ON THE LEAST SQUARES SOLUTION OF INVERSE EIGENVALUE PROBLEMS

XUZHOU CHEN* AND MOODY T. CHU†

Abstract. An inverse eigenvalue problem, where a matrix is to be constructed from some or all of its eigenvalues, may not have a real-valued solution at all. An approximate solution in the sense of least squares is sometimes desirable. Two types of least squares problems are formulated and explored in this paper. In spite of their different appearance, the two problems are shown to be equivalent. Thus one new numerical method, modified from the conventional alternating projection method, is proposed. The method converges linearly and globally, and can be used to generate good starting values for other faster but more expensive and locally convergent methods. The idea can be applied to multiplicative inverse eigenvalue problems for the purpose of preconditioning. Numerical examples are presented.

Key words. Inverse eigenvalue problem, Least Squares, Lift and Projection.

AMS(MOS) subject classifications. 65F15, 65H15.

1. Introduction. Inverse eigenvalue problems are of great importance to many applications. Discussions on various aspects of the existence theory as well as numerical techniques have been extensive. See, for example, [1, 2, 5, 6, 11, 12, 13]. In this paper we shall consider inverse eigenvalue problems of the following form [13]:

(IEP). Given real, symmetric $n \times n$ matrices $A_0, A_1, \ldots, A_n$ and real numbers $\lambda_n^* \leq \ldots \leq \lambda_n^*$, find $d \in R^n$ such that eigenvalues $\lambda_1(d) \leq \ldots \leq \lambda_n(d)$ of

$$A(d) := A_0 + \sum_{k=1}^{n} d_k A_k \tag{1}$$

satisfy $\lambda_i(d) = \lambda_i^*$ for $i = 1, 2, \ldots, n$.

Our interest in the IEP is motivated by the problem of optimally preconditioning a given matrix by a diagonal matrix [15, 16], which is closely related to the following multiplicative inverse eigenvalue problem:

(MIEP). Given a real, symmetric $n \times n$ matrix A and real numbers $\lambda_1^* \leq \ldots \leq \lambda_n^*$, find $d \in R^n$ such that the eigenvalues $\lambda_1(d) \leq \ldots \leq \lambda_n(d)$ of the matrix

$$A(d) := DA, \tag{2}$$

where $D := \mathrm{diag}\{d_1, \ldots, d_n\}$, satisfy $\lambda_i(d) = \lambda_i^*$ for $i = 1, 2, \ldots, n$.

The symmetry in the description of problems above and in the discussion hereafter is not essential. The work can be generalized to more general matrices where complex-valued eigenvalues may be involved. In that case, the only extra burden is the task of pairing off the eigenvalues. This process will become clear in the sequel.

* Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205

† Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205. This research was supported in part by National Science Foundation under grant DMS-9123448.

The MIEP may be written in the form of the IEP by selecting $A_0 = 0$ and $A_k = e_k a_k^T$ for $k = 1, \ldots, n$ where a_k^T is the kth row of A and e_k denotes the kth standard basis in R^n. The matrices A_k in this setting, however, are not symmetric. If, in addition, A is positive definite, then the matrix DA is similar to $L^T DL$ where L is the Cholesky factor of $A = LL^T$. We may then convert the MIEP to an IEP by using symmetric matrices $A_0 = 0$ and $A_k = L^T E_k L$ with $E_k := \mathrm{diag}(e_k)$.

Friedland [11] has showed that the MIEP (even without the symmetry) is always solvable over the complex field and that the number of solutions is at most $n!$, provided all the principal minors of A are distinct from zero. When restricted to the real field, however, no theory thus far can generally guarantee the existence of a solution for either the IEP or the MIEP. This non-existence of a solution can easily be seen by examples. This paper thus concerns the solution of the IEP or the MIEP in the least squares sense.

We begin in the next section by considering the 2×2 case to illustrate the non-existence of a solution for the MIEP. This example also demonstrates an interesting connection between a least squares MIEP and an optimal preconditioner discussed in [15, 16]. In §3 we describe the least squares formulation of an inverse eigenvalue problem. Our first formulation is simply to generalize the conventional IEP. We relax the problem to the extent that the set of eigenvalues to be matched is not necessarily the entire spectrum and the number of free parameters in d is not necessarily the same as the dimension of the underlying matrices. It will become clear that in order to solve such a least squares problem, a combinatorics problem naturally arises. This formulation also inherits the troublesome non-smoothness of eigenvalues. Our second formulation is to cast the least squares approximation problem as a problem of finding the distance between two interesting geometric objects. One of our contributions in this paper is to show that, in spite of their different appearance, these two formulations are equivalent. The second formulation has the advantages that no differentiation of eigenvalues is required and that a variation of the so-called alternating projection method can be applied. We call our new method a *lift* and a *projection* (LP). More details on how the proximity map should be defined are furnished in §4. The conventional Newton method is briefly reviewed in §5. Together with the LP method, we propose a hybrid method to circumvent some of the difficulties associated with the Newton method. Finally, we compare the performance of the proposed numerical methods in §6.

2. An Example of MIEP. The case for the MIEP when A is a 2×2 matrix is simple yet illuminating. We work out the analysis in this section.

Given $A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$ and $\lambda^* := (\lambda_1^*, \lambda_2^*)$, the characteristic polynomial of $A(d) = DA$ for $d = [d_1, d_2]^T \in R^2$ is

$$c_A(x) = x^2 - (ad_1 + cd_2)x + (ac - b^2)d_1 d_2. \tag{3}$$

The MIEP is equivalent to selecting d_1 and d_2 so that λ_1^* and λ_2^* are the roots of (3). Toward this, we solve d_1 and d_2 in terms of λ_1^* and λ_2^*. It is not difficult to see that a necessary and sufficient condition for the MIEP to have a real solution is

$$(a^2c^2 - 2acb^2 + b^4)\lambda_1^{*2} - 2(a^2c^2 - b^4)\lambda_1^*\lambda_2^* + (a^2c^2 - 2acb^2 + b^4)\lambda_2^{*2} \geq 0. \tag{4}$$

