1.0000} & -0.1566 & 0.1062 \\ -0.3266 & -0.1791 & 0.2695 & -0.1566 & \mathbf{1.0000} & 0.2703 \\ 0.1261 & -0.3365 & -0.1934 & 0.1062 & 0.2703 & \mathbf{1.0000} \end{bmatrix},$$

where $\mathcal{I}_s = \{(1,1),(2,2),(3,3),(4,4),(5,5),(6,6),(2,4),(4,2)\}$. We use the two eigenvalues $\{1.7761, 1.3629\}$ of $\widehat{C}$ and associated eigenvectors as prescribed eigendata.

By applying the proposed ADMM-based algorithms with $\beta = 20$ and $\delta = \gamma = 1.0$ to Example 4.1, we get the physical solution as follows:

$$\overline{C} = \begin{bmatrix} \mathbf{1.0000} & 0.0899 & 0.2254 & -0.0290 & -0.3359 & 0.1239 \\ 0.0899 & \mathbf{1.0000} & 0.2044 & -\mathbf{0.5707} & -0.1914 & -0.3219 \\ 0.2254 & 0.2044 & \mathbf{1.0000} & 0.1262 & 0.2758 & -0.1645 \\ -0.0290 & -\mathbf{0.5707} & 0.1262 & \mathbf{1.0000} & -0.1573 & 0.1260 \\ -0.3359 & -0.1914 & 0.2758 & -0.1573 & \mathbf{1.0000} & 0.2656 \\ 0.1239 & -0.3219 & -0.1645 & 0.1260 & 0.2656 & \mathbf{1.0000} \end{bmatrix}.$$

The numerical results are displayed in Table 1.

Table 1: Convergence results for Example 4.1

ADMM		MADMM		RADMM	
IT.	Res.	IT.	Res.	IT.	Res.
8	2.5×10^{-11}	16	1.7×10^{-10}	8	1.3×10^{-12}

Example 4.2 *This is a PESDIEP arising in structural engineering with* $n = 66$*. Let the matrix* $\widehat{C}$ *be BCSSTK02* (2211 *nonzero entries*)*, which comes from the set BCSSTRUC1 in the Harwell-Boeing collection[11]. Then we set*

$$C_o := \widehat{C} + R_C . * \widehat{C},$$

where "." means element-by-element multiplication and* R_C *is a real symmetric matrix whose entries are generated pseudo-randomly and they are uniformly distributed within* $[-0.2, 0.2]$*. We choose* p *eigenpairs of* $\widehat{C}$ *as prescribed eigendata and* $\mathcal{I}_s$ *is the index set corresponding to the entries of* C_o *whose values are equal to zeros.*

Table 2 lists the numerical results for solving Example 4.2 by the proposed ADMM-based algorithms with $\beta = 6$ and $\delta = \gamma = 1.0$.

Table 2: Convergence results for Example 4.2

	ADMM		MADMM		RADMM	
p	IT.	Res.	IT.	Res.	IT.	Res.
4	25	1.8×10^{-11}	25	2.2×10^{-11}	30	1.1×10^{-11}
8	24	1.9×10^{-11}	24	1.1×10^{-10}	29	1.4×10^{-11}
12	23	2.1×10^{-11}	24	8.6×10^{-9}	28	1.6×10^{-11}
16	24	2.4×10^{-11}	25	1.4×10^{-10}	29	2.0×10^{-11}

We observe from Tables 1 and 2 that the proposed ADMM-based algorithms are very effective for solving the small PESDIEP.

4.2 Comparison with an interior-point approach

In this section, we compare the proposed ADMM-based algorithms with the interior point method (IPM) in [54] for solving the PESDIEP. To use the IPM, one may turn the PESDIEP into the following standard form:

$$\begin{array}{ll} \min & \frac{1}{2} z \\ \text{s.t.} & CX = X\Lambda, \\ & C_{ij} = (C_o)_{ij} \quad \forall (i,j) \in \mathcal{I}_s, \\ & \sqrt{z} \geq \|C - C_o\|, \\ & C \in \mathcal{S}^n_+. \end{array}$$

For simplicity, this problem is solved by using the package SDPT3 [54]. We focus on the following example.

