Invited Paper

Formulations and theorems of quadratically convergent methods for inverse symmetric eigenvalue problems

Kensuke Aishima^{1 a)}

¹ Faculty of Computer and Information Sciences, Hosei University, Tokyo
184-8585, Japan

^{a)} aishima@hosei.ac.jp

Received October 14, 2019; Revised March 17, 2020; Published July 1, 2020

Abstract: Inverse eigenvalue problems arise in a variety of applications, and thus various Newton's methods, which quadratically converge, have been developed both in theory and practice. Among many studies over thirty years, two extremely significant developments are found. Firstly, smooth matrix decompositions have been successfully applied since the 1990s. Secondly, a matrix multiplication based method has been recently proposed. In this paper, such efficient modern solvers are classified in the context of classical Newton's methods according to their mathematical formulations, and then the corresponding convergence theorems and their relationship are surveyed.

Key Words: inverse eigenvalue problems, generalized symmetric eigenvalue problems, Newton's method, quadratic convergence, multiple eigenvalues

1. Introduction

Matrix eigenvalue problems are very important in many large scale scientific or engineering computations. Inverse eigenvalue problem concerns reconstructions of structured matrices from spectral data, arising from a variety of applications. There is a long history of developments of many algorithms for solving the inverse eigenvalue problems. Excellent overviews are given in Chu [14], Chu–Golub [16, 17], and so forth. Thanks to such literatures, the inverse eigenvalue problems attract current attention as an important technique of the mathematical modeling in a variety of research fields, including data science as well as classical scientific computing. A typical example is to construct a structured low rank matrix, where the prescribed eigenvalues are zero. The low rank matrix completion is successfully applied to recommender systems, collaborative filtering, compressed sensing, and so forth.

In this paper, we consider the following simple and typical inverse eigenvalue problems. Let $A_0, A_1, \dots, A_n$ be real symmetric $n \times n$ matrices and $\lambda_1^* \leq \lambda_2^* \leq \dots \leq \lambda_n^*$ be real numbers. In addition, let $\mathbf{c} = [c_1, \dots, c_n]^T \in \mathbb{R}^n$, $\Lambda^* = \text{diag}(\lambda_1^*, \dots, \lambda_n^*)$. Define

$$A(\mathbf{c}) := A_0 + c_1 A_1 + \dots + c_n A_n \quad (1)$$

and denote its eigenvalues by $\lambda_1(\mathbf{c}) \leq \lambda_2(\mathbf{c}) \leq \dots \leq \lambda_n(\mathbf{c})$ in the ascending order. We aim to find a solution $\mathbf{c}^* \in \mathbb{R}^n$ such that $\lambda_i(\mathbf{c}^*) = \lambda_i^*$ for all $1 \leq i \leq n$. Such inverse eigenvalue problems often

arise in inverse vibration problems, inverse Sturm-Liouville problems, and nuclear spectroscopy [14, 16, 17, 24]. In this study, we focus on numerical algorithms for solving the above inverse eigenvalue problems. As in [7, 10, 11, 24], we assume that the prescribed eigenvalues are all distinct, i.e.,

$$\lambda_1^* < \lambda_2^* < \cdots < \lambda_n^*. \quad (2)$$

Multiple eigenvalues are discussed in Section 5 as an exceptional case. Let $X^* \in \mathbb{R}^{n \times n}$ denote an orthogonal matrix whose columns are the eigenvectors of $A(\mathbf{c}^*)$. Throughout the paper, I is an identity matrix and O is a zero matrix. For any matrix, let $\|\cdot\|$ denote the spectral norm and $[\cdot]_{ij}$ denote the (i, j) elements for $1 \leq i, j \leq n$.

One of the effective numerical algorithms is a projected gradient method, given by an ordinary differential equation characterizing a steepest descent flow. Although such a numerical method is globally convergent as in [13, 15] and [17, §3.2.5], in general, the convergence rate is slow. Thus, the globally convergent method is often used to get an approximate solution in a particular domain, and then locally fast methods are applied. One of the locally fast methods is a Newton's method that is quadratically convergent.

The basic formulations of the quadratically convergent methods are given in [24] by Friedland, Nocedal, and Overton in 1987. Over the passed thirty years, many efforts have been made for their developments both in theory and practice. From the theoretical viewpoint, in 1999, Chan, Xu, and Zhou pointed out that the proof of quadratic convergence of one of the above methods [24, Theorem 3.3] was incorrect, and then gave a correct proof in [11]. In addition, backward error analysis for the inverse symmetric eigenvalue problems is given in [30, 43]. As the algorithmic progress, to the best of the author's knowledge, there have been two significant developments. Firstly, smooth matrix decompositions are successfully applied to Newton's methods since the 1990s [18, 19, 28, 29, 46]. Secondly, a matrix multiplication based method has been recently proposed [1, 2], where the matrix multiplications can be efficiently performed by the Level 3 BLAS (Basic Linear Algebra Subprograms) routines. Moreover, concerning its accuracy, high-precision matrix multiplication by error-free transformations is currently available [36, 37]. The purpose of this paper is to organize the above efficient modern solvers in the context of the classical methods in [24], and to survey the corresponding convergence theorems and their relationship, clarifying the recent progress.

Since they are a kind of Newton's methods, a linear system for the Jacobi matrix must be solved per iteration. Thus, inexact linear system solvers are successfully applied as in [7, 8, 10, 40]. The Ulm-like method can also be applied as in [39]. For simplicity, however, we do not focus on the details of such inexact Newton methods. We discuss the basic formulations and the corresponding convergence theorems.

From such a perspective, we classify the basic formulations into three versions as follows:

- (a) Direct formulation using eigenvalues
- (b) Use of determinant equations and smooth matrix decompositions
- (c) Formulation by matrix equations

The idea of item (a) is to define $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ by

$$\mathbf{f}(\mathbf{c}) = [\lambda_1(\mathbf{c}) - \lambda_1^*, \lambda_2(\mathbf{c}) - \lambda_2^*, \dots, \lambda_n(\mathbf{c}) - \lambda_n^*]^T,$$

where $\lambda_i(\mathbf{c})$ ($1 \leq i \leq n$) are the eigenvalues of $A(\mathbf{c})$ in (1) and λ_i^* ($1 \leq i \leq n$) are the prescribed eigenvalues. This is a standard approach to inverse problems. Thus, it can be extended to other complicated inverse eigenvalue problems. However, a numerical solution of the eigenvalue problem is required in each iteration.

Item (b) concerns other formulations. Firstly, define

$$\mathbf{f}(\mathbf{c}) = [\det(A(\mathbf{c}) - \lambda_1^* I), \det(A(\mathbf{c}) - \lambda_2^* I), \dots, \det(A(\mathbf{c}) - \lambda_n^* I)]^T,$$

where $\det$ means determinant. According to [24], although the Jacobi matrix can be computed, this Newton's method is not so efficient. Since the 1990s, however, smooth matrix decompositions have attracted attentions for solving inverse eigenvalue problems. In other words, if we replace the above $\mathbf{f}$ by slightly different functions using smooth matrix decompositions, we obtain efficient quadratically convergent algorithms [18, 19, 28, 29, 46]. This approach can be extended to other complicated problems for parametrized matrices.

Item (c) has two versions. The first one is to find X satisfying the following matrix equations:

$$X = \exp(Y), \quad Y = -Y^T, \quad X^T A(\mathbf{c}) X = \Lambda^* := \text{diag}(\lambda_1^*, \dots, \lambda_n^*).$$

To avoid the matrix exponential, the Cayley transform is applied, resulting in quadratically convergent algorithm without computing eigenpairs per iteration [24, Method III]. Recently, another nice idea has been presented. This is more simple formulation in some sense, i.e.,

$$\begin{cases} X^T X = I \\ X^T A(\mathbf{c}) X = \Lambda^* \end{cases},$$

resulting in a quadratically convergent algorithm primarily comprising matrix multiplications [1, 2]. The above formulations by matrix equations are efficient for the inverse symmetric eigenvalue problems, though they are not easily applicable to other kinds of problems.

From the above classification, their relationships are clarified with mathematical rigor. To this end, we introduce smooth LU decomposition, as an example of (b), and show that the Jacobi matrix can be computed analogously to (a). Such a derivation is different from the original paper [19]. Moreover, we explicitly design an algorithm using the block smooth LU decompositions for adapting to multiple eigenvalues. As for convergence analysis, since they are Newton's methods, quadratic convergence is naturally proved. Importantly, the quadratic convergence is theoretically guaranteed even if multiple eigenvalues are specified. In this paper, we survey the convergence theorems with the discussion of multiple eigenvalues. Among them, it is worth noting that the convergence theorem of the matrix multiplication based method [1, 2] in (c) states an explicit constant concerning the condition and the order estimation using an optimization problem.

Our mathematical discussions use differentiations of parametrized matrices. In general, for the parametrized matrices $M(t) \in \mathbb{R}^{\ell \times m}$, $N(t) \in \mathbb{R}^{m \times n}$, if they are differentiable for some t , the differentiation of the matrix product $M(t)N(t)$ is represented by

$$\frac{dM(t)N(t)}{dt} = \frac{dM(t)}{dt}N(t) + M(t)\frac{dN(t)}{dt}$$

in view of the product rule. In the following, we often use the above calculation.

The remainder of this paper is organized as follows. The above three formulations are explained in the following three sections (Sections 2, 3, and 4). In Section 5, we survey convergence theorems for the Newton-like methods. Section 6 is devoted to descriptions of recent progress in other research fields related to the Newton methods. Finally, we give concluding remarks in Section 7.

2. Direct formulation using eigenvalues

In this section, let us see a typical formulation of Newton's method for inverse problems. For any $\mathbf{c} \in \mathbb{R}^n$, define $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ by

$$\mathbf{f}(\mathbf{c}) = [\lambda_1(\mathbf{c}) - \lambda_1^*, \lambda_2(\mathbf{c}) - \lambda_2^*, \dots, \lambda_n(\mathbf{c}) - \lambda_n^*]^T, \quad (3)$$

where $\lambda_i(\mathbf{c})$ ($1 \leq i \leq n$) are the eigenvalues of $A(\mathbf{c})$ in (1) and λ_i^* ($1 \leq i \leq n$) are the prescribed eigenvalues. We assume that all the prescribed eigenvalues are distinct as in (2). Thus, since all the eigenvalues of $A(\mathbf{c})$ are also distinct in some neighborhood of $\mathbf{c}^*$, the above function $\mathbf{f}$ is analytic in such a neighborhood. Hence, let $\mathbf{x}_i(\mathbf{c})$ ($1 \leq i \leq n$) denote the normalized eigenvectors corresponding to $\lambda_i(\mathbf{c})$ ($1 \leq i \leq n$). Then, using the Kronecker delta $\delta_{ij} := [I]_{ij}$ ($1 \leq i, j \leq n$), we see that

$$\mathbf{x}_i(\mathbf{c})^T \mathbf{x}_j(\mathbf{c}) = \delta_{ij} \quad (1 \leq i, j \leq n), \quad (4)$$

$$\mathbf{x}_i(\mathbf{c})^T A(\mathbf{c}) \mathbf{x}_j(\mathbf{c}) = \delta_{ij} \lambda_i(\mathbf{c}) \quad (1 \leq i, j \leq n), \quad (5)$$

corresponding to

$$X(\mathbf{c})^T X(\mathbf{c}) = I, \quad X(\mathbf{c})^T A(\mathbf{c}) X(\mathbf{c}) = \Lambda^*(\mathbf{c}) := \text{diag}(\lambda_1(\mathbf{c}), \dots, \lambda_n(\mathbf{c})). \quad (6)$$

Using the above relations, we have the Jacobi matrix as

$$\begin{aligned} [J(\mathbf{c})]_{ij} &= \frac{\partial \lambda_i(\mathbf{c})}{\partial c_j} \\ &= \frac{\partial (\mathbf{x}_i(\mathbf{c})^T A(\mathbf{c}) \mathbf{x}_i(\mathbf{c}))}{\partial c_j} \\ &= \left(\frac{\partial \mathbf{x}_i(\mathbf{c})}{\partial c_j} \right)^T A(\mathbf{c}) \mathbf{x}_i(\mathbf{c}) + \mathbf{x}_i(\mathbf{c})^T \left(\frac{\partial A(\mathbf{c})}{\partial c_j} \right) \mathbf{x}_i(\mathbf{c}) + \mathbf{x}_i(\mathbf{c})^T A(\mathbf{c}) \left(\frac{\partial \mathbf{x}_i(\mathbf{c})}{\partial c_j} \right) \\ &= \left(\frac{\partial \mathbf{x}_i(\mathbf{c})}{\partial c_j} \right)^T \lambda_i(\mathbf{c}) \mathbf{x}_i(\mathbf{c}) + \mathbf{x}_i(\mathbf{c})^T \left(\frac{\partial A(\mathbf{c})}{\partial c_j} \right) \mathbf{x}_i(\mathbf{c}) + \mathbf{x}_i(\mathbf{c})^T \lambda_i(\mathbf{c}) \left(\frac{\partial \mathbf{x}_i(\mathbf{c})}{\partial c_j} \right) \\ &= \lambda_i(\mathbf{c}) \left(\left(\frac{\partial \mathbf{x}_i(\mathbf{c})}{\partial c_j} \right)^T \mathbf{x}_i(\mathbf{c}) + \mathbf{x}_i(\mathbf{c})^T \left(\frac{\partial \mathbf{x}_i(\mathbf{c})}{\partial c_j} \right) \right) + \mathbf{x}_i(\mathbf{c})^T \left(\frac{\partial A(\mathbf{c})}{\partial c_j} \right) \mathbf{x}_i(\mathbf{c}), \end{aligned} \quad (7)$$

where the second equality is due to (5). Differentiating (4), we see

$$\left(\frac{\partial \mathbf{x}_i(\mathbf{c})}{\partial c_j} \right)^T \mathbf{x}_i(\mathbf{c}) + \mathbf{x}_i(\mathbf{c})^T \left(\frac{\partial \mathbf{x}_i(\mathbf{c})}{\partial c_j} \right) = 0. \quad (8)$$

Therefore, we obtain

$$[J(\mathbf{c})]_{ij} = \mathbf{x}_i(\mathbf{c})^T \left(\frac{\partial A(\mathbf{c})}{\partial c_j} \right) \mathbf{x}_i(\mathbf{c}) = \mathbf{x}_i(\mathbf{c})^T A_j \mathbf{x}_i(\mathbf{c}) \quad (9)$$

from (7) and (1).