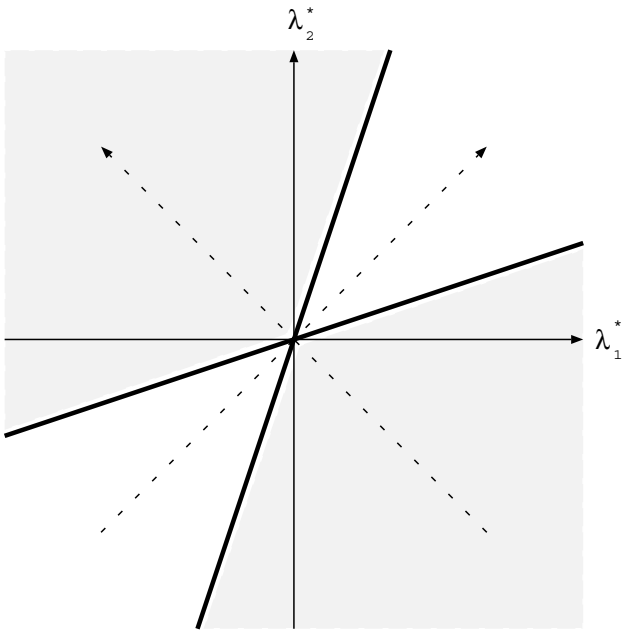

FIG. 1. *Region of* $(\lambda_1^*, \lambda_2^*)$ *on which the MIEP is solvable.*

After rotating the $(\lambda_1^*, \lambda_2^*)$-axes counterclockwise by an angle of $\pi/4$, the condition (4) is reduced to

$$(b^2 - ac)\left(b^2\hat{\lambda}_1^2 - ac\hat{\lambda}_2^2\right) \geq 0. \tag{5}$$

Given any a, b and c, the inequality (5) provides a easy way to check whether the MIEP with a given pair of eigenvalues λ_1^*, λ_2^* is solvable.

When A is symmetric and positive definite, a typical domain of feasible λ_1^* and λ_2^* occupies a shaded region like that in Figure 1. If we assume that $\lambda_1^* \leq \lambda_2^*$, as in the discussion, then only the shaded area above the main diagonal line is needed. It is clear from Figure 1 that there is a good possibility that an arbitrarily given pair of eigenvalues $(\lambda_1^*, \lambda_2^*)$ fail to be feasible for the MIEP.

Suppose now we want to consider a least squares problem of MIEP in the sense of finding $d^* \in R^2$ so that $\|\lambda(d^*) - \lambda^*\|_2 = \min_{d\in R^2} \|\lambda(d) - \lambda^*\|$. Consider the case when the prescribed eigenpair $\lambda^* = (\lambda_1^*, \lambda_2^*)$ is outside the shaded region. Figure 1 clearly suggests that the corresponding $\lambda(d^*)$ of the optimal d^* should be exactly the point on the borderline that is closest to the point λ^*.

Another interesting observation is that if the diagonal matrix D is required to be positive definite, then the condition number $\kappa(DA)$ referred to in [15] is precisely the ratio $\frac{\lambda_2(d)}{\lambda_1(d)} > 0$. The eigenvalues of DA with $D := \mathrm{diag}\{1/a, 1/c\}$, the optimal positive definite diagonal preconditioner [10, 21] in the sense of [15, 16], are precisely those lying on the upper borderline in the first quadrant of Figure 1.

3. The Least Squares IEP. Thus far, most algorithms discussed in the literature for solving the IEP have assumed that a (real) solution somehow is known to exist a prior. This piece of information usually is not available in practice. Furthermore, as we have seen in §2, a real solution may not exist at all. In this paper we want to study an approximate solution to the IEP in the least squares sense. A natural reformulation of the the IEP leads to the following problem:

(LSIEP1). Given real, symmetric $n \times n$ matrices $A_0, A_1, \ldots, A_\ell$ and real numbers $\lambda_1^* \leq \ldots \leq \lambda_m^*$ $(m \leq n)$, find $d \in R^\ell$ and a permutation $\sigma = \{\sigma_1, ..., \sigma_m\}$ with $1 \leq \sigma_1 < \cdots < \sigma_m \leq n$ such that the function

$$F(d, \sigma) := \frac{1}{2} \sum_{i=1}^{m} (\lambda_{\sigma_i}(d) - \lambda_i^*)^2, \tag{6}$$

where $\lambda_i(d)$, $i = 1, ..., n$, are eigenvalues of the matrix

$$A(d) = A_0 + \sum_{i=1}^{\ell} d_i A_i, \tag{7}$$

is minimized.

It is noteworthy that the set of prescribed eigenvalues has cardinality m which might be less than n. Consequently, associated with the LSIEP1 is a combinatorics problem

$$\min_{1 \leq \sigma_1 < \ldots < \sigma_m \leq n} \sum_{i=1}^{m} (\lambda_{\sigma_i}(d) - \lambda_i^*)^2, \tag{8}$$

that looks for the closest match between a subset of spectrum of $A(d)$ and the prescribed eigenvalues. Note also that the number of available parameters for adjusting the matrix $A(d)$ is ℓ which could be greater than or less than n.

Clearly, the LSIEP1 is a non-linear least squares problem. It is important to observe that the function $F(d, \sigma)$ is not differentiable in d when the permutation σ is changed. Still, many standard optimization techniques might be applicable. In particular, the Newton method for solving $\nabla F = 0$ is possible. At a substantial cost, the Hessian matrix of F can be explicitly formulated (See (28)). We shall see that the Hessian matrix still exists even if multiple eigenvalues of $A(d)$ are present. (The unbounded terms in (28) will eventually be canceled out at coalescent eigenvalues.) As the separation of eigenvalues decreases, however, the Hessian matrix becomes increasingly ill-conditioned.