Example 4.3 *We consider the PESDIEP with different n and p. We first randomly generate an n-by-n correlation matrix $\widehat{C}$ by* `MATLAB 8.1's gallery('randcorr',n)`*. Then we set C_o to be*

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

where $r_{ij} = r_{ji}$ are random numbers in $[-0.2, 0.2]$ and $\mathcal{I}_s$ is the index set corresponding to the entries of C_o whose values are equal to or greater than 0.4. We randomly choose p eigenvalues of $\widehat{C}$ and associated eigenvectors as prescribed eigendata.

We set $\epsilon = 10^{-7}$. The IPM is implemented with the default tolerance. Table 3 shows the numerical results for Example 4.3.

We see from Table 3 that the proposed ADMM-based algorithms behavior more efficiently than the interior point algorithm in [54] for solving the PESDIEP. Here, we set $\beta = 15$ and $\delta = \gamma = 0.9$ for the proposed ADMM-based algorithms. We also see that there is a sudden jump in the iteration numbers of the proposed ADMM-based algorithms (see the case of $n = 200$ and $p = 10$). This means that, for the tested problem, the proposed ADMM-based algorithms may converge slowly after reaching a modest accuracy and $\beta = 15$ is not necessarily optimal for these algorithms. As noted in [31], the parameter β has a direct impact on the convergence of ADMM-based algorithms. The convergence of ADMM-based algorithms is often characterized in terms of the residuals [12, 31]

$$\|R^k\| := \|C^k - Y^k\|.$$

To illustrate the convergence of the proposed ADMM-based algorithms and the influence of β, for one test of Example 4.3 with $n = 200$ and $p = 10$, Figure 1 depicts the resulting residuals of the ADMM-based algorithms for $\beta = 15$ and $\delta = \gamma = 0.9$ and the number of iterations for $\delta = \gamma = 0.9$ and varying β. We see from Figure 1 that the ADMM generates acceptable residual ($\|R^k\| = O(10^{-7})$) after 7 iterations and then it converges slowly for a higher accuracy. For the MADMM and RADMM, we have similar observations. Moreover, for the tested problem, the optimal selection of β is between 2 and 6. As noted in [12, 25], one may improve the accuracy effectively by combing the proposed ADMM-based algorithms with Newton-type methods (e.g., the interior point algorithm in [54]). However, how to find the optimal parameter β is beyond the scope of this paper. We point out that there exist some techniques on how to choose the optimal parameters β, δ and γ for ADMM-based methods (see for instance [14, 24, 31, 34, 35, 56]).

4.3 Large problems

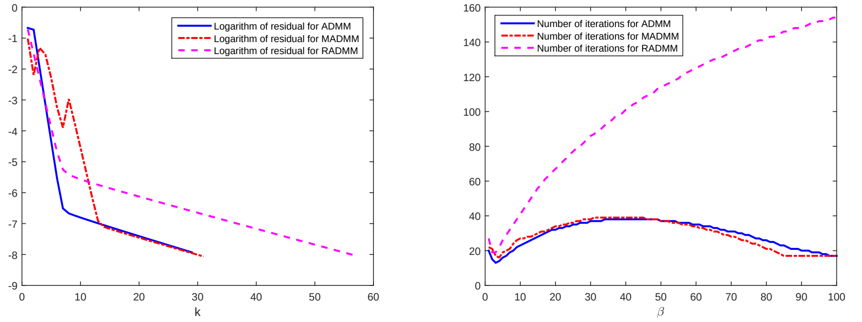
In this section, we report some numerical tests for solving the large PESDIEP by the proposed ADMM-based algorithms.

For demonstration purpose, in Table 4, we report the numerical results for Examples 4.3 with $\epsilon = 10^{-6}$.