In general, noting the Taylor series at $\mathbf{c}$, we have

$$\mathbf{0} = \mathbf{f}(\mathbf{c}^*) = \mathbf{f}(\mathbf{c}) + J(\mathbf{c})(\mathbf{c}^* - \mathbf{c}) + \mathcal{O}(\|\mathbf{c} - \mathbf{c}^*\|^2) \quad (10)$$

for the solution $\mathbf{c}^*$ of the nonlinear equation. Hence, the Newton's iterations are constructed such that

$$\mathbf{0} = \mathbf{f}(\mathbf{c}^{(k)}) + J(\mathbf{c}^{(k)})(\mathbf{c}^{(k+1)} - \mathbf{c}^{(k)}) \quad (k = 0, 1, \dots). \quad (11)$$

If $J(\mathbf{c}^{(k)})$ ($k = 0, 1, \dots$) are nonsingular, we have the Newton's method:

$$\mathbf{c}^{(k+1)} = \mathbf{c}^{(k)} - J(\mathbf{c}^{(k)})^{-1} \mathbf{f}(\mathbf{c}^{(k)}) \quad (k = 0, 1, \dots). \quad (12)$$

This is the standard Newton's method for general nonlinear equations. For the inverse eigenvalue problems, we write the above method in a slightly different form. From (1), (5), and (9), we have

$$[J(\mathbf{c})\mathbf{c}]_i = \lambda_i(\mathbf{c}) - \mathbf{x}_i(\mathbf{c})^T A_0 \mathbf{x}_i(\mathbf{c}) \quad (1 \leq i \leq n). \quad (13)$$

In addition, from (11), we see

$$\mathbf{0} = \mathbf{f}(\mathbf{c}^{(k)}) + J(\mathbf{c}^{(k)})\mathbf{c}^{(k+1)} - J(\mathbf{c}^{(k)})\mathbf{c}^{(k)} \quad (k = 0, 1, \dots).$$

Moreover, using (3), we have an iterative method that can be described as follows.

The Newton's method does not break down whenever the Jacobi matrix $J(\mathbf{c})$ in (9) is nonsingular. Importantly, if $\|J(\mathbf{c})^{-1}\|$ is bounded in a neighborhood of $\mathbf{c}^*$, quadratic convergence

---

Algorithm 1 The Newton's method [24, Method I]

Require: $\lambda_1^* < \dots < \lambda_n^*$, $A_0, A_1, \dots, A_n \in \mathbb{R}^{n \times n}$, $\mathbf{c}^{(0)} \in \mathbb{R}^n$.

- 1: For $A(\mathbf{c}^{(0)})$, find the eigenpairs $(\lambda_i^{(0)}, \mathbf{x}_i^{(0)})$ for all $1 \leq i \leq n$
  - 2: **for** $k := 0, 1, \dots$ **do**
  - 3: $[J^{(k)}]_{ij} = \mathbf{x}_i^{(k)\top} A_j \mathbf{x}_i^{(k)}$
  - 4: $[\mathbf{d}^{(k)}]_i = \lambda_i^* - \mathbf{x}_i^{(k)\top} A_0 \mathbf{x}_i^{(k)}$
  - 5: $\mathbf{c}^{(k+1)} = J^{(k)^{-1}} \mathbf{d}^{(k)}$
  - 6:   For $A(\mathbf{c}^{(k+1)})$, find the eigenpairs $(\lambda_i^{(k+1)}, \mathbf{x}_i^{(k+1)})$ for all $1 \leq i \leq n$
  - 7: **end for**
-

$$\|\mathbf{c}^{(k+1)} - \mathbf{c}^*\| = \mathcal{O}(\|\mathbf{c}^{(k)} - \mathbf{c}^*\|^2) \quad (14)$$

is naturally proved by the difference between (11) and

$$\mathbf{0} = \mathbf{f}(\mathbf{c}^{(k)}) + J(\mathbf{c}^{(k)})(\mathbf{c}^* - \mathbf{c}^{(k)}) + \mathcal{O}(\|\mathbf{c}^{(k)} - \mathbf{c}^*\|^2),$$

given by $\mathbf{c} := \mathbf{c}^{(k)}$ in (10).

One may notice that, in general, constructing $J^{(k)}$ requires $\mathcal{O}(n^4)$ operations. From the practical view points, however, there are some examples where the number of the nonzero elements of A_j is $\mathcal{O}(n)$ for each j . Typical examples are the Toeplitz inverse problem and the discretization of inverse Sturm-Liouville problem [14, 16, 17, 24]. In such situations, Algorithms 1 can be computed in $\mathcal{O}(n^3)$ operations.

Since the Newton's method is straightforward for solving inverse problems, Algorithm 1 can be extended to inverse generalized eigenvalue problems and inverse quadratic eigenvalue problems [20, 21]. However, Algorithm 1 requires a numerical solution of the eigenvalue problem for $A(\mathbf{c}^{(k+1)})$ in each iteration. There are some quasi-Newton's methods without exactly solving such eigenvalue problems [8, 10, 11, 24, 39]. Such methods are based on the Householder tridiagonalization and the inverse iterations, important parts of the standard methods for solving symmetric eigenvalue problems [25, 38]. The algorithm is summarized as follows.

Algorithm 2 The quasi-Newton's method [24, Method II]

Require: $\lambda_1^* < \dots < \lambda_n^*$, $A_0, A_1, \dots, A_n \in \mathbb{R}^{n \times n}$, $\mathbf{c}^{(0)} \in \mathbb{R}^n$.

- 1: For $A(\mathbf{c}^{(0)})$, find the eigenpairs $(\lambda_i^{(0)}, \mathbf{x}_i^{(0)})$ for all $1 \leq i \leq n$
  - 2: **for** $k := 0, 1, \dots$ **do**
  - 3: $[J^{(k)}]_{ij} = \mathbf{x}_i^{(k)\top} A_j \mathbf{x}_i^{(k)}$
  - 4: $[\mathbf{d}^{(k)}]_i = \lambda_i^* - \mathbf{x}_i^{(k)\top} A_0 \mathbf{x}_i^{(k)}$
  - 5: $\mathbf{c}^{(k+1)} = J^{(k)^{-1}} \mathbf{d}^{(k)}$
  - 6:   Tridiagonalization by the Householder transformation: $U^{(k+1)\top} A(\mathbf{c}^{(k+1)}) U^{(k+1)} = T^{(k+1)}$
  - 7: $\mathbf{y}_i^{(k+1)} = (T^{(k+1)} - \lambda_i^* I)^{-1} U^{(k+1)\top} \mathbf{x}_i^{(k)}$ for all $1 \leq i \leq n$
  - 8: $\tilde{\mathbf{y}}_i^{(k+1)} = U^{(k+1)} \mathbf{y}_i^{(k+1)}$ for all $1 \leq i \leq n$
  - 9: $\mathbf{x}_i^{(k+1)} = \tilde{\mathbf{y}}_i^{(k+1)} / \|\tilde{\mathbf{y}}_i^{(k+1)}\|$ for all $1 \leq i \leq n$
  - 10: **end for**
-

Although the eigenvalue computations can be avoided, this makes quadratic convergence less straightforward. Intuitively, since the convergence rate of the shifted inverse iteration with the Rayleigh quotient is cubic [38], the above approximations of the eigenvectors do not cause loss of the quadratic convergence. According to [24, Theorem 3.3], the quadratic convergence was proved in the convergence analysis for a kind of Newton's methods. In 1999, however, Chan, Xu, and Zhou pointed out that the proof of [24, Theorem 3.3] was incorrect and then gave a correct proof in [11]. The incorrect proof is due to the loss of orthogonality of $X^{(k)}$, while the norms of the column vectors are normalized to 1. In [11], the proof of quadratic convergence $\|\mathbf{c}^{(k+1)} - \mathbf{c}^*\| = \mathcal{O}(\|\mathbf{c}^{(k)} - \mathbf{c}^*\|^2)$ is given, where the orthogonality is not assumed.

3. Use of determinant equations and smooth matrix decompositions

In this section, we discuss a couple of formulations different from the previous section.

In 1981, Biegler-König [9] introduced

$$\mathbf{f}(\mathbf{c}) = [\det(A(\mathbf{c}) - \lambda_1^* I), \det(A(\mathbf{c}) - \lambda_2^* I), \dots, \det(A(\mathbf{c}) - \lambda_n^* I)]^T, \quad (15)$$

where $\det$ means determinant.

To form the corresponding Jacobi matrix, we firstly consider the differentiation of a parameterized matrix. In general, if all the elements of an $n \times n$ matrix $F(t)$ are differentiable functions of $t \in \mathbb{R}$, then for any t such that $F(t)$ is nonsingular, we have

$$p(t) := \det(F(t)), \quad \frac{dp(t)}{dt} = p(t) \text{Tr} \left(F(t)^{-1} \frac{dF(t)}{dt} \right), \quad (16)$$

where Tr means trace. Its proof is written in [27, Appendix II], known as the Trace-Theorem originally proved by Davidenko. Using this formula, we have the Jacobi matrix of $\mathbf{f}(\mathbf{c})$ as

$$[J(\mathbf{c})]_{ij} = \frac{\partial \det(A(\mathbf{c}) - \lambda_i^* I)}{\partial c_j} = \det(A(\mathbf{c}) - \lambda_i^* I) \text{Tr}((A(\mathbf{c}) - \lambda_i^* I)^{-1} A_j). \quad (17)$$

One of the naive numerical methods for computing $\mathbf{f}(\mathbf{c})$ and $J(\mathbf{c})$ is to use the LU decompositions. In other words, for any $\mathbf{c}$, using appropriate permutation matrices $P_{\text{left},i}(\mathbf{c})$ and $P_{\text{right},i}(\mathbf{c})$ for $i = 1, \dots, n$, we compute

$$L_i(\mathbf{c})U_i(\mathbf{c}) := P_{\text{left},i}(\mathbf{c})(A(\mathbf{c}) - \lambda_i^* I)P_{\text{right},i}(\mathbf{c}) \quad (i = 1, \dots, n), \quad (18)$$

where $L_i(\mathbf{c})$ ($i = 1, \dots, n$) are the lower triangular matrices whose diagonal elements are 1, and $U_i(\mathbf{c})$ ($i = 1, \dots, n$) are the upper triangular matrices. Letting

$$\text{sign}_i(\mathbf{c}) = \det(P_{\text{left},i}(\mathbf{c})) \det(P_{\text{right},i}(\mathbf{c})),$$

we see

$$\begin{aligned} [\mathbf{f}(\mathbf{c})]_i &= \det(A(\mathbf{c}) - \lambda_i^* I) = \text{sign}_i(\mathbf{c}) \det(U_i(\mathbf{c})) = \text{sign}_i(\mathbf{c}) \prod_{j=1}^n [U_i(\mathbf{c})]_{jj} \quad (i = 1, \dots, n), \\ [J(\mathbf{c})]_{ij} &= \text{sign}_i(\mathbf{c}) \prod_{j=1}^n [U_i(\mathbf{c})]_{jj} \text{Tr}(P_{\text{right},i}(\mathbf{c})U_i(\mathbf{c})^{-1}L_i(\mathbf{c})^{-1}P_{\text{left},i}(\mathbf{c})A_j) \quad (1 \leq i, j \leq n). \end{aligned}$$

Thus, the Newton's method can be executed. Since this approach is flexible, it can be applied to inverse polynomial eigenvalue problems as in [23]. However, n LU decompositions require $\mathcal{O}(n^4)$ arithmetic operations. To reduce the arithmetic operations, we compute the eigenvalue decomposition, i.e., $A(\mathbf{c}) = X(\mathbf{c})\Lambda(\mathbf{c})X(\mathbf{c})^T$, $\Lambda(\mathbf{c}) = \text{diag}(\lambda_1(\mathbf{c}), \dots, \lambda_n(\mathbf{c}))$, $X(\mathbf{c})^T X(\mathbf{c}) = I$. Then, from (17),

$$\begin{aligned} [J(\mathbf{c})]_{ij} &= \det(A(\mathbf{c}) - \lambda_i^* I) \text{Tr}((A(\mathbf{c}) - \lambda_i^* I)^{-1} A_j) \\ &= \det(\Lambda(\mathbf{c}) - \lambda_i^* I) \text{Tr}(X(\mathbf{c})(\Lambda(\mathbf{c}) - \lambda_i^* I)^{-1} X(\mathbf{c})^T A_j) \\ &= \left(\prod_{\ell=1}^n (\lambda_\ell(\mathbf{c}) - \lambda_i^*) \right) \text{Tr}(X(\mathbf{c})^T A_j X(\mathbf{c})(\Lambda(\mathbf{c}) - \lambda_i^* I)^{-1}) \\ &= \sum_{\ell'=1}^n \mathbf{x}_{\ell'}(\mathbf{c})^T A_j \mathbf{x}_{\ell'}(\mathbf{c}) \prod_{\ell=1, \ell \neq \ell'}^n (\lambda_\ell(\mathbf{c}) - \lambda_i^*). \end{aligned}$$

This is consistent with the following calculations using $\det(A(\mathbf{c}) - \lambda_i^* I) = \prod_{\ell=1}^n (\lambda_\ell(\mathbf{c}) - \lambda_i^*)$:

$$\begin{aligned}
[J(\mathbf{c})]_{ij} &= \frac{\partial \det(A(\mathbf{c}) - \lambda_i^* I)}{\partial c_j} \\
&= \frac{\partial \prod_{\ell=1}^n (\lambda_\ell(\mathbf{c}) - \lambda_i^*)}{\partial c_j} \\
&= \sum_{\ell'=1}^n \frac{\partial \lambda_{\ell'}(\mathbf{c})}{\partial c_j} \prod_{\ell=1, \ell \neq \ell'}^n (\lambda_\ell(\mathbf{c}) - \lambda_i^*) \\
&= \sum_{\ell'=1}^n \mathbf{x}_{\ell'}(\mathbf{c})^\top A_j \mathbf{x}_{\ell'}(\mathbf{c}) \prod_{\ell=1, \ell \neq \ell'}^n (\lambda_\ell(\mathbf{c}) - \lambda_i^*).
\end{aligned}$$

The above calculations imply the Davidenko's Trace theorem in (16) because, for the eigenvalue decomposition of $F(t)$: $F(t) = X(t)\Lambda(t)X(t)^\top$, $\Lambda(t) = \text{diag}(\lambda_1(t), \dots, \lambda_n(t))$, $X(t) = [\mathbf{x}_1(t), \dots, \mathbf{x}_n(t)]$, we have

$$\begin{aligned}
\frac{dp(t)}{dt} &= \frac{d \prod_{\ell=1}^n \lambda_\ell(t)}{dt} \\
&= \sum_{\ell'=1}^n \frac{d\lambda_{\ell'}(t)}{dt} \prod_{\ell=1, \ell \neq \ell'}^n \lambda_\ell(t) \\
&= \sum_{\ell'=1}^n \mathbf{x}_{\ell'}(t)^\top \frac{dF(t)}{dt} \mathbf{x}_{\ell'}(t) \prod_{\ell=1, \ell \neq \ell'}^n \lambda_\ell(t) \\
&= \sum_{\ell'=1}^n \lambda_{\ell'}(t)^{-1} \mathbf{x}_{\ell'}(t)^\top \frac{dF(t)}{dt} \mathbf{x}_{\ell'}(t) p(t) \\
&= p(t) \text{Tr} \left(X(t)^\top \frac{dF(t)}{dt} X(t) \Lambda(t)^{-1} \right) \\
&= p(t) \text{Tr} \left(X(t) \Lambda(t)^{-1} X(t)^\top \frac{dF(t)}{dt} \right) \\
&= p(t) \text{Tr} \left(F(t)^{-1} \frac{dF(t)}{dt} \right).
\end{aligned}$$

Although the above derivation requires differentiability of all eigenvalues with respect to t , the theorem itself requires only differentiability and non-singularity of $F(t)$.