There is an alternative way to propose a least squares problem. Specifically, let $\mathcal{O}(n)$ and D^{n-m} denote, respectively, the sets of all orthogonal matrices in $R^{n \times n}$ and all $(n-m) \times (n-m)$ real diagonal matrices. For a given $\Lambda_m^* := \mathrm{diag}\{\lambda_1^*, \ldots, \lambda_m^*\}$, consider the subset

$$\Gamma := \left\{ Q \mathrm{diag}(\Lambda_m^*, \Lambda) Q^T \mid Q \in \mathcal{O}(n), \Lambda \in D^{n-m} \right\} \tag{9}$$

and the affine subspace

$$\mathcal{A} := \{A(d) | d \in R^\ell\} \tag{10}$$

with $A(d)$ defined by (7). Since Γ contains all symmetric matrices in $R^{n\times n}$ with $\lambda_1^*, ..., \lambda_m^*$ as part of the spectrum, finding the shortest distance between $\mathcal{A}$ and Γ would be another meaningful least squares approximation. We formulate the problem as follows:

(LSIEP2). Find $d \in R^{\ell}$, $Q \in \mathcal{O}(n)$, and $\Lambda \in D^{n-m}$ such that the function

$$G(d, Q, \Lambda) := \frac{1}{2}\|A(d) - Q\,\mathrm{diag}\,(\Lambda_m^*, \Lambda)\,Q^T\|_F^2, \tag{11}$$

where $\|\cdot\|_F$ denotes the Frobenius matrix norm, is minimized.

At the first glance, the LSIEP1 and the LSIEP2 appear to be very different. In particular, it appears that no permutation of eigenvalues is involved in the LSIEP2. However, a process of implicit sorting is indeed happening inside the LSIEP2 as we will show below that the LSIEP1 and LSIEP2 are equivalent.

We first show the following theorem for arbitrary σ and d.

THEOREM 3.1. *For any d and σ, define $\Lambda_\sigma(d) := diag\{\lambda_{\sigma_1}(d), \ldots, \lambda_{\sigma_m}(d)\}$. Let $\overline{\sigma}$ denote the complement of σ over the set $\{1, \ldots, n\}$. Define $\Lambda_\sharp := \Lambda_{\overline{\sigma}}(d)$. Suppose columns of $Q_\sharp$ are orthonormal eigenvectors of $A(d)$ arranged in such a way that*

$$Q_\sharp^T A(d) Q_\sharp = diag(\Lambda_\sigma(d), \Lambda_{\overline{\sigma}}(d)). \tag{12}$$

Then $G(d, Q_\sharp, \Lambda_\sharp) = F(d, \sigma)$.

Proof. From (12) we have

$$\begin{aligned} G(d, Q_\sharp, \Lambda_\sharp) &= \frac{1}{2}\|A(d) - Q_\sharp \mathrm{diag}(\Lambda_m^*, \Lambda_\sharp) Q_\sharp^T\|_F^2 \\ &= \frac{1}{2}\|Q_\sharp^T A(d) Q_\sharp - \mathrm{diag}(\Lambda_m^*, \Lambda_\sharp)\|_F^2 \\ &= \frac{1}{2}\|\mathrm{diag}(\Lambda_\sigma(d) - \Lambda_m^*, \Lambda_{\overline{\sigma}}(d) - \Lambda_\sharp)\|_F^2 \\ &= \frac{1}{2}\|\Lambda_\sigma(d) - \Lambda_m^*\|_F^2. \end{aligned}$$

The assertion therefore follows. □

We now establish the relationship between the LSIEP1 and the LSIEP2.

THEOREM 3.2. *Suppose (d^*, σ^*) and (d_+, Q_+, Λ_+) are the global minimizers of the LSIEP1 and the LSIEP2, respectively. Then*

1. *The permutation σ^* solves (8) with $d = d^*$.*
2. *$d^* = d_+$.*
3. *The columns of Q_+ are orthonormal eigenvectors of $A(d^*)$ arranged in such a way that $Q_+^T A(d^*) Q_+ = diag\,(\Lambda_{\sigma^*}(d^*), \Lambda_{\overline{\sigma^*}}(d^*))$.*
4. *$\Lambda_+ = \Lambda_{\overline{\sigma^*}}(d^*)$.*
5. *$F(d^*, \sigma^*) = G(d_+, Q_+, \Lambda_+)$.*

Proof. Since

$$F(d^*, \sigma^*) = \min_d \min_\sigma F(d, \sigma) = \min_\sigma F(d^*, \sigma),$$

it is obvious that σ^* must solve (8) with $d = d^*$.

Let $Q(d^*)$ denote the orthogonal matrix of which columns are eigenvectors of $A(d^*)$ arranged in such a way that $Q(d^*)^T A(d^*)Q(d^*) = \mathrm{diag}\,(\Lambda_{\sigma^*}(d^*), \Lambda_{\overline{\sigma^*}}(d^*))$. It follows from Theorem 3.1 that

$$G(d_+, Q_+, \Lambda_+) \le G(d^*, Q(d^*), \Lambda_{\overline{\sigma^*}}(d^*)) = F(d^*, \sigma^*). \tag{13}$$

On the other hand, we have

$$\begin{aligned} G(d_+, Q_+, \Lambda_+) &= \min_{d,\Lambda} \min_{Q} \frac{1}{2}\|A(d) - Q\mathrm{diag}(\Lambda_m^*, \Lambda)Q^T\|_F^2 \\ &= \min_{d,\Lambda} \frac{1}{2}\|A(d) - Q(d)\mathrm{diag}\,(\Lambda_m^*, \Lambda)\, Q(d)^T\|_F^2 \qquad (14) \\ &= \min_{d,\Lambda} \frac{1}{2}\|\mathrm{diag}\,(\Lambda_\sigma(d) - \Lambda_m^*, \Lambda_{\overline{\sigma}}(d) - \Lambda)\,\|_F^2 \qquad (15) \\ &\ge \min_{d} \frac{1}{2}\|\Lambda_\sigma(d) - \Lambda_m^*\|_F^2. \qquad (16) \end{aligned}$$

The equality (14) follows from the Wielandt-Hoffman theorem [18, Theorem 6.3.5] where columns of the orthogonal matrix $Q(d)$ are eigenvectors of $A(d)$ arranged in such a way that elements in the diagonal matrix $Q(d)^T A(d)Q(d)$ are in the same ordering as those of $\mathrm{diag}(\Lambda_m^*, \Lambda)$. The permutation σ in (15) simply reflects such an rearrangement of eigenvalues of $A(d)$. (This is the implicit sorting referred to above.) Together with (13), we find that the equality in (16) holds if and only if $d = d^* = d_+$, $\sigma = \sigma^*$ and $\Lambda_+ = \Lambda_{\overline{\sigma^*}}(d^*)$. □

Theorem 3.2 warrants that the LSIEP1 can be solved by dealing with the LSIEP2 only. The LSIEP2 also provides a geometric interpretation of the LSIEP1. We think such a connection is quite intriguing.

