East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.2023-234.231223

Vol. **15**, No. 1, pp. 185-204
February 2025

On the Parameterized Polynomial Inverse Eigenvalue Problem

Mei-Ling Xiang[1,*] and Hua Dai[2]

[1] *School of Mathematics and Physics, Nanjing Institute of Technology, Nanjing 210016, China.*
[2] *School of Mathematics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China.*

Received 25 September 2023; Accepted (in revised version) 23 December 2023.

Abstract. The paper focuses on the solvability and computability of the parameterized polynomial inverse eigenvalue problem (PPIEP). Employing multiparameter eigenvalue problems, we establish a sufficient solvability condition for the PPIEP. Three numerical methods are used to solve PPIEPs. The first one is the Newton method based on locally smooth QR-decomposition with the column pivoting and the second the Newton method based on the smallest singular value. In order to reduce the computational cost of computing the smallest singular values and the corresponding unit left and right singular vectors in each iteration, we approximate these values by using one-step inverse iterations. Subsequently, we introduce another method — viz. a Newton-like method based on the smallest singular value. Each of three methods exhibits locally quadratic convergence under appropriate conditions. Numerical examples demonstrate the effectiveness of the methods proposed.

AMS subject classifications: 65F18, 15A29
Key words: Polynomial inverse eigenvalue problem, multiparameter eigenvalue problem, Newton method, Newton-like method.

1. Introduction

Over the years, there has been significant discourse surrounding inverse eigenvalue problems, primarily driven by their wide-ranging applications. The applications encompass various fields, including but not limited to dynamic behavior control of damped mass-spring systems [22], structural design [27], finite element model updating [10, 11], and circuit theory [34]. Corresponding to a wide range of applications, there exist diverse categories of inverse eigenvalue problems — e.g. parameterized, structured, partially described, and others [9]. In this work, we study a particular inverse eigenvalue problem — viz. the parameterized polynomial inverse eigenvalue problem (PPIEP), which is determined as follows.

*Corresponding author. *Email addresses:* `xiangmeiling@nuaa.edu.cn` (M.-L. Xiang), `hdai@nuaa.edu.cn` (H. Dai)

Problem 1.1 (PPIEP)**.** Let $A_m \in \mathbf{R}^{n\times n}$ be a non-singular matrix and $\{A_j^{(q)}\}_{j=0}^{mn} \in \mathbf{R}^{n\times n}$, $q = 0,1,\ldots,m-1$. For any distinct complex numbers $\lambda_1,\lambda_2,\cdots,\lambda_{mn}$, find a vector $c = (c_1,c_2,\cdots,c_{mn})^T \in \mathbf{R}^{mn}$ or $\mathbf{C}^{mn}$ such that the polynomial eigenvalue problem

$$\left[\lambda^m A_m + \lambda^{m-1}A_{m-1}(c) + \cdots + \lambda A_1(c) + A_0(c)\right]x = 0$$

with the matrices

$$A_q(c) = A_0^{(q)} + \sum_{j=1}^{mn} c_j A_j^{(q)}, \quad q = 0,1,\ldots,m-1 \tag{1.1}$$

has the eigenvalues $\lambda_1,\lambda_2,\cdots,\lambda_{mn}$.

The following notations will be used throughout the discussion. The sets of all real and complex $m\times n$ matrices are respectively denoted by $\mathbf{R}^{m\times n}$ and $\mathbf{C}^{m\times n}$ and we write $\mathbf{R}^m$ for $\mathbf{R}^{m\times 1}$ and $\mathbf{C}^m$ for $\mathbf{C}^{m\times 1}$. Besides, $\mathscr{U}_n$ and $\mathscr{R}_n$ refer to the sets of all $n\times n$ unitary and all $n\times n$ upper-triangular matrices, respectively. In addition, I is the identity matrix of a suitable size, e_j the j-th column vector of I, and $\|\cdot\|$ represents the Euclidean vector norm or the induced matrix norm. Moreover, if A is an $m\times n$ matrix, then A^T and $\sigma_{\min}(A)$ respectively denote the transpose and the smallest singular value of A.