From the above mathematical discussions, we see that, since the differentiation of a parameterized matrix can be computed by its eigenpairs, $J(\mathbf{c})$ can be computed using the eigenpairs of $A(\mathbf{c})$, while it is very complicated. Although the representation of $J(\mathbf{c})$ can be simplified, the Newton's method derived from such improvements is not so efficient compared to the direct formulation as in (3) because the convergence is slower due to ill-conditioning; see [24, Method IV] for the details.

However, such an idea to use the determinant equations has attracted revived interest since the 1990s because smooth matrix decompositions can be successfully applied to similar nonlinear functions [18, 19, 28, 29, 46]. In the following, as an example, the application of the smooth LU decomposition by Dai, Bai, and Wei in [19] is explained.

Although this algorithm computes (18) in the same manner as in (15), the next formulation is slightly different. For $i = 1, \dots, n$, the upper triangle matrices can be divided into

$$U_i(\mathbf{c}) =: \begin{bmatrix} U_{i,11}(\mathbf{c}) & U_{i,12}(\mathbf{c}) \\ 0 & u_{i,nn}(\mathbf{c}) \end{bmatrix}, \quad U_{i,11}(\mathbf{c}) \in \mathbb{R}^{(n-1) \times (n-1)}, \quad U_{i,12}(\mathbf{c}) \in \mathbb{R}^{(n-1)}, \quad u_{i,nn}(\mathbf{c}) \in \mathbb{R}. \quad (19)$$

Let

$$\mathbf{f}(\mathbf{c}) = [u_{1,nn}(\mathbf{c}), u_{2,nn}(\mathbf{c}), \dots, u_{n,nn}(\mathbf{c})]^\top.$$

The reason for defining this function is as follows. Note that $\text{rank}(A(\mathbf{c}^*) - \lambda_i^* I) = n - 1$ for $i = 1, \dots, n$ because we assume that all the eigenvalues are simple in this section. Thus, in a neighborhood of $\mathbf{c}^*$,

the LU decompositions as in (18) can be computed by appropriate $P_{\text{left},i}(\mathbf{c})$ and $P_{\text{right},i}(\mathbf{c})$. Therefore, finding $\mathbf{c}$ such that $\mathbf{f}(\mathbf{c}) = \mathbf{0}$ amounts to solving the inverse eigenvalue problems.

In addition, we assume that, for $i = 1, \dots, n$, the permutation matrices $P_{\text{left},i}(\mathbf{c})$ and $P_{\text{right},i}(\mathbf{c})$ for a fixed $\mathbf{c}$ in (18) are constant matrices in a neighborhood of $\mathbf{c}$. Note that, for each i , $u_{i,nn}(\mathbf{c})$ is a rational function of $\mathbf{c}$ and is therefore differentiable. Then, in [19], Dai, Bai, and Wei prove

$$[J(\mathbf{c})]_{ij} = \frac{\partial u_{i,nn}(\mathbf{c})}{\partial c_j} = \mathbf{e}_n^T L_i(\mathbf{c})^{-1} P_{\text{left},i} A_j P_{\text{right},i} (\mathbf{e}_n - I_{:,1:n-1} U_{i,11}(\mathbf{c})^{-1} U_{i,12}(\mathbf{c})), \quad (20)$$

where $\mathbf{e}_j$ ($j = 1, \dots, n$) are the j th column vectors of $I \in \mathbb{R}^{n \times n}$, and $I_{:,1:j} := [\mathbf{e}_1, \dots, \mathbf{e}_j] \in \mathbb{R}^{n \times j}$.

In the following, we derive the above Jacobi matrix in a different manner analogous to (6) in the previous section. To this end, we consider the LDU decompositions:

$$P_{\text{left},i}(A(\mathbf{c}) - \lambda_i^* I) P_{\text{right},i} = L_i(\mathbf{c}) D_i(\mathbf{c}) \tilde{U}_i(\mathbf{c}) \quad (i = 1, \dots, n), \quad (21)$$

where $L_i(\mathbf{c})$ ($i = 1, \dots, n$) are the unit lower triangular matrices, $D_i(\mathbf{c})$ ($i = 1, \dots, n$) are diagonal matrices, and $\tilde{U}_i(\mathbf{c})$ ($i = 1, \dots, n$) are the unit upper triangular matrices. Note that the diagonal elements of $D_i(\mathbf{c})$ ($i = 1, \dots, n$) are equal to those of $U_i(\mathbf{c})$ ($i = 1, \dots, n$) in (18). Hence, let

$$\mathbf{f}(\mathbf{c}) = [d_{1,nn}(\mathbf{c}), d_{2,nn}(\mathbf{c}), \dots, d_{n,nn}(\mathbf{c})]^T. \quad (22)$$

Analogous to (6), we see

$$L_i(\mathbf{c})^{-1} L_i(\mathbf{c}) = \tilde{U}_i(\mathbf{c}) \tilde{U}_i(\mathbf{c})^{-1} = I, \quad L_i(\mathbf{c})^{-1} P_{\text{left},i}(A(\mathbf{c}) - \lambda_i^* I) P_{\text{right},i} \tilde{U}_i(\mathbf{c})^{-1} = D_i(\mathbf{c}) \quad (23)$$

for $i = 1, \dots, n$. Using the second equation in (23), we compute the Jacobi matrix as

$$[J(\mathbf{c})]_{ij} = \frac{\partial d_{i,nn}(\mathbf{c})}{\partial c_j} = \mathbf{e}_n^T \frac{\partial L_i(\mathbf{c})^{-1} P_{\text{left},i}(A(\mathbf{c}) - \lambda_i^* I) P_{\text{right},i} \tilde{U}_i(\mathbf{c})^{-1}}{\partial c_j} \mathbf{e}_n,$$

where $\mathbf{e}_n$ is the n th column vector of $I \in \mathbb{R}^{n \times n}$. In the right-hand side, we see

$$\begin{aligned} \frac{\partial L_i(\mathbf{c})^{-1} P_{\text{left},i}(A(\mathbf{c}) - \lambda_i^* I) P_{\text{right},i} \tilde{U}_i(\mathbf{c})^{-1}}{\partial c_j} &= \frac{\partial L_i(\mathbf{c})^{-1}}{\partial c_j} P_{\text{left},i}(A(\mathbf{c}) - \lambda_i^* I) P_{\text{right},i} \tilde{U}_i(\mathbf{c})^{-1} \\ &\quad + L_i(\mathbf{c})^{-1} P_{\text{left},i} \frac{\partial A(\mathbf{c})}{\partial c_j} P_{\text{right},i} \tilde{U}_i(\mathbf{c})^{-1} \\ &\quad + L_i(\mathbf{c})^{-1} P_{\text{left},i}(A(\mathbf{c}) - \lambda_i^* I) P_{\text{right},i} \frac{\partial \tilde{U}_i(\mathbf{c})^{-1}}{\partial c_j} \\ &= L_i(\mathbf{c})^{-1} P_{\text{left},i} A_j P_{\text{right},i} \tilde{U}_i(\mathbf{c})^{-1} \\ &\quad + \frac{\partial L_i(\mathbf{c})^{-1}}{\partial c_j} L_i(\mathbf{c}) D_i(\mathbf{c}) + D_i(\mathbf{c}) U_i(\mathbf{c}) \frac{\partial \tilde{U}_i(\mathbf{c})^{-1}}{\partial c_j}. \end{aligned}$$

Since the diagonal elements of $L_i(\mathbf{c})^{-1}$ are 1 in the same manner as $L_i(\mathbf{c})$, the differentiation of the lower right element is 0. In other words, for $j = 1, \dots, n$, we have

$$[L_i(\mathbf{c})^{-1}]_{nn} = 1, \quad \left[\frac{\partial L_i(\mathbf{c})^{-1}}{\partial c_j} \right]_{nn} = 0 \quad (i = 1, \dots, n).$$

This yields

$$\mathbf{e}_n^T \left(\frac{\partial L_i(\mathbf{c})^{-1}}{\partial c_j} L_i(\mathbf{c}) \right) \mathbf{e}_n = 0 \quad (i = 1, \dots, n).$$

Similarly,

$$\mathbf{e}_n^T \left(\tilde{U}_i(\mathbf{c}) \frac{\partial \tilde{U}_i(\mathbf{c})^{-1}}{\partial c_j} \right) \mathbf{e}_n = 0 \quad (i = 1, \dots, n).$$

Therefore, we obtain

$$[J(\mathbf{c})]_{ij} = \mathbf{e}_n^T L_i(\mathbf{c})^{-1} P_{\text{left},i} A_j P_{\text{right},i} \tilde{U}_i(\mathbf{c})^{-1} \mathbf{e}_n. \quad (24)$$

This is mathematically equivalent to (20), as proved below. For $i = 1, \dots, n$, the upper triangle matrices can be divided into

$$\tilde{U}_i(\mathbf{c}) =: \begin{bmatrix} \tilde{U}_{i,11}(\mathbf{c}) & \tilde{U}_{i,12}(\mathbf{c}) \\ 0 & 1 \end{bmatrix}, \quad \tilde{U}_{i,11}(\mathbf{c}) \in \mathbb{R}^{(n-1) \times (n-1)}, \quad \tilde{U}_{i,12}(\mathbf{c}) \in \mathbb{R}^{(n-1)}. \quad (25)$$

From an easy calculation, we have

$$\tilde{U}_i(\mathbf{c})^{-1} =: \begin{bmatrix} \tilde{U}_{i,11}(\mathbf{c})^{-1} & -\tilde{U}_{i,11}(\mathbf{c})^{-1} \tilde{U}_{i,12}(\mathbf{c}) \\ 0 & 1 \end{bmatrix}. \quad (26)$$

In addition, noting

$$\begin{aligned} D_{i,11}(\mathbf{c}) &:= \text{diag}(d_{i,11}(\mathbf{c}), d_{i,22}(\mathbf{c}), \dots, d_{i,(n-1)(n-1)}(\mathbf{c})), \\ D_{i,11}(\mathbf{c}) \tilde{U}_{i,11}(\mathbf{c}) &= U_{i,11}(\mathbf{c}), \quad D_{i,11}(\mathbf{c}) \tilde{U}_{i,12}(\mathbf{c}) = U_{i,12}(\mathbf{c}), \end{aligned}$$

we see that (20) is equivalent to (24).

The Newton's method using the smooth LDU decompositions can be described as follows.

Algorithm 3 The Newton's method using the smooth LDU decompositions

---

Require: $\lambda_1^* < \dots < \lambda_n^*$, $A_0, A_1, \dots, A_n \in \mathbb{R}^{n \times n}$, $\mathbf{c}^{(0)} \in \mathbb{R}^n$.

- 1: Compute LDU: $L_i^{(0)} D_i^{(0)} \tilde{U}_i^{(0)} := P_{\text{left},i}^{(0)} (A(\mathbf{c}^{(0)}) - \lambda_i^* I) P_{\text{right},i}^{(0)}$ for all $1 \leq i \leq n$
  - 2: **for** $k := 0, 1, \dots$ **do**
  - 3: $[\mathbf{f}^{(k)}]_i = d_{i,nn}^{(k)}$ for all $1 \leq i \leq n$
  - 4: $[J^{(k)}]_{ij} = \mathbf{e}_n^T L_i^{(k)-1} P_{\text{left},i}^{(k)} A_j P_{\text{right},i}^{(k)} \tilde{U}_i^{(k)-1} \mathbf{e}_n$ for all $1 \leq i, j \leq n$
  - 5: $\mathbf{c}^{(k+1)} = \mathbf{c}^{(k)} - J^{(k)-1} \mathbf{f}^{(k)}$
  - 6:   Compute LDU: $L_i^{(k+1)} D_i^{(k+1)} \tilde{U}_i^{(k+1)} := P_{\text{left},i}^{(k+1)} (A(\mathbf{c}^{(k+1)}) - \lambda_i^* I) P_{\text{right},i}^{(k+1)}$ for all $1 \leq i \leq n$
  - 7: **end for**
-

From the algorithm design, we see quadratic convergence $\|\mathbf{c}^{(k+1)} - \mathbf{c}^*\| = \mathcal{O}(\|\mathbf{c}^{(k)} - \mathbf{c}^*\|^2)$ if the Jacobi matrices are nonsingular.

There exist other smooth matrix decompositions such as the QR decomposition and the singular value decomposition. The QR decomposition is applied to the inverse eigenvalue problems in [18, 28, 29]. For the application of the singular value decomposition, see [46] for the details. Although the differentiations of the smooth SVD in [46] can be computed in the same manner as the differentiations of (6), similar calculations of the smooth QR decompositions cannot be found in the literature. Thus, we present such calculations of differentiations below.