4. Lift and Projection. Taking advantage of the equivalence between the LSIEP1 and the LSIEP2, we now propose an iterative method that alternates points between Γ and $\mathcal{A}$. The idea is essentially the same as the so-called alternating projection method for convex sets [4, 9, 17], except that one of our sets, namely Γ, is not convex. Our contribution here is to show that the proximity maps can still be well defined. We call our method a *lift* and a *projection* (LP). A similar idea of the lift and projection has been used in [7]. For the LSIEP2, an extra combinatorics problem is involved. The cost of computation is one spectral decomposition plus one sorting per lift and two triangular linear system solving per projection.

Before introducing our method, we stress that no differentiation of eigenvalues is involved in the LP method. We also note that the LP method converges slowly but globally. We may, therefore, take advantage of a hybrid method by first applying the LP method to attain a low order of accuracy and then switching to a faster but locally convergent method for a high order of accuracy. The LP method may also be used at a step where $A(d)$ appears to have multiple or nearly coalescent eigenvalues. This hybrid method is discussed in §5.

For each given $d^{(k)} \in R^\ell$, the LP method iterates the following two steps:

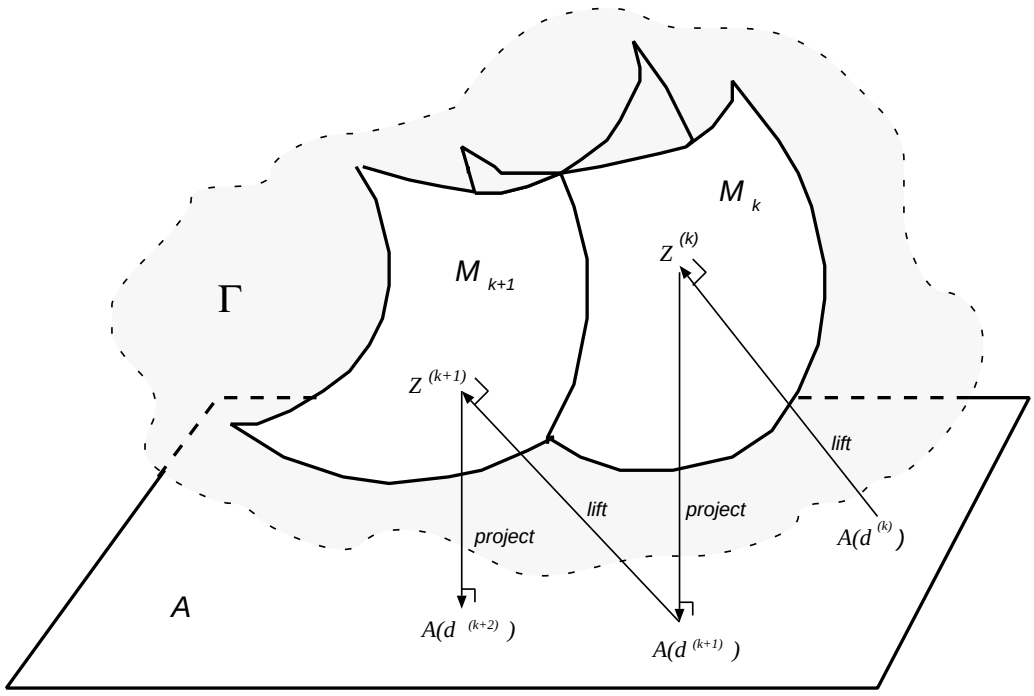

FIG. 2. *Geometric sketch of lift and projection.*

1. **(Lift)** Find the point $Z^{(k)} \in \Gamma$ such that $\mathrm{dist}(A(d^{(k)}), Z^{(k)}) = \mathrm{dist}(A(d^{(k)}), \Gamma)$. We call $Z^{(k)}$ a *lift* of $A(d^{(k)})$ onto Γ.
2. **(Projection)** Find the point $d^{(k+1)} \in R^{\ell}$ such that $\mathrm{dist}(A(d^{(k+1)}), Z^{(k)}) = \mathrm{dist}(Z^{(k)}, \mathcal{A})$. The point $A(d^{(k+1)}) \in \mathcal{A}$ is called a *projection* of $Z^{(k)}$ onto $\mathcal{A}$.

A schematic diagram of the iteration is illustrated in Figure 2 although the topology of Γ is much more complicated.

The projection of $Z^{(k)} \in R^{n\times n}$ onto $\mathcal{A}$ is easy to do. The vector $d^{(k+1)}$ is the solution of the linear system

$$\sum_{i=1}^{\ell} \langle A_i, A_j \rangle d_i^{(k+1)} = \langle Z^{(k)} - A_0, A_j \rangle, \ , j = 1, \ldots, \ell. \tag{17}$$

where $\langle A, B \rangle := \mathrm{trace}(A^T B)$ is the Frobenius inner product for matrix A and B. Note that the coefficient matrix in (17) is independent of k. So the left-hand side of (17) needs to be factorized only once.

The lift step is not as easy because elements in Γ involve $n - m$ undetermined eigenvalues. Motivated by the proof of Theorem 3.2, however, the step can proceed as follows: Suppose $A(d^{(k)}) = Q(d^{(k)})\mathrm{diag}(\Lambda_{\sigma^{(k)}}(d^{(k)}), \Lambda_{\overline{\sigma^{(k)}}}(d^{(k)}))\ Q(d^{(k)})^T$ is the spectral decomposition of $A(d^{(k)})$ where $\sigma^{(k)} = \sigma^{(k)}(d^{(k)})$ is the permutation that solves the combinatorics problem (8) with $d = d^{(k)}$ and $Q(d^{(k)})$ is the corresponding orthogonal matrix of eigenvectors. Then the shortest distance between $A(d^{(k)})$ and Γ is attained at the point [3, 5]

$$Z^{(k)} := Q(d^{(k)})\mathrm{diag}\left(\Lambda_m^*, \Lambda_{\overline{\sigma^{(k)}}}(d^{(k)})\right) Q(d^{(k)})^T. \tag{18}$$

In other words, in order to find the shortest distance from $A(d^{(k)})$ to Γ, it suffices to find the shortest distance from $A(d^{(k)})$ to a *substructure* $\mathcal{M}_k$ of Γ, where the substructure

$$\mathcal{M}_k := \left\{ Q\mathrm{diag}\left(\Lambda_M^*, \Lambda_{\overline{\sigma^{(k)}}}(d^{(k)})\right) Q^T | Q \in \mathcal{O}(n) \right\} \tag{19}$$

has a much simpler topology than Γ because the diagonal elements are fixed. (See Figure 2.) The price we pay for this roundabout is to solve (8) per step. It is worth noting that when the iterates are reaching convergence the permutations $\sigma^{(k)}$ should become stablized.