We see from Table 4 that the proposed algorithms works efficiently for solving the large PESDIEP in terms of both the number of iterations and computing time. Moreover, our algorithms

Table 3: Comparison of ADMM and an interior-point approach for Example 4.3 ("—" means failed to converge)

		$\beta = 15$ and $\delta = \gamma = 0.9$											
		IPM			ADMM			MADMM			RADMM		
n	p	IT.	CT.	Res.	IT.	CT.	Res.	IT.	CT.	Res.	IT.	CT.	Res.
30	6	31	2.3250	1.6×10^{-8}	7	0.0470	5.7×10^{-9}	14	0.0320	1.4×10^{-9}	11	0.0310	1.8×10^{-10}
40	8	28	3.6660	2.2×10^{-8}	7	0.0470	6.9×10^{-9}	14	0.0630	1.7×10^{-9}	11	0.0460	2.0×10^{-10}
50	10	31	4.5870	1.2×10^{-8}	7	0.0780	6.7×10^{-9}	14	0.0940	1.6×10^{-9}	11	0.0620	1.9×10^{-10}
60	10	30	7.3170	1.9×10^{-8}	7	0.0470	6.8×10^{-9}	14	0.0780	1.6×10^{-9}	11	0.0620	1.8×10^{-10}
100	10	33	55.178	3.1×10^{-9}	7	0.1090	6.4×10^{-9}	14	0.1560	1.5×10^{-9}	11	0.1240	2.0×10^{-10}
200	10	36	1610.5	3.7×10^{-10}	27	0.7330	8.4×10^{-11}	28	0.7950	8.4×10^{-11}	51	1.3100	8.4×10^{-11}
300	10		—		7	0.4520	7.6×10^{-9}	14	0.8110	1.8×10^{-9}	11	0.5770	2.0×10^{-10}



Logarithm of the residual $\|R^k\|$ Number of iterations for $\beta \in [1, 100]$

Figure 1: One test of Example 4.3 with $n = 200$ and $p = 10$

is very robust since it is almost independent to the dimension and number of linear constraints of the problem.

To further illustrate the efficiency of the proposed ADMM-based algorithms, we consider a practical example in structural engineering [28, 32].

Example 4.4 *This is a PESDIEP arising in structural engineering with* $n = 1806$. *Let the matrix* $\widehat{C}$ *be BCSSTK14 (32630 nonzero entries), which comes from the set BCSSTRUC2 in the Harwell-Boeing collection[11]. Then we set*

$$C_o := \widehat{C} + R_C . * \widehat{C},$$

where R_C *is a real symmetric matrix whose entries are generated pseudo-randomly and they are uniformly distributed within* $[-0.2, 0.2]$. *We choose* p *eigenpairs of* $\widehat{C}$ *as prescribed eigendata.* $\mathcal{I}_s$ *is the index set corresponding to the entries of* C_o *whose values are equal to zeros.*

For demonstration purpose, we apply the ADMM-based algorithms to Example 4.4, where the stopping criterion is given by

$$\texttt{RRes.} := \big(\|C^k X - X\Lambda\|^2 + \sum_{(i,j)\in\mathcal{I}_s} \big((C^k)_{ij} - (C_o)_{ij}\big)^2\big)^{1/2} / (\|C^k X\| + \|X\Lambda\|) \le 5.0 \times 10^{-5}.$$

The initial guess is as above.

Tables 5 gives the numerical results for Example 4.4. Table 5 show that the proposed ADMM-based algorithms converge to the desired relative accuracy with only a small number of iterations. Also, we observe that, for the large PESDIEP in Example 4.4, the major computing time is spent on the CG method for solving the linear equation (9). An interesting question is how to find a good preconditioner for (9), which can improve the performance of our algorithms.