For a parameterized matrix $F(t)$, $t \in \mathbb{R}$, if $F(t)$ is nonsingular for some t , there exists the next matrix decomposition:

$$F(t) = Q(t) D(t) \tilde{U}(t), \quad D(t) = \text{diag}(d_{11}(t), \dots, d_{nn}(t)),$$

where $Q(t)$ is orthogonal, $d_{ii}(t) > 0$ ($i = 1, \dots, n$), and $\tilde{U}(t)$ is the unit upper triangular matrix. For the QR decomposition $F(t) = Q(t) R(t)$, we see $R(t) = D(t) \tilde{U}(t)$ in view of the uniqueness of the QR decomposition. Letting $\mathbf{e}_n$ denote the n th column vector of $I \in \mathbb{R}^{n \times n}$, we have

$$\begin{aligned} \frac{dd_{nn}(t)}{dt} &= \frac{d \mathbf{e}_n^T Q(t)^T F(t) \tilde{U}(t)^{-1} \mathbf{e}_n}{dt} \\ &= \mathbf{e}_n^T \left(\frac{dQ(t)^T}{dt} F(t) \tilde{U}(t)^{-1} + Q(t)^T \frac{dF(t)}{dt} \tilde{U}(t)^{-1} + Q(t)^T F(t) \frac{d\tilde{U}(t)^{-1}}{dt} \right) \mathbf{e}_n \\ &= \mathbf{e}_n^T \left(\frac{dQ(t)^T}{dt} Q(t) D(t) + Q(t)^T \frac{dF(t)}{dt} \tilde{U}(t)^{-1} + D(t) \tilde{U}(t) \frac{d\tilde{U}(t)^{-1}}{dt} \right) \mathbf{e}_n. \end{aligned}$$

For the above equation,

$$\mathbf{e}_n^T \left(\frac{dQ(t)^T}{dt} Q(t) D(t) \right) \mathbf{e}_n = 0, \quad \mathbf{e}_n^T \left(D(t) \tilde{U}(t) \frac{d\tilde{U}(t)^{-1}}{dt} \right) \mathbf{e}_n = 0$$

can be proved as follows. Firstly, using $Q(t)^T Q(t) = I$, we have

$$\frac{dQ(t)^T}{dt} Q(t) + Q(t)^T \frac{dQ(t)}{dt} = O.$$

This yields

$$\mathbf{e}_n^T \left(\frac{dQ(t)^T}{dt} Q(t) \right) \mathbf{e}_n = 0.$$

Since $D(t)$ is a diagonal matrix, we see

$$\mathbf{e}_n^T \left(\frac{dQ(t)^T}{dt} Q(t) D(t) \right) \mathbf{e}_n = 0.$$

Secondly, noting that the differentiation of the lower right element of $\tilde{U}(t)^{-1}$ is 0, we have

$$\mathbf{e}_n^T \left(D(t) \tilde{U}(t) \frac{d\tilde{U}(t)^{-1}}{dt} \right) \mathbf{e}_n = 0.$$

Thus, we obtain

$$\frac{dd_{nn}(t)}{dt} = \mathbf{e}_n^T \left(Q(t)^T \frac{dF(t)}{dt} \tilde{U}(t)^{-1} \right) \mathbf{e}_n.$$

Therefore, the Jacobi matrix can be computed, and thus the smooth QR decomposition is successfully applied to inverse eigenvalue problems [18, 28, 29].

4. Formulation by matrix equations

In this section, quadratically convergent methods are derived from formulations by matrix equations. The strength is to avoid solving eigenvalue problem per iteration. Moreover, n matrix decompositions are also avoided, unlike in the smooth matrix decomposition methods.

Firstly, let us see the Cayley transform method. In [24, Method III], a new approach is proposed with the use of the matrix exponential and the Cayley transform. Note that the solution of the inverse eigenvalue problem can be described as

$$X^{*T} A(\mathbf{c}^*) X^* = \Lambda^*, \quad (27)$$

where X^* is an orthogonal matrix. Let $\mathbf{c}^{(k)}$ and $X^{(k)}$ denote the current approximations of $\mathbf{c}^*$ and X^* respectively, where $X^{(k)}$ is assumed to be an orthogonal matrix. Let us write $X^* = X^{(k)} e^{Y^{(k)}}$, where $Y^{(k)}$ is a skew-symmetric matrix. Then, using (27) and the Taylor series of the exponential function, we have

$$X^{(k)T} A(\mathbf{c}^*) X^{(k)} = e^{Y^{(k)}} \Lambda^* e^{-Y^{(k)}} = \Lambda^* + Y^{(k)} \Lambda^* - \Lambda^* Y^{(k)} + \mathcal{O}(\|Y^{(k)}\|^2).$$

Similarly to the standard Newton's method, omitting the second order term in $Y^{(k)}$, we obtain

$$X^{(k)T} A(\mathbf{c}^{(k+1)}) X^{(k)} = \Lambda^* + \tilde{Y}^{(k)} \Lambda^* - \Lambda^* \tilde{Y}^{(k)}, \quad (28)$$

where $\tilde{Y}^{(k)}$ ($k = 0, 1, \dots$) are defined as skew-symmetric matrices in the same manner as $Y^{(k)}$ ($k = 0, 1, \dots$). We can calculate $\mathbf{c}^{(k+1)}$ by the diagonal elements of the equation above. More specifically, letting

$$\begin{aligned} [J^{(k)}]_{ij} &= \mathbf{x}_i^{(k)\top} A_j \mathbf{x}_i^{(k)} \quad (1 \leq i, j \leq n), \\ [\mathbf{d}^{(k)}]_i &= \lambda_i^* - \mathbf{x}_i^{(k)\top} A_0 \mathbf{x}_i^{(k)} \quad (1 \leq i \leq n) \end{aligned}$$

in the same manner as Algorithm 1, we see

$$\mathbf{c}^{(k+1)} = (J^{(k)})^{-1} \mathbf{d}^{(k)}.$$

Once $\mathbf{c}^{(k+1)}$ is obtained, from the off-diagonal elements of (28), we have

$$S^{(k+1)} = X^{(k)\top} A(\mathbf{c}^{(k+1)}) X^{(k)}, \quad [\tilde{Y}^{(k)}]_{ij} = -[\tilde{Y}^{(k)}]_{ji} = -\frac{[S^{(k+1)}]_{ij}}{\lambda_j^* - \lambda_i^*} \quad (1 \leq i, j \leq n, i \neq j). \quad (29)$$

Hence, the skew-symmetric matrix $\tilde{Y}^{(k)}$ can be obtained. Now construct an orthogonal matrix $Z^{(k)}$ using the Cayley transform

$$Z^{(k)} = (I + \tilde{Y}^{(k)}/2)(I - \tilde{Y}^{(k)}/2)^{-1} (\approx e^{\tilde{Y}^{(k)}}),$$

and compute $X^{(k+1)} = X^{(k)} Z^{(k)}$. This algorithm is the so-called Cayley transform method below.

Algorithm 4 The Cayley transform method [24, Method III].

Require: $\lambda_1^* < \dots < \lambda_n^*$, $A_0, \dots, A_n \in \mathbb{R}^{n \times n}$; an orthogonal matrix $X^{(0)} \in \mathbb{R}^{n \times n}$

```

1: for  $k = 0, 1, \dots$  do
2:    $[J^{(k)}]_{ij} = \mathbf{x}_i^{(k)\top} A_j \mathbf{x}_i^{(k)} \quad (1 \leq i, j \leq n)$ 
3:    $[\mathbf{d}^{(k)}]_i = \lambda_i^* - \mathbf{x}_i^{(k)\top} A_0 \mathbf{x}_i^{(k)} \quad (1 \leq i \leq n)$ 
4:    $\mathbf{c}^{(k+1)} = (J^{(k)})^{-1} \mathbf{d}^{(k)}$ 
5:    $S^{(k+1)} = X^{(k)\top} A(\mathbf{c}^{(k+1)}) X^{(k)}$ 
6:    $[\tilde{Y}^{(k)}]_{ii} = 0 \quad (1 \leq i \leq n)$ 
7:    $[\tilde{Y}^{(k)}]_{ij} = -[\tilde{Y}^{(k)}]_{ji} = -[S^{(k+1)}]_{ij} / (\lambda_j^* - \lambda_i^*) \quad (1 \leq i, j \leq n, i \neq j)$ 
8:    $X^{(k+1)} = X^{(k)} (I + \tilde{Y}^{(k)}/2)(I - \tilde{Y}^{(k)}/2)^{-1}$ 
9: end for
```

From the algorithm design, we expect that the convergence rate is $\|Y^{(k+1)}\| = \mathcal{O}(\|Y^{(k)}\|^2)$, where the mathematical analysis is summarized in the next section.

Algorithm 4 is effective and important in the research fields of inverse eigenvalue problems. From the mathematical view points, a geometric interpretation is introduced in [12, 14, 17]. To reduce the computational cost, [7, 8, 40] introduce some inexact solvers in line 4.

Next, we design an algorithm by a slightly different formulation using the following relations:

$$\begin{cases} X^{*\top} X^* = I \\ X^{*\top} A(\mathbf{c}^*) X^* = \Lambda^* \end{cases} \quad (30)$$

In addition, for a given $X^{(k)} \in \mathbb{R}^{n \times n}$, define $E^{(k)} \in \mathbb{R}^{n \times n}$ such that

$$X^{(k)} = X^* (I + E^{(k)}). \quad (31)$$

Firstly, noting $X^{*\top} X^* = I$ in (30), we obtain

$$I + E^{(k)} + E^{(k)\top} + \Delta_1^{(k)} = X^{(k)\top} X^{(k)}, \quad \Delta_1^{(k)} := E^{(k)\top} E^{(k)}. \quad (32)$$

Then, removing the quadratic term $\|\Delta_1^{(k)}\| \leq \|E^{(k)}\|^2$, we have the next matrix equation for $\tilde{E}^{(k)}$:

$$I + \tilde{E}^{(k)} + \tilde{E}^{(k)\top} = X^{(k)\top} X^{(k)}. \quad (33)$$

Secondly, noting $X^{*\top} A(\mathbf{c}^*) X^* = \Lambda^*$ in (30), we see

$$\Lambda^* + \Lambda^* E^{(k)} + E^{(k)\top} \Lambda^* + \Delta_2^{(k)} = X^{(k)\top} A(\mathbf{c}^*) X^{(k)}, \quad \Delta_2^{(k)} := E^{(k)\top} \Lambda^* E^{(k)}. \quad (34)$$

As in (33), omitting $\Delta_2^{(k)}$, we have the following equation for $\tilde{E}^{(k)}$ and $\mathbf{c}^{(k+1)}$:

$$\Lambda^* + \Lambda^* \tilde{E}^{(k)} + \tilde{E}^{(k)\top} \Lambda^* = X^{(k)\top} A(\mathbf{c}^{(k+1)}) X^{(k)}. \quad (35)$$

The above key matrix equations (33) and (35) are summarized as

$$\begin{cases} I + \tilde{E}^{(k)} + \tilde{E}^{(k)\top} = X^{(k)\top} X^{(k)} \\ \Lambda^* + \Lambda^* \tilde{E}^{(k)} + \tilde{E}^{(k)\top} \Lambda^* = X^{(k)\top} A(\mathbf{c}^{(k+1)}) X^{(k)} \end{cases}, \quad (36)$$

where the elements of $\tilde{E}^{(k)}$ and $\mathbf{c}^{(k+1)}$ are unknown variables. In fact, similarly to the Cayley transform method, all the unknown variables can be obtained by easy calculations.

First, the diagonal parts of the first equation in (36) indicate

$$[\tilde{E}^{(k)}]_{ii} = \frac{\mathbf{x}_i^{(k)\top} \mathbf{x}_i^{(k)} - 1}{2} \quad (1 \leq i \leq n).$$

Next, to compute $\mathbf{c}^{(k+1)}$, note that, in the second equation in (36),

$$[\Lambda^* + \Lambda^* \tilde{E}^{(k)} + \tilde{E}^{(k)\top} \Lambda^*]_{ii} = \lambda_i^* \mathbf{x}_i^{(k)\top} \mathbf{x}_i^{(k)} \quad (1 \leq i \leq n)$$

for the diagonal parts in the left hand-side. As for the right hand-side of the second equation in (36), letting

$$[J^{(k)}]_{ij} = \mathbf{x}_i^{(k)\top} A_j \mathbf{x}_i^{(k)} \quad (1 \leq i, j \leq n),$$

we see

$$[X^{(k)\top} A(\mathbf{c}^{(k+1)}) X^{(k)}]_{ii} = [J^{(k)} \mathbf{c}^{(k+1)}]_i + \mathbf{x}_i^{(k)\top} A_0 \mathbf{x}_i^{(k)} \quad (1 \leq i \leq n).$$

Therefore, letting

$$[\mathbf{d}^{(k)}]_i = \lambda_i^* \mathbf{x}_i^{(k)\top} \mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k)\top} A_0 \mathbf{x}_i^{(k)} \quad (1 \leq i \leq n),$$

we obtain

$$\mathbf{c}^{(k+1)} = (J^{(k)})^{-1} \mathbf{d}^{(k)}. \quad (37)$$

Finally, the off-diagonal parts of $\tilde{E}^{(k)}$ can be computed as follows. Since we have already obtained (37), for each (i, j) element of (36), we see the following 2×2 linear system

$$\begin{cases} [\tilde{E}^{(k)}]_{ij} + [\tilde{E}^{(k)}]_{ji} = \mathbf{x}_i^{(k)\top} \mathbf{x}_j^{(k)} \\ \lambda_i^* [\tilde{E}^{(k)}]_{ij} + \lambda_j^* [\tilde{E}^{(k)}]_{ji} = \mathbf{x}_i^{(k)\top} A(\mathbf{c}^{(k+1)}) \mathbf{x}_j^{(k)} \end{cases} \quad (1 \leq i, j \leq n, i \neq j).$$

Therefore, we obtain

$$[\tilde{E}^{(k)}]_{ij} = \frac{\lambda_j^* \mathbf{x}_i^{(k)\top} \mathbf{x}_j^{(k)} - \mathbf{x}_i^{(k)\top} A(\mathbf{c}^{(k+1)}) \mathbf{x}_j^{(k)}}{\lambda_j^* - \lambda_i^*} \quad (1 \leq i, j \leq n, i \neq j),$$

where we now assume all the prescribed eigenvalues are distinct as in (2).

As a result, since all the elements of $\tilde{E}^{(k)}$ are obtained, we can compute the next step

$$X^{(k+1)} = X^{(k)}(I - \tilde{E}^{(k)}),$$

where $I - \tilde{E}^{(k)}$ is the first order approximation of $(I + \tilde{E}^{(k)})^{-1}$ using the Neumann series. In Algorithm 5, we present the resulting algorithm.

From the algorithm design, we expect that the convergence rate is $\|E^{(k+1)}\| = \mathcal{O}(\|E^{(k)}\|^2)$, where the mathematical analysis is summarized in the next section.