THEOREM 4.1. *The LP method is a descent method in the sense that*

$$\|A(d^{(k+1)}) - Z^{(k+1)}\|_F^2 \leq \|A(d^{(k+1)}) - Z^{(k)}\|_F^2 \leq \|A(d^{(k)}) - Z^{(k)}\|_F^2. \tag{20}$$

Thus the LP method generates a sequence of matrix pairs $\{(Z^{(k)}, A(d^{(k)}))\}$ *that converges to a (local) stationary point for the problem of minimizing (11).*

Proof. The second inequality is obvious because $A(d^{(k+1)})$ is the projection point of $Z^{(k)}$ onto $\mathcal{A}$. The first inequality follows from the Wielandt-Hoffman theorem. More precisely, we observe that

$$\begin{aligned} \|A(d^{(k+1)}) - Z^{(k)}\|_F^2 &= \|A(d^{(k+1)}) - Q(d^{(k)})\mathrm{diag}\left(\Lambda_m^*, \Lambda_{\overline{\sigma^{(k)}}}(d^{(k)})\right) Q(d^{(k)})^T\|_F^2 \\ &\geq \|A(d^{(k+1)}) - Q(d^{(k+1)})\mathrm{diag}\left(\Lambda_m^*, \Lambda_{\overline{\sigma^{(k+1)}}}(d^{(k+1)})\right) Q(d^{(k+1)})^T\|_F^2 \\ &= \|A(d^{(k+1)}) - Z^{(k+1)}\|_F^2 \end{aligned}$$

where the inequality follows from the definition of $Q(d^{(k+1)})$. □

The alternating projection method [4, 9, 17] has been used to tackle the IEP by first reformulating the problem so as to create convex constraints. In contrast, our application to the LSIEP2 is quite direct. Our approach is interesting and remarkable in two aspects. One is that, even though the set Γ is complicated, we can simply work with one of its substructures. The other is that, even though the substructure $\mathcal{M}_k$ is not convex, the so called proximity map can still be formulated by using the Wielandt-Hoffman theorem.

5. The Newton Method. We have indicated earlier that the function $F(d, \sigma)$ is not smooth in σ. Since the permutation σ is only a discrete variable, such a non-smoothness does not necessarily preclude a proper application of classical least squares techniques to the LSIEP1. In this section we briefly describe how this can be done by the Newton method.

For any $A(d)$ defined by (7), let $(q_i(d), \lambda_i(d))$ denote an eigenpair of $A(d)$, i.e.,

$$A(d)q_i(d) = \lambda_i(d)q_i(d),\ i = 1, 2, ..., n. \tag{21}$$

Having found an optimal permutation $\sigma = \{\sigma_1, ..., \sigma_m\}$ for problem (8), we categorize (21) into two groups:

$$q_{\sigma_i}(d)^T A(d) q_{\sigma_i}(d) = \lambda_{\sigma_i}(d),\ i = 1, 2, ..., m; \tag{22}$$

$$q_{\overline{\sigma}_j}(d)^T A(d) q_{\overline{\sigma}_j}(d) = \lambda_{\overline{\sigma}_j}(d),\ j = 1, 2, ..., n-m. \tag{23}$$

The notion here is to assume σ is fixed and to iterate the variable d based on the relationship (22) since only elements in $\lambda_\sigma(d) = [\lambda_{\sigma_1}, \ldots, \lambda_{\sigma_m}]^T$ are involved in (7). It is important to mention that in our numerical experiments, we do have seen that the permutation σ are changing at the initial stage of iteration.

Upon differentiating both sides of (22), we obtain the Jacobian matrix $J(d) = [J_{ik}(d)]$ of $\lambda_\sigma(d)$ where

$$J_{ik}(d) := \frac{\partial \lambda_{\sigma_i}(d)}{\partial d_k} = q_{\sigma_i}(d)^T A_k q_{\sigma_i}(d),\ i = 1, ..., m;\ k = 1, ..., \ell. \tag{24}$$

It is not difficult to see that the first and second derivatives of $F(d, \sigma)$ are given, respectively, by

$$\nabla F(d, \sigma) = J(d)^T(\lambda_\sigma(d) - \lambda^*), \tag{25}$$

and

$$\nabla^2 F(d, \sigma) = J(d)^T J(d) + S(d), \tag{26}$$

where

$$S(d) := \sum_{i=1}^{m} (\lambda_{\sigma_i}(d) - \lambda_i^*)\nabla^2(\lambda_{\sigma_i}(d) - \lambda_i^*). \tag{27}$$

The Hessian matrix in (27) can be calculated, for example, by using the formula [19]

$$\frac{\partial^2(\lambda_i(d))}{\partial d_k \partial d_j} = 2 \sum_{\substack{t=1 \\ \lambda_t \neq \lambda_i}}^{n} \frac{[q_t(d)^T A_k q_i(d)][q_t(d)^T A_j q_i(d)]}{\lambda_i(d) - \lambda_t(d)} \tag{28}$$

Note that the summation in (28) is over those t for which $\lambda_t \neq \lambda_i$, so the formula is valid even if λ_i is repeated. One step of the conventional Newton method applied to LSIEP1 amounts to solving the linear system

$$(J(d^{(k)})^T J(d^{(k)}) + S(d^{(k)}))\Delta d^{(k)} = -J(d^{(k)})^T(\lambda_\sigma(d^{(k)}) - \lambda^*). \tag{29}$$

and then advancing to $d^{(k+1)} := d^{(k)} + \Delta d^{(k)}$.