Table 4: Numerical results for Example 4.3

$p = 30$, $\beta = 8$, and $\delta = \gamma = 1.0$									
	ADMM			MADMM			RADMM		
n	IT.	CT.	Res.	IT.	CT.	Res.	IT.	CT.	Res.
100	8	0.14	4.7×10^{-8}	13	0.19	2.4×10^{-7}	8	0.13	6.7×10^{-9}
200	8	0.28	4.0×10^{-8}	14	0.45	9.0×10^{-8}	16	0.52	2.7×10^{-10}
500	9	1.70	4.9×10^{-9}	14	2.59	7.8×10^{-8}	20	3.39	3.1×10^{-10}
1,000	9	6.55	4.8×10^{-9}	15	11.7	1.1×10^{-7}	19	13.3	3.8×10^{-10}
2,000	8	27.2	4.3×10^{-8}	14	48.0	7.6×10^{-8}	16	51.6	7.8×10^{-11}
3,000	8	76.5	4.2×10^{-8}	15	143.4	1.7×10^{-7}	16	145.7	1.4×10^{-11}
5,000	8	290.4	4.2×10^{-8}	13	473.6	2.3×10^{-7}	13	459.5	3.3×10^{-10}
$n = 2000$, $\beta = 8$, and $\delta = \gamma = 1.0$									
	ADMM			MADMM			RADMM		
p	IT.	CT.	Res.	IT.	CT.	Res.	IT.	CT.	Res.
10	8	27.0	2.4×10^{-8}	15	51.0	8.9×10^{-8}	15	49.9	2.5×10^{-10}
20	8	27.5	3.4×10^{-8}	14	48.8	7.2×10^{-8}	16	52.0	2.2×10^{-10}
50	8	28.5	5.6×10^{-8}	14	50.8	1.1×10^{-7}	17	58.6	4.9×10^{-10}
80	8	31.4	6.9×10^{-8}	13	53.9	3.5×10^{-7}	17	66.1	5.0×10^{-10}
100	8	32.8	7.8×10^{-8}	13	53.9	4.0×10^{-7}	15	57.0	6.7×10^{-10}
150	8	37.6	9.4×10^{-8}	13	59.6	4.5×10^{-7}	17	74.3	6.6×10^{-10}
200	8	43.2	1.1×10^{-7}	13	70.4	5.4×10^{-7}	15	76.9	8.9×10^{-10}

Table 5: Numerical results for Example 4.4

$\beta = 100$, and $\delta = \gamma = 1.0$									
	ADMM			MADMM			RADMM		
p	IT.	CT.	RRes.	IT.	CT.	RRes.	IT.	CT.	RRes.
4	6	424.5	3.9×10^{-5}	10	671.3	2.4×10^{-5}	3	243.4	2.5×10^{-5}
8	5	364.7	2.6×10^{-5}	7	490.0	3.4×10^{-5}	3	245.3	4.3×10^{-5}

Acknowledgments

We would like to thank the editor and the anonymous referees for their helpful and valuable comments. The research of Z.-J. Bai was partially supported by the National Natural Science

Foundation of China grant 11271308.

References

[1] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, *Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results*, SIAM J. Optim., 8 (1998), pp. 746–768.

[2] A. Antoniou and W.-S. Lu, *Practical Optimization, Algorithms and Engineering Applications*, Springer, New York, 2007.

[3] Z. J. Bai, M. X. Chen, and X. M. Yuan, *Applications of the alternating direction method of multipliers to the semidefinite inverse quadratic eigenvalue problem with partial eigenstructure*, Inverse Problems, 29 (2013), 075011.

[4] Z. J. Bai, D. L. Chu, and D. F. Sun, *A dual optimization approach of inverse quadratic eigenvalue problems with partial eigenstructure*, SIAM J. Sci. Comput., 29 (2007), pp. 2531–2561.

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

[6] R. B. Bapat and T. E. S. Raghavan, *Nonnegative Matrices and Applications*, Encyclopedia of Mathematics and its Applications 64, Cambridge University Press, 1997.

[7] M. Baruch, *Optimization procedure to correct stiffness and flexibility matrices using vibration data*, AIAA J., 16 (1978), pp. 1208–1210.

[8] A. Berman and E.J. Nagy, *Improvement of large analytical model using test data*, AIAA J., 21 (1983), pp. 1168–1173.

[9] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics in Applied Mathematics, 9, SIAM, Philadelphia, 1994.

[10] D. P. Bertsekas, *On the method of multipliers for convex programming*, I.E.E.E. Trans. AC-20 (1982), pp. 385–388.

[11] R. Boisvert, R. Pozo, K. Remington,, B. Miller, and R. Lipman, *Matrix Market*, National Institute of Standards and Technology, Gaithersburg, 2007.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Machine Learning, 3, 2010, pp. 1–122.

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[14] X. J. Cai, G. Y. Gu, B. S. He, and X. M. Yuan, *A proximal point algorithm revisit on the alternating direction method of multipliers*, Sci. China Ser. A, 56 (2013), pp. 2179–2186.

[15] 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.

[16] M. T. Chu, *Inverse eigenvalue problems*, SIAM Rev., 40 (1998), pp. 1–39.

[17] 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.

[18] 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.