The strengths of Algorithm 5 are summarized as follows:

- Algorithm 5 is primarily comprising matrix multiplications.

Algorithm 5 The matrix multiplication based method [1].

Require: $\lambda_1^* < \dots < \lambda_n^*$, $A_0, \dots, A_n \in \mathbb{R}^{n \times n}$; $X^{(0)} \in \mathbb{R}^{n \times n}$

```

1: for  $k = 0, 1, \dots$  do
2:    $R^{(k)} = X^{(k)\top} X^{(k)}$ 
3:    $[\tilde{E}^{(k)}]_{ii} = ([R^{(k)}]_{ii} - 1)/2 \quad (1 \leq i \leq n)$ 
4:    $[J^{(k)}]_{ij} = \mathbf{x}_i^{(k)\top} A_j \mathbf{x}_i^{(k)} \quad (1 \leq i, j \leq n)$ 
5:    $[\mathbf{d}^{(k)}]_i = \lambda_i^* [R^{(k)}]_{ii} - \mathbf{x}_i^{(k)\top} A_0 \mathbf{x}_i^{(k)} \quad (1 \leq i \leq n)$ 
6:    $\mathbf{c}^{(k+1)} = (J^{(k)})^{-1} \mathbf{d}^{(k)}$ 
7:    $S^{(k+1)} = X^{(k)\top} A(\mathbf{c}^{(k+1)}) X^{(k)}$ 
8:    $[\tilde{E}^{(k)}]_{ij} = (\lambda_j^* [R^{(k)}]_{ij} - [S^{(k+1)}]_{ij}) / (\lambda_j^* - \lambda_i^*) \quad (1 \leq i, j \leq n, i \neq j)$ 
9:    $X^{(k+1)} = X^{(k)}(I - \tilde{E}^{(k)})$ 
10: end for

```

- Although Algorithm 5 is similar to the Cayley transform method in Algorithm 4, Algorithm 5 does not require the Cayley transform, which reduces the operations in each iteration.
- The derivation of Algorithm 5 can be extended to inverse generalized eigenvalue problems in [18–20, 30], while the Cayley transform method cannot be applied to such problems.

More details of the strength are explained in [1, §3]. In 2017, Shen, Li, and Yao proposed a similar algorithm with an approximation of the Cayley transform in [42]. Their relation is clarified in [2, §2].

5. Convergence theory with discussions of multiple eigenvalues

---

In this section, we consider the situation where multiple eigenvalues are prescribed. For simplicity, we assume that the first p eigenvalues are multiple, i.e.,

$$\lambda_1^* = \dots = \lambda_p^* < \lambda_{p+1}^* < \dots < \lambda_n^*$$

in the same manner as [24, 40–42]. It is easy to generalize the following discussion to an arbitrary set of prescribed eigenvalues. In the following, we discuss convergence analysis for the algorithms in the previous sections.

5.1 Discussion of the Jacobi matrix

All of the existing quadratically convergent methods require to solve linear systems. In all the algorithms excluding Algorithm 3, the coefficient matrices are defined as

$$[J(X^{(k)})]_{ij} := \mathbf{x}_i^{(k)\top} A_j \mathbf{x}_i^{(k)} \quad (k = 0, 1, \dots),$$

where $X^{(k)} := [\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_n^{(k)}]$ for $k = 0, 1, \dots$ are approximated eigenvector matrices in the iterative methods. In this section, since we consider multiple eigenvalues, the orthogonal matrix X^* that satisfies (27) is not uniquely determined. Hence, for some $\mathbf{c}^*$, let

$$\mathcal{X}^* := \{X^* \mid X^{*\top} X^* = I, X^{*\top} A(\mathbf{c}^*) X^* = \Lambda^*\}.$$

Then, one may think

$$[J(X^*)]_{ij} := \mathbf{x}_i^{*\top} A_j \mathbf{x}_i^*, \quad \sup_{X^* \in \mathcal{X}^*} \|J(X^*)^{-1}\|$$

for the convergence analysis.

In the following, we mainly discuss Algorithm 1, while similar analysis is possible to other algorithms. In general, the above supremum does not exist as shown in [24, §3.1]. Since the convergence proof is not easy due to the above singularity of $J(X^*)$ for some X^* , the following inequalities

$$\|J(X^{(k)})^{-1}\| < \infty \quad (k = 0, 1, \dots), \quad \limsup_{k \rightarrow \infty} \|J(X^{(k)})^{-1}\| < \infty \quad (38)$$

are always assumed for the proof of the quadratic convergence also in recent papers [40–42]. The basic theorem is as follows.

Theorem 1 ([24, Theorem 3.2]). *Suppose that Algorithm 1 is applied to a set of prescribed eigenvalues: $\lambda_1^* = \dots = \lambda_p^* < \lambda_{p+1}^* < \dots < \lambda_n^*$. Assume that (38) holds in the iterative process. Then, $\|\mathbf{c}^{(k+1)} - \mathbf{c}^*\| = \mathcal{O}(\|\mathbf{c}^{(k)} - \mathbf{c}^*\|^2)$ in a neighborhood of $\mathbf{c}^*$.*

The above conditions as in (38) are not so restrictive because the set of the singular matrices are almost surely avoided in the iterative process. The quadratic convergence is usually observed in numerical tests even if multiple eigenvalues are present. The rate is generically quadratic both in theory and practice as in [24, §3.1].

From the above mathematical discussion, one may notice that, for some k , $X^{(k)}$ is not unique whenever $A(\mathbf{c}^{(k)})$ has multiple eigenvalues, while such a situation is rarely occurs. To analyze this case, a concept of relative generalized Jacobian matrices is introduced in [44]. On the basis of the relative generalized Jacobian matrices, the quadratic convergence is proved as follows. Let

$$\mathcal{D} = \{\mathbf{c} \in \mathbb{R}^n \mid \text{Eigenvalues of } A(\mathbf{c}) \text{ are all distinct}\}.$$

In addition, let $X^*(\mathbf{c}) = [\mathbf{x}_1^*(\mathbf{c}), \dots, \mathbf{x}_n^*(\mathbf{c})]$ denote an eigenvector matrix of $A(\mathbf{c})$ for $\mathbf{c} \in \mathcal{D}$. Moreover, let $[J^*(\mathbf{c})]_{ij} := \mathbf{x}_i^*(\mathbf{c})^T A_j \mathbf{x}_i^*(\mathbf{c})$ for all $1 \leq i, j \leq n$. Then, the relative generalized Jacobian matrices in [44] are defined as

$$\{J^* := \lim_{k \rightarrow \infty} J^*(\mathbf{c}^{(k)}) \text{ with } \{\mathbf{c}^{(k)}\}_{k=0,1,\dots} \subset \mathcal{D} \text{ and } \mathbf{c}^{(\infty)} = \mathbf{c}^*\}. \quad (39)$$

According to [40, 42, 44], quadratic convergence for some methods is proved if all of $\{J^*\}$ are nonsingular, where such assumptions appear to be almost surely satisfied. To verify this, [44] presents the next example:

$$A_0 = O, \quad A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \lambda_1^* = \lambda_2^* = 1, \quad (40)$$

where the solution is $\mathbf{c}^* = [1, 0]^T$. Obviously, $\mathcal{D} = \{\mathbf{c} \in \mathbb{R}^2 \mid c_2 \neq 0\}$. Thus, the relative Jacobian matrices $\{J^*\}$ are

$$\left\{ \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \right\},$$

which are nonsingular. Thus, in this case, Algorithm 1 is quadratically convergent.

Although Algorithm 1 is quadratically convergent for any eigenvalue set under mild assumptions, Algorithm 2 interpreted as a quasi-Newton method must be modified for the convergence due to the loss of the orthogonality of $X^{(k)}$ ($k = 0, 1, \dots$). More precisely, although the orthogonality of $[\mathbf{x}_{p+1}^{(k)}, \dots, \mathbf{x}_n^{(k)}]$ is refined per iteration with the aid of the shifted inverse iterations, $[\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_p^{(k)}]$ requires some care. In other words, after one step of the shifted inverse iteration:

$$\tilde{\mathbf{y}}_i^{(k+1)} = (A(\mathbf{c}^{(k+1)}) - \lambda_i^* I)^{-1} \mathbf{x}_i^{(k)} \quad (1 \leq i \leq n),$$

the QR decomposition of the matrix $\tilde{Y}_p^{(k+1)} := [\tilde{\mathbf{y}}_1^{(k+1)}, \dots, \tilde{\mathbf{y}}_p^{(k+1)}]$ is usually applied, namely

$$X_p^{(k+1)} R^{(k+1)} := Y_p^{(k+1)}, \quad X_p^{(k+1)} \in \mathbb{R}^{n \times p}, \quad R_p^{(k+1)} \in \mathbb{R}^{p \times p},$$

where $X_p^{(k+1)}$ is an orthogonal matrix, and $R^{(k+1)}$ is an upper triangular matrix. Let $\mathbf{x}_i^{(k+1)}$ be the i th column vectors of $X_p^{(k+1)}$ for $1 \leq i \leq p$. For the distinct eigenvalues, the updated vectors $\mathbf{x}_{p+1}^{(k+1)}, \dots, \mathbf{x}_n^{(k+1)}$ are simply $\mathbf{x}_i^{(k+1)} = \tilde{\mathbf{y}}_i^{(k+1)} / \|\tilde{\mathbf{y}}_i^{(k+1)}\|$ for $p+1 \leq i \leq n$ in the same manner as Algorithm 2.

In the actual computations, $A(\mathbf{c})$ is reduced to a tridiagonal matrix by the Householder transformation so as to reduce the arithmetic operations. The algorithm is presented below.

As stated before, the proof of quadratic convergence in [24, Theorem 3.3] is incorrect. In 2016, Shen, Li, and Yao have proved quadratic convergence of Algorithm 6 in [41] in the same manner as Theorem 1.

Algorithm 6 The quasi-Newton's method [24, Method II]

Require: $\lambda_1^* = \dots = \lambda_p^* < \lambda_{p+1}^* < \lambda_n^*$, $A_0, A_1, \dots, A_n \in \mathbb{R}^{n \times n}$, $\mathbf{c}^{(0)} \in \mathbb{R}^n$.

- 1: For $A(\mathbf{c}^{(0)})$, find the eigenpairs $(\lambda_i^{(0)}, \mathbf{x}_i^{(0)})$ for all $1 \leq i \leq n$
  - 2: **for** $k := 0, 1, \dots$ **do**
  - 3: $[J^{(k)}]_{ij} = \mathbf{x}_i^{(k)\top} A_j \mathbf{x}_i^{(k)}$
  - 4: $[\mathbf{d}^{(k)}]_i = \lambda_i^* - \mathbf{x}_i^{(k)\top} A_0 \mathbf{x}_i^{(k)}$
  - 5: $\mathbf{c}^{(k+1)} = J^{(k)^{-1}} \mathbf{d}^{(k)}$
  - 6:   Tridiagonalization by the Householder transformation: $U^{(k+1)\top} A(\mathbf{c}^{(k+1)}) U^{(k+1)} = T^{(k+1)}$
  - 7: $\mathbf{y}_i^{(k+1)} = (T^{(k+1)} - \lambda_i^* I)^{-1} U^{(k+1)\top} \mathbf{x}_i^{(k)}$ for all $1 \leq i \leq n$
  - 8: $\tilde{\mathbf{y}}_i^{(k+1)} = U^{(k+1)} \mathbf{y}_i^{(k+1)}$ for all $1 \leq i \leq n$
  - 9:   For $Y_p^{(k+1)} := [\tilde{\mathbf{y}}_1^{(k+1)}, \dots, \tilde{\mathbf{y}}_p^{(k+1)}]$, compute the QR decomposition: $X_p^{(k+1)} R^{(k+1)} := \tilde{Y}_p^{(k+1)}$
  - 10:   Let $\mathbf{x}_i^{(k+1)}$ be the i th column vectors of $X_p^{(k+1)}$ for $1 \leq i \leq p$
  - 11: $\mathbf{x}_i^{(k+1)} = \tilde{\mathbf{y}}_i^{(k+1)} / \|\tilde{\mathbf{y}}_i^{(k+1)}\|$ for $p+1 \leq i \leq n$
  - 12: **end for**
-

5.2 Use of block versions of the smooth matrix decompositions

For the multiple eigenvalues, smooth matrix decompositions can be extended to block versions. In [18], the QR-like decomposition as the block version is successfully applied to the inverse symmetric eigenvalue problems including multiple eigenvalues. According to [19, Remark 4.1], the block version of the smooth LU decomposition can be used in the same manner as the QR decomposition in [18]. For the readers' convenience, we derive the method based on the block version explicitly.