The Newton method, especially in the forming of $S(d)$, is very expensive. A possible strategy for remedying this situation is to employ some kinds of hybrid methods. For example, we could use the LP method in the initial stage to reach convergence at a relatively low order of accuracy. We then switch to the Newton method for achieving high order of accuracy. The approach also has the advantage that the permutation σ might get stablized before the Newton method is called. More precisely, we propose the following hybrid method:

ALGORITHM 5.1. LP-Newton Method

1. *Choose an arbitrary starting vector* $d^{(0)} \in R^\ell$.
2. *For* $k = 0, 1, 2, \ldots$, *do the LP iteration as follows:*
   (a) *Compute the spectrum decomposition* $Q(d^{(k)})$ *and* $\Lambda(d^{(k)})$.

(b) Find $\sigma^{(k)}$ that solves (8) with $d = d^{(k)}$.
(c) Form $Z^{(k)}$ according to (18).
(d) Compute $d^{(k+1)}$ from (17).
(e) Stop if $\|d^{(k+1)} - d^{(k)}\| < \epsilon_1$.

3. Set $d^{(0)} :=$ the limit point of the LP iteration.

4. For $k = 0, 1, 2, \ldots$, do the Newton iteration as follows:
(a) Generate the Jacobian matrix (24) and
(b) Solve the linear equation (29) for $\Delta d^{(k)}$.
(c) $d^{(k+1)} := d^{(k)} + \Delta d^{(k)}$.
(d) Stop if $\|d^{(k+1)} - d^{(k)}\| < \epsilon_2$.

Choosing ϵ_1 small enough will ensure the global convergence of this method.

6. Numerical Experiment. In this section we present some test results of our methods. We understand that there are many other algorithms for solving non-linear least squares problems. Some locally convergent methods with cost reduction in mind include, for example, the Gauss-Newton method that does not compute $S(d)$ and the Shamanskii method that does not evaluate the Hessian so frequently. We choose to compare the three methods discussed in this paper — the LP method, the LP-Newton method and the Newton method.

The experiment was carried out by MATLAB on a DECstation 5000/200. Initial values $d^{(0)}$ were generated randomly. To assess the efficiency, we carefully measured the CPU time for each test. Numerical results indicate that the LP method usually approached a stationary point quickly at the first few steps. The improvement then slowed down. This is a common phenomenon of linear convergence. In contrast, the Newton method converged slowly at the initial stage, but eventually the rates were picked up and became quadratic. We also have observed cases where the Newton method failed to converge. All of these observations seem to suggest that a hybrid method should be more suitable for the least squares inverse eigenvalue problems.

The associated combinatorics problem (8) was converted into a linear sum assignment problem. That is, for each given d, we first create the cost matrix $C = [c_{ij}]$ where

$$c_{ij} := \begin{cases} |\lambda_i(d) - \lambda_j^*|, & \text{if } 1 \le j \le m, \\ 0, & \text{otherwise.} \end{cases}$$

We then apply an existing algorithm *LSAPR* [20] to find a permutation ϕ^* that solves

$$\min_{\phi \in S_n} \sum_{i=1}^{n} c_{i,\phi(i)} \tag{30}$$

where S_n is the symmetric group of all permutations of $\{1, 2, \ldots, n\}$. The core of *LSAPR* is the so called shortest augmenting path techniques. Once such an optimal ϕ^* is found, the solution σ to (8) is given by

$$\sigma = \{i | \phi^*(i) \le m\}. \tag{31}$$

In the case when $m = n$, i.e., when the entire spectrum of A is to be matched, it is not necessary to solve (30) since the optimal σ is simply the permutation that arranges eigenvalues $\lambda_i(d)$ in the same ordering as λ_i^*.

After each iteration we measure $e_k = \|d^{(k)} - d^{(k-1)}\|$. The iteration is terminated whenever e_k is small enough. For our experiment, the threshold is set at 10^{-8} for all examples. For the LP-Newton method, the LP iteration is terminated when $e_k < \epsilon_1$. It remains an open question as to how small ϵ_1 should be so that the overall cost of computation would be optimized.

Example 1. In this example we test the performance of each individual method. In particular, we compare the performance of the LP-Newton method when the number of the LP iterations is specified. The same test data:

$$\begin{aligned} A_0 &= \begin{bmatrix} 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \end{bmatrix}; \\ A_k &= 4e_k e_k^T,\ k = 1, 2, ..., 5 \\ d^{(0)} &= [0.63160, 0.23780, 0.90920, 0.98660, 0.50070]^T; \\ \lambda^* &= [1, 1, 2, 3, 4]^T \end{aligned}$$

are used in all experiments. It turns out that all methods converge to the same least squares solution:

$$\begin{aligned} d^* &= [0.44230, 0.60440, 0.65660, 0.60440, 0.44230]^T; \\ \lambda(d^*) &= [0.58884, 1.0422, 2.07421, 3.1446, 4.1501]^T. \end{aligned}$$

Test results, measured in terms of the CPU time in seconds versus the number n of LP iterations allowed in the LP step, are recorded in Figure 3. Under the same stopping criterion, i.e., $\|e_k\| \leq 10^{-8}$, the result at $n = 0$ represents the performance of the pure Newton method, whereas the result at $n = 250$ represents the performance of the pure LP method.

The sudden drop of the CPU time in Figure 3 is interesting and important. Since we know the cost of the LP iteration is linear in n, the drop of the CPU should be attributed solely to *one less* Newton iteration required to reach the accuracy of convergence. Note that the magnitude of each drop is approximately the same, supporting the conjecture that the drop is caused by one single and resembling event. The staircase in Figure 3 is also interesting. It indicates that the extra LP iterations between two consecutive drops are futile because the number of Newton iterations required to reach the same accuracy has not been reduced.

Although the size of our test problem is small so that the Newton method is not unbearably expensive, Figure 3 does suggest that switching to Newton method after a certain number (23 in this case) of LP iterations will minimize the over all CPU times.