[19] M. T. Chu and G. H. Golub, *Structured inverse eigenvalue problems*, Acta Numer., 11 (2002), pp. 1–71.

[20] M. T. Chu and G. H. Golub, *Inverse Eigenvalue Problems: Theory, Algorithms, and Applications*, Oxford University Press, Oxford, 2005.

[21] 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.

[22] R. W. Clough and J. Penzien, *Dynamics of Structures*, McGraw-Hill, Dusseldorf, 1975.

[23] 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.

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

[25] J. Eckstein and M. C. Ferris, *Operator-splitting methods for monotone affine variational inequalities, with a parallel application to optimal control*, INFORMS Journal on Computing, 10 (1998), pp. 218–235.

[26] P. D. Egleston, T. D. Lenker, and S. K. Narayan, *The nonnegative inverse eigenvalue problem*, Linear Algebra Appl., 379 (2004), pp. 475–490.

[27] M. I. Friswell, D. J. Inman, and D. F. Pilkeyand, *The direct updating of damping and stiffness matrices*, AIAA J., 36 (1998), pp. 491–493.

[28] M. I. Friswell and J. E. Mottershead, *Finite Element Model Updating in Structural Dynamics*, Kluwer Academic Publishers, 1995.

[29] 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.

[30] 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.

[31] E. Ghadimi, A. Teixeira, I. Shames,and M. Johansson, *Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems*, arXiv:1306.2454v2, 2014.

[32] G. M. L. Gladwell, *Inverse Problems in Vibration*, Kluwer Academic Publishers, 2004.

[33] 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.

[34] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, SIAM Studies in Applied Mathematics, Philadelphia, PA, 1989.

[35] E. G. Gol'shtein and N. V. Tret'yakov, *Modified Lagrangian in convex programming and their generalizations*, Math. Program. Studies, 10, 1979, pp. 86–97.

[36] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd edition, Johns Hopkins University Press, Baltimore and London, 1996.

[37] 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.

[38] 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.

[39] M. R. Hestenes, *Multiplier and gradient methods*, J. Optim. Theory Appl., 4 (1969), pp. 303–320.

[40] N. J. Higham, *Computing a nearest symmetric positive semidefinite matrix*, Linear Algebra Appl., 103 (1988), pp. 103–118.

[41] J. B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, 1993.

[42] Y. C. Kuo, W. W. Lin, and S. F. Xu, *New methods for finite element model updating problems*, AIAA J. 44 (2006), pp. 1310–1316.

[43] M. M. Lin, B. Dong, and M. T. Chu, *Semi-definite programming techniques for structured quadratic inverse eigenvalue problems*, Numer Algor., 53 (2010), pp. 419–437.

[44] H. Minc, *Nonnegative Matrices*, John Wiley&Sons, New York, 1988.

[45] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 1999.

[46] R. Orsi, *Numerical methods for solving inverse eigenvalue problems for nonnegative matrices*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 190–212.

[47] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, In Optimization edited by R. Fletcher, pp. 283-298, Academic Press, New York, 1969.

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

[49] R. T. Rockafellar, *Conjugate Duality and Optimization*, CBMS-NSF Regional Conf. Ser. in Appl. Math. 16, SIAM, Philadelphia, 1974.

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

[51] E. Senata, *Non-negative Matrices and Markov Chains*, 2nd rev. ed., Springer-Verlag, New York, 2006.

[52] J. G. Sun, *Backward perturbation analysis of certain characteristic subspaces*, Numer. Math., 65 (1993), 357–382.

[53] M. J. Todd, *Semidefinite optimization*, Acta Numer. 10 (2001), pp. 515–560.

[54] 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/ mattohkc/sdpt3.html

[55] S. F. Xu, *An Introduction to Inverse Algebraic Eigenvalue Problems*, Peking University Press and Vieweg Publishing, 1998.

[56] C. H. Ye and X. M. Yuan, *A descent method for structured monotone variational inequalities*, Optim. Methods Softw., 22 (2007), pp. 329–338.

[57] Z. Zhao, Z. J. Bai, and G. Z. Chen, *On the alternating direction method of multipliers for nonnegative inverse eigenvalue problems with partial eigendata*, J. Comput. Appl. Math., 239 (2013), pp. 114–134.