Similarly to (21), we compute the smooth LDU decompositions. For $i = 1$, let $L_1(\mathbf{c})$, $D_1(\mathbf{c})$, $\tilde{U}_1(\mathbf{c})$ be divided into

$$L_1(\mathbf{c}) =: \begin{bmatrix} L_{1,11}(\mathbf{c}) & O \\ L_{1,21}(\mathbf{c}) & L_{1,22}(\mathbf{c}) \end{bmatrix}, \quad (41)$$

$$L_{1,11}(\mathbf{c}) \in \mathbb{R}^{(n-p) \times (n-p)}, \quad L_{1,12}(\mathbf{c}) \in \mathbb{R}^{(n-p) \times p}, \quad L_{1,22}(\mathbf{c}) \in \mathbb{R}^{p \times p},$$

$$D_1(\mathbf{c}) =: \begin{bmatrix} D_{1,11}(\mathbf{c}) & O \\ O & D_{1,22}(\mathbf{c}) \end{bmatrix}, \quad (42)$$

$$D_{1,11}(\mathbf{c}) \in \mathbb{R}^{(n-p) \times (n-p)}, \quad D_{1,22}(\mathbf{c}) \in \mathbb{R}^{p \times p},$$

$$\tilde{U}_1(\mathbf{c}) =: \begin{bmatrix} \tilde{U}_{1,11}(\mathbf{c}) & \tilde{U}_{1,12}(\mathbf{c}) \\ O & \tilde{U}_{1,22}(\mathbf{c}) \end{bmatrix}, \quad (43)$$

$$\tilde{U}_{1,11}(\mathbf{c}) \in \mathbb{R}^{(n-p) \times (n-p)}, \quad \tilde{U}_{1,12}(\mathbf{c}) \in \mathbb{R}^{(n-p) \times p}, \quad \tilde{U}_{1,22}(\mathbf{c}) \in \mathbb{R}^{p \times p}.$$

Using the above definitions, we compute the block LDU decomposition:

$$P_{\text{left},1}(A(\mathbf{c}) - \lambda_1^* I) P_{\text{right},1} = \hat{L}_1(\mathbf{c}) \hat{D}_1(\mathbf{c}) \hat{U}_1(\mathbf{c}), \quad (44)$$

where $\hat{L}_1(\mathbf{c})$ and $\hat{U}_1(\mathbf{c})$ are defined as

$$\hat{L}_1(\mathbf{c}) = \begin{bmatrix} L_{1,11}(\mathbf{c}) & O \\ L_{1,21}(\mathbf{c}) & I_p \end{bmatrix}, \quad (45)$$

$$\hat{D}_1(\mathbf{c}) = \begin{bmatrix} D_{1,11}(\mathbf{c}) & O \\ O & \hat{D}_{1,22}(\mathbf{c}) \end{bmatrix}, \quad (46)$$

$$\hat{U}_1(\mathbf{c}) = \begin{bmatrix} \tilde{U}_{1,11}(\mathbf{c}) & \tilde{U}_{1,12}(\mathbf{c}) \\ O & I_p \end{bmatrix}, \quad (47)$$

$$\hat{D}_{1,22}(\mathbf{c}), I_p \in \mathbb{R}^{p \times p}.$$

More specifically, $\hat{D}_{1,22}(\mathbf{c})$ is uniquely determined by

$$\widehat{D}_{1,22}(\mathbf{c}) := I_{:, (n-p+1):n}^T P_{\text{left},1}(A(\mathbf{c}) - \lambda_1^* I) P_{\text{right},1} I_{:, (n-p+1):n} - L_{1,21}(\mathbf{c}) D_{1,11}(\mathbf{c}) U_{1,12}(\mathbf{c}),$$

where $I_{:, (n-p+1):n}$ is the last p column vectors of $I \in \mathbb{R}^{n \times n}$.

Note that, if $A(\mathbf{c})$ has an eigenvalue λ_1^* with p -fold degeneracy, the rank of $A(\mathbf{c}) - \lambda_1^* I$ is $n - p$ and therefore $\widehat{D}_{1,22}(\mathbf{c}) = O$. This is the idea of the algorithm using the block version of the smooth LU decomposition.

For the algorithm design, we consider the vectorization of a matrix. For any matrix $A = [\mathbf{a}_1, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times n}$, where $\mathbf{a}_i$ for $i = 1, \dots, n$ are the i th column vectors, we introduce the next linear operator

$$\text{vec}(A) = [\mathbf{a}_1^T, \dots, \mathbf{a}_n^T]^T \in \mathbb{R}^{mn}.$$

Using this operator, define

$$\mathbf{f}(\mathbf{c}) = [\text{vec}(\widehat{D}_{1,22}(\mathbf{c}))^T, d_{p+1,nn}(\mathbf{c}), \dots, d_{n,nn}(\mathbf{c})]^T \in \mathbb{R}^{n+p^2-p}$$

in the same manner as (22). Then, we have

$$\widehat{D}_{1,22}(\mathbf{c}) = I_{:, (n-p+1):n}^T \widehat{L}_1(\mathbf{c})^{-1} P_{\text{left},1}(A(\mathbf{c}) - \lambda_1^* I) P_{\text{right},1} \widehat{U}_1(\mathbf{c})^{-1} I_{:, (n-p+1):n}$$

in the same manner as (23), where

$$\widehat{L}_1(\mathbf{c})^{-1} = \begin{bmatrix} L_{1,11}(\mathbf{c})^{-1} & O \\ -L_{1,21}(\mathbf{c}) L_{1,11}(\mathbf{c})^{-1} & I_p \end{bmatrix}, \quad \widehat{U}_1(\mathbf{c})^{-1} = \begin{bmatrix} \widetilde{U}_{1,11}(\mathbf{c})^{-1} & -\widetilde{U}_{1,11}(\mathbf{c})^{-1} \widetilde{U}_{1,12}(\mathbf{c}) \\ O & I_p \end{bmatrix}. \quad (48)$$

Noting the above structures, we have

$$\frac{\partial \widehat{D}_{1,22}(\mathbf{c})}{\partial c_j} = I_{:, (n-p+1):n}^T \widehat{L}_1(\mathbf{c})^{-1} P_{\text{left},1}(\mathbf{c}) A_j P_{\text{right},1}(\mathbf{c}) \widehat{U}_1(\mathbf{c})^{-1} I_{:, (n-p+1):n}$$

in the same manner as (24). Thus, letting

$$J(\mathbf{c}) = \begin{bmatrix} \text{vec}\left(\frac{\partial \widehat{D}_{1,22}(\mathbf{c})}{\partial c_1}\right) & \cdots & \text{vec}\left(\frac{\partial \widehat{D}_{1,22}(\mathbf{c})}{\partial c_n}\right) \\ \frac{\partial \widehat{d}_{p+1,nn}(\mathbf{c})}{\partial c_1} & \cdots & \frac{\partial \widehat{d}_{p+1,nn}(\mathbf{c})}{\partial c_n} \\ \vdots & \cdots & \vdots \\ \frac{\partial \widehat{d}_{n,nn}(\mathbf{c})}{\partial c_1} & \cdots & \frac{\partial \widehat{d}_{n,nn}(\mathbf{c})}{\partial c_n} \end{bmatrix} \in \mathbb{R}^{(n+p^2-p) \times n}, \quad (49)$$

we have

$$\mathbf{0} = \mathbf{f}(\mathbf{c}^*) = \mathbf{f}(\mathbf{c}) + J(\mathbf{c})(\mathbf{c}^* - \mathbf{c}) + \mathcal{O}(\|\mathbf{c} - \mathbf{c}^*\|^2) \quad (50)$$

because $\mathbf{f}(\mathbf{c})$ is an analytic function. We see that $\mathbf{f}(\mathbf{c})$ is an overdetermined system and therefore the Gauss–Newton method should be used instead of the Newton method. The Gauss–Newton method is to produce a sequence $\{\mathbf{c}^{(k)}\}_{k=0,1,\dots}$ such that

$$\mathbf{c}^{(k+1)} = \arg \min_{\mathbf{c}^{(k+1)}} \|\mathbf{f}(\mathbf{c}^{(k)}) + J(\mathbf{c}^{(k)})(\mathbf{c}^{(k+1)} - \mathbf{c}^{(k)})\|,$$

resulting in the next iterative process

$$\mathbf{c}^{(k+1)} = \mathbf{c}^{(k)} - (J(\mathbf{c}^{(k)})^T J(\mathbf{c}^{(k)}))^{-1} J(\mathbf{c}^{(k)})^T \mathbf{f}(\mathbf{c}^{(k)}).$$

The resulting algorithm is described as follows.

Noting (50), we see the quadratic convergence $\|\mathbf{c}^{(k+1)} - \mathbf{c}^*\| = \mathcal{O}(\|\mathbf{c}^{(k)} - \mathbf{c}^*\|^2)$ as long as $\mathbf{c}^{(k)}$ ($k = 0, 1, \dots$) are in a neighborhood of $\mathbf{c}^*$ such that $J(\mathbf{c}^*)^T J(\mathbf{c}^*)$ is nonsingular, as stated in the next theorem.

Theorem 2 ([18, 19]). *Suppose that Algorithm 7 is applied to a set of prescribed eigenvalues: $\lambda_1^* = \dots = \lambda_p^* < \lambda_{p+1}^* < \dots < \lambda_n^*$. Assume that*

$$\|(J^{(k)})^T J^{(k)}\|^{-1} < \infty \quad (k = 0, 1, \dots), \quad \limsup_{k \rightarrow \infty} \|(J^{(k)})^T J^{(k)}\|^{-1} < \infty \quad (51)$$

in the iterative process, and $P_{\text{left},i}^{(k)} = P_{\text{left},i}^{(0)}$ and $P_{\text{right},i}^{(k)} = P_{\text{right},i}^{(0)}$ for all k . Then, $\|\mathbf{c}^{(k+1)} - \mathbf{c}^*\| = \mathcal{O}(\|\mathbf{c}^{(k)} - \mathbf{c}^*\|^2)$ in a neighborhood of $\mathbf{c}^*$.

Algorithm 7 The Newton's method using the smooth LDU decompositions

---

Require: $\lambda_1^* = \dots = \lambda_p^* < \lambda_{p+1}^* < \dots < \lambda_n^*$, $A_0, A_1, \dots, A_n \in \mathbb{R}^{n \times n}$, $\mathbf{c}^{(0)} \in \mathbb{R}^n$.

- 1: Compute block LDU: $\widehat{L}_i^{(0)} \widehat{D}_i^{(0)} \widehat{U}_i^{(0)} := P_{\text{left},i}^{(0)} (A(\mathbf{c}^{(0)}) - \lambda_i^* I) P_{\text{right},i}^{(0)}$ for all $1 \leq i \leq n$
  - 2: **for** $k := 0, 1, \dots$ **do**
  - 3: $\mathbf{f}^{(k)} = [\text{vec}(\widehat{D}_{1,22}^{(k)})^T, d_{p+1,nn}^{(k)}, \dots, d_{n,nn}^{(k)}]^T$
  - 4: $[J^{(k)}]_{ij} = [\text{vec}(I_{:, (n-p+1):n}^T \widehat{L}_1^{(k)-1} P_{\text{left},1}^{(k)} A_j P_{\text{right},1}^{(k)} \widehat{U}_1^{(k)-1} I_{:, (n-p+1):n})]_{ij}$ for $1 \leq i \leq p$, $1 \leq j \leq n$
  - 5: $[J^{(k)}]_{ij} = \mathbf{e}_n^T \widehat{L}_i^{(k)-1} P_{\text{left},i}^{(k)} A_j P_{\text{right},i}^{(k)} (\mathbf{e}_n - I_{:, 1:n-1} \widehat{U}_{i,11}^{(k)-1} \widehat{U}_{i,12}^{(k)})$ for $p+1 \leq i \leq n$, $1 \leq j \leq n$
  - 6: $\mathbf{c}^{(k+1)} = \mathbf{c}^{(k)} - (J^{(k)})^T J^{(k)} \mathbf{f}^{(k)}$
  - 7:   Compute block LDU: $\widehat{L}_i^{(k+1)} \widehat{D}_i^{(k+1)} \widehat{U}_i^{(k+1)} := P_{\text{left},i}^{(k+1)} (A(\mathbf{c}^{(k+1)}) - \lambda_i^* I) P_{\text{right},i}^{(k+1)}$ for all $1 \leq i \leq n$
  - 8: **end for**
-

5.3 Minimization approach with the use of formulations by matrix equations

We discuss here modifications of Algorithms 4 and 5 in Section 4 showing the formulations by matrix equations. The modifications are based on a kind of optimizations of the eigenvectors.

Firstly, we consider how to modify the Cayley transform method, presented in Algorithm 4 to adapt it to multiple eigenvalues. The remedy for the multiple eigenvalues is essentially related to the lack of the uniqueness of X^* . In other words,

$$A(\mathbf{c}) = X \Lambda^* X^T$$

can be considered a system of $n(n+1)/2$ equations in $n(n+1)/2$ unknowns, comprising the n dimensional vector $\mathbf{c}$ and the orthogonal matrix X which has $n(n-1)/2$ degrees of freedom. However, we now assume that it has $p(\geq 2)$ multiple eigenvalues, and thus $p(p-1)/2$ of these degrees of freedom are of no help in solving the above problem because they corresponds to the rotation of $[\mathbf{x}_1, \dots, \mathbf{x}_p]$. In other words, the above equation is inherently overdetermined. However, this do not cause difficulties of our discussion because we assume that a solution exists. Hence, we ignore the equations in the leading principal $p \times p$ matrices of $A(\mathbf{c}) = X \Lambda^* X^T$ and, for $k = 0, 1, \dots$, simply determine

$$[\widetilde{Y}^{(k)}]_{ij} = 0, \quad (1 \leq i, j \leq p) \quad (52)$$

in (28); see [24, §3.3] for its theoretical justification. We show later another theoretical justification based on the discussion of modifications of Algorithm 5 just after presenting Algorithm 9. In Algorithm 8, we present the modified version of Algorithm 4.

Algorithm 8 The Cayley transform method [24, Method III].

Require: $\lambda_1^* = \dots = \lambda_p^* < \lambda_{p+1}^* < \dots < \lambda_n^*$, $A_0, \dots, A_n \in \mathbb{R}^{n \times n}$; an orthogonal matrix $X^{(0)} \in \mathbb{R}^{n \times n}$

- 1: **for** $k = 0, 1, \dots$ **do**
  - 2: $[J^{(k)}]_{ij} = \mathbf{x}_i^{(k)T} A_j \mathbf{x}_i^{(k)} \quad (1 \leq i, j \leq n)$
  - 3: $[\mathbf{d}^{(k)}]_i = \lambda_i^* - \mathbf{x}_i^{(k)T} A_0 \mathbf{x}_i^{(k)} \quad (1 \leq i \leq n)$
  - 4: $\mathbf{c}^{(k+1)} = (J^{(k)})^{-1} \mathbf{d}^{(k)}$
  - 5: $S^{(k+1)} = X^{(k)T} A(\mathbf{c}^{(k+1)}) X^{(k)}$
  - 6: $[\widetilde{Y}^{(k)}]_{ij} = 0 \quad (1 \leq i, j \leq p)$
  - 7: $[\widetilde{Y}^{(k)}]_{ii} = 0 \quad (p+1 \leq i \leq n)$
  - 8: $[\widetilde{Y}^{(k)}]_{ij} = -[\widetilde{Y}^{(k)}]_{ji} = -[S^{(k+1)}]_{ij} / (\lambda_j^* - \lambda_i^*) \quad (p+1 \leq i \leq n, 1 \leq j \leq n, i > j)$
  - 9: $X^{(k+1)} = X^{(k)} (I + \widetilde{Y}^{(k)}/2)(I - \widetilde{Y}^{(k)}/2)^{-1}$
  - 10: **end for**
-

The proof of quadratic convergence is given in [24, Theorem 3.4].

Theorem 3 ([24]). *Suppose that Algorithm 8 is applied to a set of prescribed eigenvalues: $\lambda_1^* = \dots = \lambda_p^* < \lambda_{p+1}^* < \dots < \lambda_n^*$. Assume that*

$$\|J^{(k)}\|^{-1} < \infty \quad (k = 0, 1, \dots), \quad \limsup_{k \rightarrow \infty} \|J^{(k)}\|^{-1} < \infty \quad (53)$$

in the iterative process, and that, for a sufficiently small ϵ , $\|Y^{(0)}\| < \epsilon$ holds. Then, $\|Y^{(k+1)}\| = \mathcal{O}(\|Y^{(k)}\|^2)$.