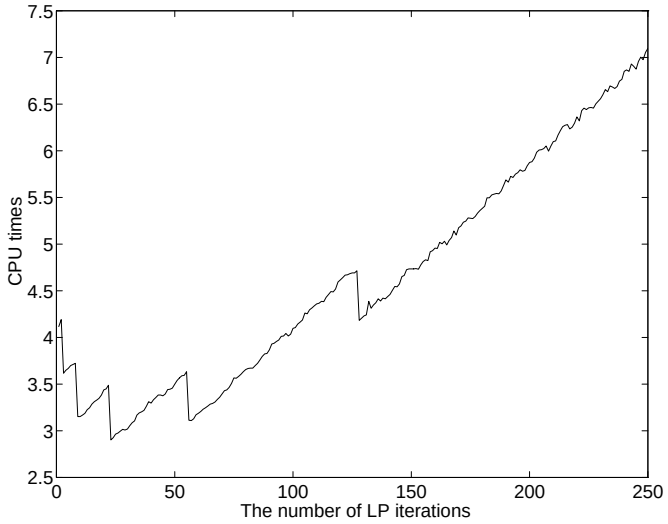

FIG. 3. *The numbers of LP iterations vs the CPU times for Example 1.*

Example 2. In this example we want to construct a symmetric Toeplitz matrix with partially prescribed eigenvalues. This is a special case of the famous inverse Toeplitz eigenvalue problem. We formulate the problem as a least squares problem with test data given below:

$$\begin{aligned}
A_0 &= 0; \\
A_k &= (A_{ij}^{(k)}) \in R^{20\times 20} \\
A_{ij}^{(k)} &:= \begin{cases} 1, & if\ |i-j| = k-1; \\ 0, & otherwise. \end{cases} \\
d^{(0)} &= [1.1650, 0.6268, 0.0751, 0.3516, -0.6965, 1.6961, 0.0591, \\
&\quad 1.7971, 0.2641, 0.8717, -1.4462, -0.7012, 1.2460, -0.6390, \\
&\quad 0.5773, -0.3600, -0.1356, -1.3493, -1.2704, 0.9845]^T; \\
\lambda^* &= [-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5]^T.
\end{aligned}$$

This time we terminate the LP step in the LP-Newton method according to Algorithm 5.1. We have observed two interesting results. One is that, although both methods start from the same point $d^{(0)}$, the LP-Newton and the Newton method converge to distinct solutions. They are, respectively,

$$\begin{aligned}
d^*_{LP-Newton} &= [0.8486, 0.8424, -0.0050, 0.3076, -0.5089, 1.6325, -0.0659, \\
&\quad 1.72764, -0.00038, 1.1018, -1.5155, -0.8286, 1.1952, -0.7433, \\
&\quad 0.0336, -0.0737, 0.0356, -1.5870, -0.1220, -0.2275]^T,
\end{aligned}$$

and

$$d^*_{Newton} = [7.3464, 1.5410, -6.9636, -1.5020, 5.6980, 5.0777, -4.4890,$$

$$-6.4796, 3.3743, 8.2720, -0.6983, -9.4107, -1.0345, 6.8718,$$
$$3.1606, -8.8348, -4.3902, 2.4603, 6.7335, -4.9145]^T.$$

The other is that, although the resulting matrices $A(d^*)$ are different, both solutions produce eigenvalues that agree with the given λ^* almost perfectly, i.e., $F(d^*, \sigma^*) \approx 10^{-8}$. Had we known that a perfect match was about to happen, we could have used the cheaper Gauss-Newton method instead of the Newton method to achieve the ultimate quadratic rate of convergence. In general, however, the Gauss-Newton method is less in favor because the LP method does not need the differentiation of eigenvalues.

The computational cost is summarized in Table 1. As far as the CPU time is concerned, it is seen that the LP-Newton method converges almost three times faster than the Newton method. It also indicates that improving the accuracy of the LP step from $\epsilon_1 = .01$ to $\epsilon_1 = 0.001$ is not necessarily advantageous even though the number of the Newton iterations is reduced by 2.

	Algorithm	Number of Iterations	CPU times (seconds)
1	Newton	24	917.215
2	LP-Newton($\epsilon_1 = .01$)	(LP) 57	305.156
		(Newton) 7	
3	LP-Newton($\epsilon_1 = .001$)	(LP) 434	349.195
		(Newton) 5	

TABLE 1
Computational cost for Example 2.

The history of errors e_k is plotted in Figure 4. Observe that the Newton method wanders for quite a few steps in the initial stage before it eventually converges to a solution. The LP method, with its descent property, helps to move into a better region for starting the Newton iteration.

Example 3. In this example we test the MIEP with partially prescribed eigenvalues. We consider the positive definite matrix:

$$A = \begin{bmatrix} T & E & 0 & 0 \\ E & T & E & 0 \\ 0 & E & T & E \\ 0 & 0 & E & T \end{bmatrix}$$

where

$$T = \begin{bmatrix} 4 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 \\ 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 4 \end{bmatrix};$$
$$E = -I_4.$$

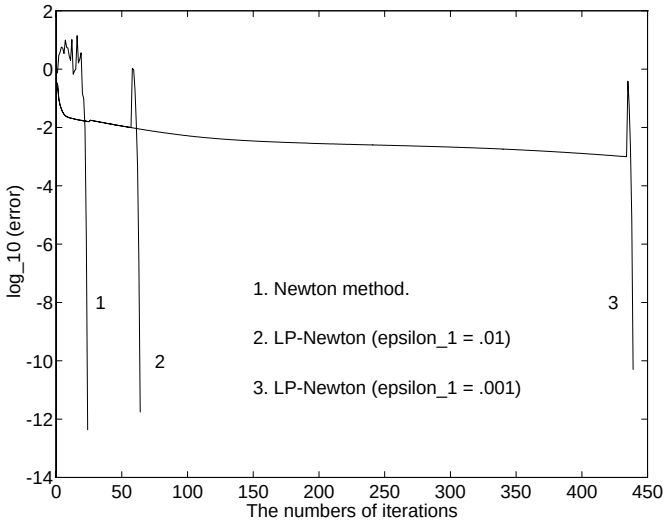

FIG. 4. *The plot of* $\log_{10}|e_k|$ *for Example 2*

We first change the problem into an IEP by meanings of the Cholesky factorization $A = LL^T$. That is, we are now considering an IEP with the following test data:

$$\begin{aligned} A_0 &= 0; \\ A_k &= L^T E_k L; \\ d^{(0)} &= [1.5578, -2.4443, -1.0982, 1.1226, 0.5817, \\ &\quad -0.2714, 0.4142, -0.9778, -1.0215, 0.3177, \\ &\quad 1.5161, 0.7494, -0.5077, 0.8853, -0.2481, -0.7262]^T; \\ \lambda^* &= [1, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50]^T. \end{aligned}$$