Our last goal is to modify Algorithm 5 to solve the problems with multiple eigenvalues. This approach is based on an optimization problem, namely the orthogonal Procrustes problem [25, §6.4.1], described as follows. Recall that $X_p^* := [\mathbf{x}_1^*, \dots, \mathbf{x}_p^*]$ associated with the multiple eigenvalues is not unique. More specifically, for any orthogonal matrix $Q \in \mathbb{R}^{p \times p}$, all the columns of $X_p^* Q$ are also eigenvectors. Hence, for a given $X^{(k)}$, let

$$\bar{X}^{(k)} := \arg \min_{X^*} \|X^* - X^{(k)}\|_F, \quad (54)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. In addition, define $F^{(k)}$ that satisfies

$$X^{(k)} = \bar{X}^{(k)}(I + F^{(k)}). \quad (55)$$

Then, we see

$$\|F^{(k)}\|_F = \|X^{(k)} - \bar{X}^{(k)}\|_F = \min_{X^*} \|X^* - X^{(k)}\|_F \quad (56)$$

from easy calculations.

For a fixed $X^{(k)}$, the above $F^{(k)}$ is locally unique, and the leading principal $p \times p$ submatrix of $F^{(k)}$ is symmetric matrix. Such a feature is relevant to the orthogonal Procrustes problem [25, §6.4.1], solved by the polar decomposition. Noting the above symmetry, we have the next lemma.

Lemma 1. *For any fixed $X^{(k)}$, suppose that $\bar{X}^{(k)}$ and $F^{(k)}$ are defined as (54) and (55), respectively. Then, $F^{(k)}$ satisfies (56) and $[F^{(k)}]_{ij} = [F^{(k)}]_{ji}$ for $1 \leq i, j \leq p$ corresponding to multiple eigenvalues $\lambda_1^* = \dots = \lambda_p^*$.*

Our aim is to obtain $\tilde{F}^{(k)}$ approximating $F^{(k)}$ using the above lemma. Similarly to (32) and (34), we have

$$\begin{aligned} I + F^{(k)} + F^{(k)\top} + \Delta_1^{(k)} &= X^{(k)\top} X^{(k)}, \quad \Delta_1^{(k)} := F^{(k)\top} F^{(k)}. \\ \Lambda^* + \Lambda^* F^{(k)} + F^{(k)\top} \Lambda^* + \Delta_2^{(k)} &= X^{(k)\top} A(\mathbf{c}^*) X^{(k)}, \quad \Delta_2^{(k)} := F^{(k)\top} \Lambda^* F^{(k)}. \end{aligned}$$

Noting the diagonal elements, we have the following linearized equations:

$$\begin{cases} \left[I + \tilde{F}^{(k)} + \tilde{F}^{(k)\top} \right]_{ii} = [X^{(k)\top} X^{(k)}]_{ii} \\ \left[\Lambda^* + \Lambda^* \tilde{F}^{(k)} + \tilde{F}^{(k)\top} \Lambda^* \right]_{ii} = [X^{(k)\top} A(\mathbf{c}^{(k+1)}) X^{(k)}]_{ii} \end{cases}$$

for $i = 1, \dots, n$ in the same manner as Algorithm 5. The linearization for the off-diagonal elements is slightly different. For $i \neq j$ corresponding to $\lambda_i^* = \lambda_j^*$, on the basis of the above symmetry of $F^{(k)}$ in Lemma 1, let

$$\begin{cases} \left[I + \tilde{F}^{(k)} + \tilde{F}^{(k)\top} \right]_{ij} = [X^{(k)\top} X^{(k)}]_{ij} \\ \left[\tilde{F}^{(k)} \right]_{ij} = \left[\tilde{F}^{(k)} \right]_{ji} \end{cases}.$$

On the other hand, for $i \neq j$ corresponding to $\lambda_i^* \neq \lambda_j^*$, let

$$\begin{cases} \left[I + \tilde{F}^{(k)} + \tilde{F}^{(k)\top} \right]_{ij} = [X^{(k)\top} X^{(k)}]_{ij} \\ \left[\Lambda^* + \Lambda^* \tilde{F}^{(k)} + \tilde{F}^{(k)\top} \Lambda^* \right]_{ij} = [X^{(k)\top} A(\mathbf{c}^{(k+1)}) X^{(k)}]_{ij} \end{cases}$$

in the same manner as Algorithm 5. We present the algorithm below.

This algorithm gives another interpretation of exceptional handling (52) in the Cayley transform method (Algorithm 8) as follows. In Algorithm 9, if we assume that, for some k , given $X^{(k)}$ is an orthogonal matrix, we see

---

Algorithm 9 The matrix multiplication based method [2].

Require: $\lambda_1^* = \dots = \lambda_p^* < \lambda_{p+1}^* < \dots < \lambda_n^*$, $A_0, \dots, A_n \in \mathbb{R}^{n \times n}$; $X^{(0)} \in \mathbb{R}^{n \times n}$

```

1: for  $k = 0, 1, \dots$  do
2:    $R^{(k)} = X^{(k)\top} X^{(k)}$ 
3:    $[\tilde{F}^{(k)}]_{ii} = ([R^{(k)}]_{ii} - 1)/2 \quad (1 \leq i \leq n)$ 
4:    $[J^{(k)}]_{ij} = \mathbf{x}_i^{(k)\top} A_j \mathbf{x}_i^{(k)} \quad (1 \leq i, j \leq n)$ 
5:    $[\mathbf{d}^{(k)}]_i = \lambda_i^* [R^{(k)}]_{ii} - \mathbf{x}_i^{(k)\top} A_0 \mathbf{x}_i^{(k)} \quad (1 \leq i \leq n)$ 
6:    $\mathbf{c}^{(k+1)} = (J^{(k)})^{-1} \mathbf{d}^{(k)}$ 
7:    $S^{(k+1)} = X^{(k)\top} A(\mathbf{c}^{(k+1)}) X^{(k)}$ 
8:   if  $1 \leq i \leq p$  and  $1 \leq j \leq p$  then
9:      $[\tilde{F}^{(k)}]_{ij} = [R^{(k)}]_{ij}/2 \quad (i \neq j)$ 
10:  else
11:     $[\tilde{F}^{(k)}]_{ij} = (\lambda_j^* [R^{(k)}]_{ij} - [S^{(k+1)}]_{ij})/(\lambda_j^* - \lambda_i^*) \quad (i \neq j)$ 
12:  end if
13:   $X^{(k+1)} = X^{(k)}(I - \tilde{F}^{(k)})$ 
14: end for

```

$$[\tilde{F}^{(k)}]_{ij} = 0, \quad (1 \leq i, j \leq p)$$

in the same manner as (52). In other words, to determine $[\tilde{Y}^{(k)}]_{ij} = 0$, $(1 \leq i, j \leq p)$ as in (52) can be interpreted as an approximation of $F^{(k)}$ in Lemma 1, though the Cayley transform is applied to $\tilde{Y}^{(k)}$.

In the following, we discuss the convergence theory. Define $\bar{J}$ as $[\bar{J}]_{ij} = \bar{\mathbf{x}}_i^{(0)\top} A_j \bar{\mathbf{x}}_i^{(0)}$ $(1 \leq i, j \leq n)$ in the same manner as line 4 in Algorithm 9. Using $\bar{J}$, we introduce

$$\alpha := \sqrt{n} \|\bar{J}^{-1}\| \sqrt{\sum_{\ell=1}^n \|A_\ell\|^2}. \quad (57)$$

The quadratic convergence is theoretically guaranteed as follows.

Theorem 4 ([2]). *Suppose that Algorithm 9 is applied to a set of the prescribed eigenvalues: $\lambda_1^* = \dots = \lambda_p^* < \lambda_{p+1}^* < \dots \leq \lambda_n^*$. In addition, define $F^{(k)}$ ($k = 0, 1, \dots$) as in Lemma 1. Moreover, suppose that $F^{(0)}$ satisfies*

$$\|F^{(0)}\| \leq \frac{\min_{\lambda_i^* \neq \lambda_j^*} |\lambda_i^* - \lambda_j^*|}{18n \|\Lambda^*\| (1 + \alpha)}, \quad (58)$$

where α is defined as (57). Then, we obtain

$$\|F^{(k)}\| \leq C_1^k \|F^{(0)}\| \quad (k = 0, 1, \dots) \quad (59)$$

$$\limsup_{k \rightarrow \infty} \frac{\|F^{(k+1)}\|}{\|F^{(k)}\|^2} \leq 3 \left(\frac{2n \|\Lambda^*\| (1 + C_2 \alpha)}{\min_{\lambda_i^* \neq \lambda_j^*} |\lambda_i^* - \lambda_j^*|} + 1 \right), \quad (60)$$

where $C_1 = 0.469 \dots$, $C_2 = 1.309 \dots$.

This theorem is essentially different from the ones for other algorithms because the condition on the quadratic convergence can be written by the initial values at $k = 0$. Roughly speaking, if $X^{(0)}$ is close to some X^* associated with some well conditioned $J(X^*)$, then $\{X^{(k)}\}_{k=0}^\infty$ should converge to some neighborhood of X^* . From such a perspective, we first construct an optimal matrix $\bar{X}^{(k)} \in \{X^*\}$ depending on $X^{(k)}$ as in Lemma 1. Next, we design the algorithm as $X^{(k+1)}$ ($k = 0, 1, \dots$) approximate $\bar{X}^{(k)}$ ($k = 0, 1, \dots$). Thus, if $\|F^{(0)}\|$ is sufficiently small compared with $\|J(\bar{X}^{(0)})^{-1}\|^{-1}$ as in (58), quadratic convergence is guaranteed as in Theorem 4. In other words, the assumption on initial values as in (58) is sufficient for the convergence, resulting in Theorem 4 that ensures quadratic convergence. This formulation is the crucial strength of Algorithm 9.

With Theorem 4 in mind, let us examine the theory of the relative Jacobian matrices expressed by (39) in more detail. There is a case which is not covered by the theory of the relative Jacobian matrices. A simple example is the following inverse eigenvalue problem:

$$A_0 = O, \quad A_1 = \begin{bmatrix} 1 & 0 \\ 0 & t \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \lambda_1^* = \lambda_2^* = 0,$$

according to [2, §4]. The solution vector is $\mathbf{c}^* = [0, 0]^T$. If $t = 1$, the relative Jacobians are the same as the previous example as in (40). Otherwise, $\mathcal{D} = \{\mathbf{c} \in \mathbb{R}^2 \mid \mathbf{c} \neq \mathbf{c}^*\}$. Thus, for $\mathbf{c} = [c_1, 0]^T$ ($c_1 \neq 0$), $\{J^*(\mathbf{c})\}$ are

$$\left\{ \begin{bmatrix} 1 & 0 \\ t & 0 \end{bmatrix}, \begin{bmatrix} t & 0 \\ 1 & 0 \end{bmatrix} \right\}, \quad (61)$$

which implies that some relative generalized Jacobian matrices are singular. Hence, to consider any eigenvector matrix, let

$$\mathbf{q}_1 = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}, \quad \mathbf{q}_2 = \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \end{bmatrix}, \quad Q = [\mathbf{q}_1, \mathbf{q}_2].$$

Noting

$$J^*(Q) := \begin{bmatrix} \mathbf{q}_1^T A_1 \mathbf{q}_1 & \mathbf{q}_1^T A_2 \mathbf{q}_1 \\ \mathbf{q}_2^T A_1 \mathbf{q}_2 & \mathbf{q}_2^T A_2 \mathbf{q}_2 \end{bmatrix} = \begin{bmatrix} \cos^2(\theta) + t \sin^2(\theta) & 2 \cos(\theta) \sin(\theta) \\ t \cos^2(\theta) + \sin^2(\theta) & -2 \cos(\theta) \sin(\theta) \end{bmatrix},$$

we see

$$\det(J^*(Q)) = -2(1+t) \cos(\theta) \sin(\theta).$$

If $t = -1$, then $J^*(Q)$ is always singular. Otherwise, in general, $J^*(Q)$ is nonsingular, unless $\cos(\theta) \sin(\theta) = 0$ associated with $\mathbf{c} = [c_1, 0]^T$ ($c_1 \neq 0$). Thus, since $J^*(Q)$ for $\mathbf{c} = [c_1, c_2]^T$ ($c_2 \neq 0$) is nonsingular, the convergence from such a direction $\mathbf{c}$ is expected. However, some relative Jacobian matrices are singular as in (61). The above assumption for the relative Jacobian matrices $\{J^*\}$ in (39) is too strong to ensure (38) for some fixed $\{X^{(k)}\}_{k=0}^\infty$.

From the mathematical viewpoint, the actual convergence behavior should be verified by the initial matrix $X^{(0)}$ and its target solution, such as $\bar{X}^{(0)} \in \{X^*\}$. In this sense, Theorem 4 is straightforward and convincing because it ensures the quadratic convergence under the assumption on initial values as in (58) with no assumption for $k \geq 1$.

6. Influence on other research fields

From the mathematical viewpoint, there are two recent developments in other research fields extremely relevant to the progress in the quadratically convergent methods for inverse symmetric eigenvalue problems.

6.1 Iterative refinement for symmetric eigenvalue problems

In 1986, Davies and Modi proposed an iterative algorithm [22] for symmetric eigenvalue problems based on the same idea as the Cayley transform method (Algorithm 4). The Davies–Modi’s algorithm is based on

$$X^T A X = \Lambda,$$

where A is a symmetric matrix, X is a normalized eigenvector matrix, and Λ is a diagonal matrix whose diagonal elements are the eigenvalues of A . The idea is also seen in Jahn’s method, explained in [45] by Wilkinson. In [24], however, Friedland, Nocedal, and Overton did not point out such a relationship. As in Algorithm 4, the Davies–Modi’s algorithm requires an orthogonal matrix $\hat{X}$ approximating X .

In each iteration, the Cayley transform is indispensable to ensure the orthogonality of the refined matrix.