The LP-Newton method converges to a least squares solution

$$\begin{aligned} d^* &= [10.2309, -3.0078, -1.6975, 10.1958, 7.2102, \\ &\quad 2.4626, 5.8098, -1.9979, -1.53203.7608, \\ &\quad 10.0604, 8.5959, 0.2992, 8.0485, 3.4645, -1.0845]^T, \end{aligned}$$

at which again λ^* is perfectly matched with 11 eigenvalues of $A(d^*) = D^*A$. In contrast, it seems reasonable to conclude from Figure 5 that the Newton method diverges after 40 iterates. On the other hand, with a good start value provided by the LP method (with $\epsilon_1 = 0.001$), the Newton method converges to a solution with desired accuracy $\epsilon = 10^{-8}$ within 3 iterates.

The computational cost is summarized in Table 2. None of the methods is cheap, but the advantage of the LP-Newton method is obvious in this case.

Acknowledgments. The authors are grateful to the referees for pointing out that the Hessian formula (28) remains valid at coalescent eigenvalues. Theorem 3.1, also suggested by the referees, not only simplifies the proof of Theorem 3.2 but is of interest in its own right.

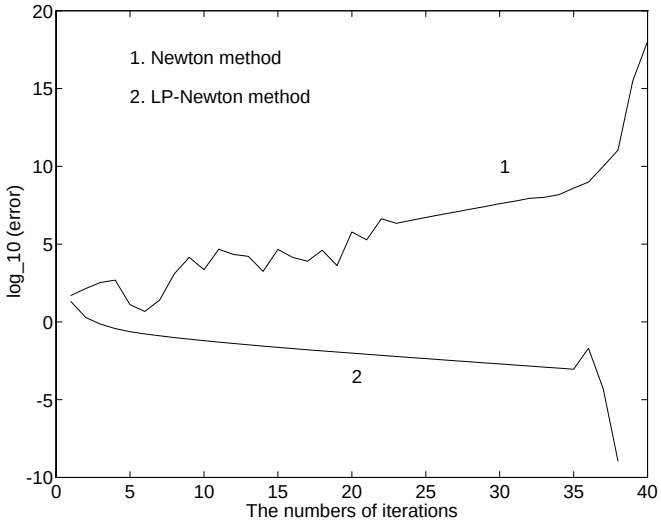

FIG. 5. *The plot of* $\log_{10}|e_k|$ *for Example 3*

	Algorithm	Number of Iterations	CPU times (seconds)
1	Newton	≥ 40	≥ 923.123
2	LP-Newton	(LP) 35	92.644
		(Newton) 3	

TABLE 2
Computational cost for Example 3.

REFERENCES

[1] F. W. Biegler-Konig, A Newton iteration process for inverse eigenvalue problems, Numer. Math., 37(1981), pp. 349-354.

[2] D. L. Boley and G. H. Golub, A survey of inverse eigenvalue problems, Inverse Problems, 3(1987), 595-622.

[3] R. W. Brockett, Dynamical systems that sort lists and solve linear programming problems, in Proc. 27th IEEE Conf. Decision and Control, (1988), 799-803, and Linear Alg. Appl., 146(1991), 79-91.

[4] W. Cheney and A. Goldstein, Proximity maps for convex sets, Proc. Amer. Math. Soc., 10(1959), 448-450.

[5] M. T. Chu, Matrix differential equations: A continuous realization process for linear algebra problems, Nonlinear Anal., TMA, Vol. 1, No. 12, pp. 1125-1146, 1992.

[6] M. T. Chu, Numerical methods for inverse singular problems, SIAM J. Numer. Anal., 29(1992), pp. 885-903.

[7] M. T. Chu, Constructing a Hermitian matrix from its diagonal entries and eigenvalues, SIAM J. Matrix Anal., to appear.

[8] J. E. Dennis, Jr. and R. B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Inc., Englewood Cliffs, 1983.

[9] F. Deutsch, Von Neumann's alternating method: the rate of convergence, Approximation Theory IV, C. Chui, L. Schumaker and J. Ward, ed., Academic Press, New York, 1983, 427-434.

[10] G. E. Forsythe and E. G. Strauss, On best conditioned matrices, Proc. Amer. Math. Soc., 6(1955), 340-345.

[11] S. Friedland, On inverse multiplicative eigenvalue problems for matrices, Lin. Alg. Appl., 12(1985), pp. 127-137.
[12] S. Friedland, Inverse eigenvalue problems, Lin. Alg. Appl., 17(1977), pp. 15-51.
[13] S. Friedland, J. Nocedal, and M.L. Overton, The formulation and analysis of numerical methods for inverse eigenvalue problems, SIAM J. Numer. Anal., 24(1987), pp. 634-667.
[14] W. K. Glunt, An alternating projections method for certain linear problems in a Hilbert space, preprint, University of Kentucky, 1992.
[15] A. Greenbaum and G. H. Rodrigue, Optimal preconditioners of a given sparsity pattern, BIT, 29(1989), 610-634.
[16] A. Greenbaum, Diagonally scalings of the Laplacian as preconditioners for other elliptic differential operators, SIAM J. Matrix Anal. Appl., 13(1992), 826-846.
[17] S. P. Han, A successive projection method, Math. Programming, 40(1988), 1-14.
[18] R. A. Horn and C. R. Johnson, Matrix Analysis, Cambridge University Press, New York, 1991.
[19] P. Lancaster, On eigenvalues of matrices dependent on a parameter, Numer. Math., 6, pp. 377-387.
[20] R. E. Burkard and U. Derigs, Assignment and Matching Problems: Solution Methods with FORTRAN-Programs, Springer-Verlag, Berlin Heidelberg New York 1980.
[21] A. van der Sluis, Condition numbers and equilibration matrices, Numer. Math., 14(1969), 14-23.