Recently, Ogita and Aishima have proposed a new iterative refinement algorithm [34] based on the following matrix equations:

$$\begin{cases} X^T X = I & (\text{orthogonality}) \\ X^T A X = \Lambda & (\text{diagonality}) \end{cases}$$

corresponding to Algorithm 5 for inverse eigenvalue problems. More precisely, Algorithm 5 in [1] is inspired by the idea in [34]. In addition, the relationship between the Davies–Modi’s algorithm and the Cayley transform method (Algorithm 4) is pointed out in [1] for the first time.

6.2 Verification of the solutions by the Newton method

One of the most important applications of Newton’s methods is to prove that a solution exists in a particular region with the aid of the interval arithmetic. In other words, we can obtain an interval including the exact solution using a test interval given by a numerical solution. The interval analysis and its effectiveness are clearly explained in [33]. The principle to the existence proof is the Brouwer’s fixed point theorem. In this section, we explain Krawczyk’s operator [26] as in Moor’s paper [32] and use it to computationally verify the existence of a solution in a trial region. As it stands, only Algorithm 1 is applied to the verified computing. The use of other algorithms is future work.

Let $\mathbb{IR}$ denote the set of intervals in $\mathbb{R}$. In addition, any real interval vector is written as

$$[\mathbf{c}] = [c_1, \dots, c_n]^T := [[\underline{c}_1, \bar{c}_1], \dots, [\underline{c}_n, \bar{c}_n]]^T \in \mathbb{IR}^n, \quad \underline{c}_i \leq c_i \leq \bar{c}_i \ (i = 1, \dots, n).$$

Define an operator P as $P(\mathbf{c}) := \mathbf{c} - J(\mathbf{c})^{-1} \mathbf{f}(\mathbf{c})$. Note that $\mathbf{c}^{(k+1)} = P(\mathbf{c}^{(k)})$ ($k = 0, 1, \dots$) is the Newton’s method. If we have an interval vector $[\mathbf{c}]$ such that $P([\mathbf{c}]) \subseteq [\mathbf{c}]$ in the exact arithmetic, P has a fixed point $\mathbf{c}^* \in [\mathbf{c}]$ in view of Brouwer’s fixed point theorem. Then, $\mathbf{c}^* = P(\mathbf{c}^*)$ implies $\mathbf{f}(\mathbf{c}^*) = \mathbf{0}$ whenever $J(\mathbf{c}^*)$ is nonsingular. Hence, we might be able to verify the solution by the interval arithmetic. In fact, this iteration with the interval Gaussian algorithm for $J([\mathbf{c}])^{-1} \mathbf{f}([\mathbf{c}])$ is applied to the inverse eigenvalue problems [6], and then verified enclosures are obtained for a couple of problems.

However, we would like to avoid the interval arithmetic for the Gaussian elimination. To this end, Krawczyk’s operator is usually applied [5, 32]. The Krawczyk’s operator K is defined as

$$K([\mathbf{c}], \mathbf{c}, G) = \mathbf{c} - G \mathbf{f}(\mathbf{c}) + (I - G J([\mathbf{c}])([\mathbf{c}] - \mathbf{c}) \in \mathbb{IR}^n,$$

where G is any nonsingular matrix, approximating $J(\mathbf{c})^{-1}$ for a given $\mathbf{c}$. If we find an interval $[\mathbf{c}]$ such that $K([\mathbf{c}], \mathbf{c}, G) \subseteq [\mathbf{c}]$, there exists $\mathbf{c}^* \in [\mathbf{c}]$ such that $\mathbf{f}(\mathbf{c}^*) = \mathbf{0}$; see [32, Theorem 1] or [33, Theorem 8.2] for the proof. The test interval $[\mathbf{c}]$ can be determined based on the Newton–Kantorovich theorem as in [5].

In 1994, the Krawczyk’s operator K is successfully applied to inverse eigenvalue problems [4]. The feature of the verification procedure of the inverse eigenvalue problems is to use verification techniques for eigenvalue problems because to compute the nonlinear function and the Jacobi matrix requires the eigenpairs. Recently, an efficient algorithm is designed with a modern sophisticated verification technique for eigenvalue problems [31].

7. Concluding remarks

In this paper, we surveyed recent progress in the quadratically convergent methods for inverse symmetric eigenvalue problems. They are well organized in the context of the classical quadratically convergent methods. The modern effective methods are classified into three versions:

- (a) Direct formulation using eigenvalues as in Section 2
- (b) Use of determinant equations and smooth matrix decompositions as in Section 3

(c) Formulation by matrix equations as in Section 4

Among them, (a) and (b) are flexible and easily applied to other complicated inverse eigenvalue problems, and (c) is the most efficient, while it is basically restricted to inverse symmetric eigenvalue problems. For clarifying the relationship between (a) and (b), we derived the method using smooth LDU decomposition as in (b), differentiating the parametrized matrices in the same manner as in (a).

Although we considered inverse symmetric eigenvalue problems for simplicity, the formulations by (a) and (b) can be applied to nonsymmetric matrices, generalized eigenvalue problems, and so forth [9, 18–21, 23, 46]. The formulation of (c) is restrictive. However, the matrix multiplication based method can be applied to generalized symmetric eigenvalue problems [1, §5], which is modified to cover multiple eigenvalues in [3]. The Cayley transform method is successfully applied to the inverse singular value problems [12].

In Section 5, we considered multiple eigenvalues, and then discussed the convergence properties of the various quadratically convergent methods. More specifically, the nonsingularities of the Jacobi matrices are investigated. As for (b), we explicitly designed the algorithm using the block LDU decompositions. It is worth noting that Theorem 4 for the matrix multiplication based method in (c) ensures the quadratic convergence in a neighborhood of the solution such that the Jacobi matrix is nonsingular. To obtain similar convergence theorems for other methods appears to be valuable future work.

From the mathematical viewpoint, the quadratically convergent methods discussed in this paper are extremely relevant to recent progress in other research fields as described in Section 6. Algorithms 5 and 9 are inspired by the idea of iterative refinement algorithm for symmetric eigenvalue problems in [34, 35]. In [35], an efficient technique for nearly multiple eigenvalues is proposed, and thus a similar approach might be useful to inverse eigenvalue problems. As for the verified computing, the Krawczyk’s operator based on Algorithm 1 can be successfully applied to verification of the existence of the solution of inverse symmetric eigenvalue problems [4, 31].

Acknowledgments

This study was supported by KAKENHI Grant Number 17K14143 and 17H02826. The author thanks to the referee for the valuable comments.

References

- [1] K. Aishima, “A quadratically convergent algorithm based on matrix equations for inverse eigenvalue problems,” *Linear Algebra Appl.*, vol. 542, pp. 310–333, 2018.
- [2] K. Aishima, “A quadratically convergent algorithm for inverse eigenvalue problems with multiple eigenvalues,” *Linear Algebra Appl.*, vol. 549, pp. 30–52, 2018.
- [3] K. Aishima, “A quadratically convergent algorithm for inverse generalized eigenvalue problems,” *J. Comput. Appl. Math.*, vol. 367, pp. 112485, 2020.
- [4] G. Alefeld, A. Gienger, and G. Mayer, “Numerical validation for an inverse matrix eigenvalue problem,” *Computing*, vol. 53, pp. 311–322, 1994.
- [5] G. Alefeld, A. Gienger, and F. Potra, “Efficient numerical validation of solutions of nonlinear systems,” *SIAM J. Numer. Anal.*, vol. 31, pp. 252–260, 1994.
- [6] G. Alefeld and G. Mayer, “A computer aided existence and uniqueness proof for an inverse matrix eigenvalue problem,” *Interval Computat.*, vol. 1, pp. 4–27, 1994.
- [7] Z. Bai, R. Chan, and B. Morini, “An inexact Cayley transform method for inverse eigenvalue problems,” *Inverse Probl.*, vol. 20, pp. 1675–1689, 2004.
- [8] Z. Bai, “Inexact Newton methods for inverse eigenvalue problems,” *Appl. Math. Comput.*, vol. 172, pp. 682–689, 2006.
- [9] F. Biegler-König, “A Newton iteration process for inverse eigenvalue problems,” *Numer. Math.*, vol. 37, pp. 349–354, 1981.
- [10] R. Chan, H. Chung, and S. Xu, “The inexact Newton-like method for inverse eigenvalue problem,” *BIT*, vol. 43, pp. 7–20, 2003.

- [11] R. Chan, S. Xu, and H. Zhou, "On the convergence rate of a quasi-Newton method for inverse eigenvalue problem," *SIAM J. Numer. Anal.*, vol. 36, pp. 436–441, 1999.
- [12] M. Chu, "Numerical methods for inverse singular value problems," *SIAM J. Numer. Anal.*, vol. 29, pp. 885–903, 1992.
- [13] M. Chu, "Matrix differential equations: a continuous realization process for linear algebra problems," *Nonlinear Anal.*, vol. 18, pp. 1125–1146, 1992.
- [14] M. Chu, "Inverse eigenvalue problems," *SIAM Rev.*, vol. 40, pp. 1–39, 1998.
- [15] M. Chu and K. Driessel, "The projected gradient method for least squares matrix approximations with spectral constraints," *SIAM J. Numer. Anal.*, vol. 27, pp. 1050–1060, 1990.
- [16] M. Chu and G. Golub, "Structured inverse eigenvalue problems," *Acta Numer.*, vol. 11, pp. 1–71, 2002.
- [17] M. Chu and G. Golub, "Inverse eigenvalue problems, Theory, Algorithms and Applications," Oxford University Press, New York, 2005.
- [18] H. Dai, "An algorithm for symmetric generalized inverse eigenvalue problems," *Linear Algebra Appl.*, vol. 296, pp. 79–98, 1999.
- [19] H. Dai, Z. Bai, and W. Ying, "On the solvability condition and numerical algorithm for the parameterized generalized inverse eigenvalue problem," *SIAM J. Matrix Anal. Appl.*, vol. 36, pp. 707–726, 2015.
- [20] H. Dai and P. Lancaster, "Newton's method for a generalized inverse eigenvalue problem," *Numer. Linear Algebra Appl.*, vol. 4, pp. 1–21, 1997.
- [21] N. Datta and V. Sokolov, "A solution of the affine quadratic inverse eigenvalue problem," *Linear Algebra Appl.*, vol. 434, pp. 1745–1760, 2011.
- [22] R. Davies and J. Modi, "A direct method for completing eigenproblem solutions on a parallel computer," *Linear Algebra Appl.*, vol. 77, pp. 61–74, 1986.
- [23] S. Elhay and Y. Ram, "An affine inverse eigenvalue problem," *Inverse probl.*, vol. 18, pp. 455–466, 2002.
- [24] S. Friedland, J. Nocedal, and M. Overton, "The formulation and analysis of numerical methods for inverse eigenvalue problems," *SIAM J. Numer. Anal.*, vol. 24, pp. 634–667, 1987.
- [25] G.H. Golub and C.F. Van Loan, "Matrix Computations, 4th ed.," The Johns Hopkins University Press, Baltimore, 2013.
- [26] R. Krawczyk, "Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken," *Computing*, vol. 4, pp. 187–201, 1969.
- [27] P. Lancaster, "Algorithm for lambda-matrices," *Numer. Math.*, vol. 6, pp. 388–394, 1964.
- [28] R.-C. LI, "Compute multiply nonlinear eigenvalues," *J. Comput. Math.*, vol. 10, pp. 1–20, 1992.
- [29] R.-C. LI, "Algorithms for inverse eigenvalue problems," *J. Comput. Math.*, vol. 10, pp. 97–111, 1992.
- [30] W. Ma, "A backward error for the symmetric generalized inverse eigenvalue problem," *Linear Algebra Appl.*, vol. 464, pp. 90–99, 2015.
- [31] S. Miyajima, "Verified solutions of inverse symmetric eigenvalue problems," *Reliab. Comput.*, vol. 24, pp. 31–44, 2017.
- [32] R. Moore, "A test for existence of solutions to nonlinear systems," *SIAM J. Numer. Anal.*, vol. 14, pp. 611–615, 1977.
- [33] R. Moore, B. Kearfott, and M. Cloud, "Introduction to interval analysis," SIAM, Philadelphia, PA, 2009.
- [34] T. Ogita and K. Aishima, "Iterative refinement for symmetric eigenvalue decomposition," *Jpn. J. Ind. Appl. Math.*, vol. 35, pp. 1007–1035, 2018.
- [35] T. Ogita and K. Aishima, "Iterative refinement for symmetric eigenvalue decomposition II: clustered eigenvalues," *Jpn. J. Ind. Appl. Math.*, vol. 36, pp. 435–459, 2019.
- [36] T. Ogita, S. Rump, and S. Oishi, "Accurate sum and dot product," *SIAM J. Sci. Comput.*, vol. 26, pp. 1955–1988, 2005.
- [37] K. Ozaki, T. Ogita, S. Oishi, and S. Rump, "Error-free transformations of matrix multiplication by using fast routines of matrix multiplication and its applications," *Numer. Algorithms*, vol. 59,

- pp. 95–118, 2012.
- [38] B.N. Parlett, “The symmetric eigenvalue problem, 2nd ed.,” *Classics in Applied Mathematics*, vol. 20, SIAM, Philadelphia, 1998.
  - [39] W. Shen, C. Li, and X. Jin, “A Ulm-like method for inverse eigenvalue problems,” *Appl. Numer. Math.*, vol. 61, pp. 356–367, 2011.
  - [40] W. Shen, C. Li, and X. Jin, “An inexact Cayley transform method for inverse eigenvalue problems with multiple eigenvalues,” *Inverse Probl.*, vol. 31, 085007, 2015.
  - [41] W. Shen, C. Li, and J. Yao, “Convergence analysis of Newton-like methods for inverse eigenvalue problems with multiple eigenvalues,” *SIAM J. Numer. Anal.*, vol. 54, pp. 2938–2950, 2016.
  - [42] W. Shen, C. Li, and J. Yao, “Approximate Cayley transform methods for inverse eigenvalue problems and convergence analysis,” *Linear Algebra Appl.*, vol. 523, pp. 187–219, 2017.
  - [43] J. Sun, “Backward errors for the inverse eigenvalue problem,” *Numer. Math.*, vol. 82, pp. 339–349, 1999.
  - [44] D. Sun and J. Sun, “Strong semismoothness of symmetric matrices and its application to inverse eigenvalue problems,” *SIAM J. Numer. Anal.*, vol. 40, pp. 2352–2367, 2003.
  - [45] J.H. Wilkinson, “The algebraic eigenvalue problem,” Clarendon Press, Oxford, 1965.
  - [46] S. Xu, “A smallest singular value method for solving inverse eigenvalue problems,” *J. Comput. Math.*, vol. 14, pp. 23–31, 1996.