The PPIEP is a class of problems. It contains the parameterized standard inverse eigenvalue problem (PSIEP) — i.e. if $m=1$, $A_m = I$, the parameterized generalized inverse eigenvalue problem (PGIEP) — i.e. if $m=1$, $A_m = A_1(c)$, and the parameterized quadratic inverse eigenvalue problem (PQIEP) — i.e. if $m=2$. The theory and applications of these problems have been extensively investigated in the past decades. The reader can consult [6,7,9,24,41,42] for PSIEP studies, [1,12,15,16,18–20,27,31] for PGIEP studies, and [21–23,38,39] for PQIEP studies. However, to the best of our knowledge, there is a lack of pertinent information regarding high-order PPIEPs. Similar to all inverse eigenvalue problems, there are two fundamental questions concerning the PPIEP associated problems — viz. their solvability and computability.

The solvability analysis of PSIEP, PGIEP, and PQIEP is believed to be beneficial in analyzing the solvability of the PPIEP, due to the unique relationship that exists among these four problems. Additionally, the numerical methods for solving PSIEP, PGIEP and PQIEP can contribute to the development of numerical methods for solving the PPIEP. Hence, we take PQIEP as an illustrative example to analyze the solvability of PSIEP, PGIEP and PQIEP, as well as the existing numerical techniques employed for solving these problems.

For the solvability issue, Xiang and Dai [38] presented a sufficient condition for the existence of a solution of PQIEP by using the theory of multiparameter eigenvalue problem [3]. Using the same theory, Ji [26] and Dai *et al.* [15] proposed sufficient conditions for the existence of solutions of PSIEP and PGIEP earlier.

For the computability issue, most of the numerical algorithms with locally quadratic convergence (under appropriate conditions) are frequently constructed by formulating an equivalent nonlinear system and then employing Newton method to solve it. Based on the determinant evaluations proposed by Lancaster [29] and Biegler-König [6] and further analyzed by Friedland *et al*. [24], Elhay and Ram [22] constructed the following nonlinear system equivalent to PQIEP:

$$d(c)=\begin{pmatrix} \det\left(\lambda_1^2A_2+\lambda_1A_1(c)+A_0(c)\right)\\ \det\left(\lambda_2^2A_2+\lambda_2A_1(c)+A_0(c)\right)\\ \vdots\\ \det\left(\lambda_{2n}^2A_2+\lambda_{2n}A_1(c)+A_0(c)\right)\end{pmatrix}=0, \tag{1.2}$$

and applied Newton method to solve it. Biegler-König [6] and Song and Dai [36] used a nonlinear system similar to (1.2) to study numerical methods for PSIEP and PGIEP, respectively. The Newton method based on the determinant evaluation — i.e. on the nonlinear system (1.2), is capable of handling both symmetric and asymmetric PQIEPs. However, this method is not computationally attractive [39] and may suffer from ill-conditioning [24].

Suppose that c^* is a solution of PQIEP and $\lambda_1(c),\lambda_2(c),\cdots,\lambda_{2n}(c)$ are the eigenvalues of the quadratic pencil $Q(\lambda,c)=\lambda^2A_2+\lambda A_1(c)+A_0(c)$. It is natural to consider solving PQIEP by its equivalent nonlinear system

$$l(c)=\begin{pmatrix} \lambda_1(c)-\lambda_1\\ \lambda_2(c)-\lambda_2\\ \vdots\\ \lambda_{2n}(c)-\lambda_{2n}\end{pmatrix}=0. \tag{1.3}$$

Since the given eigenvalues are distinct, the eigenvalues $\{\lambda_i(c)\}_{i=1}^{2n}$ of $Q(\lambda,c)$ are distinct and differentiable around c^* [2]. It allows to use the Newton method to solve the nonlinear system (1.3). To ensure that the Newton method can solve (1.3), it is needed to reorder the eigenvalues in a suitable way. Since the eigenvalues of the quadratic pencil $Q(\lambda,c)$ are complex in general, the right ordering of the eigenvalues is not a trivial task. Elhay and Ram [23] developed a Newton method based on (1.3) for solving the symmetric PQIEP under the assumption that the number of real and complex eigenvalues in each iteration remains the same as the number of real and complex eigenvalues prescribed. However, without such an assumption, the ordering problem of the eigenvalues was not addressed. Datta and Sokolov [21] solved the matching problem of the eigenvalues by Hungarian method [28], and presented a Newton method for solving the symmetric PQIEP. Based on the nonlinear system similar to (1.3), Friedland *et al*. [24] developed a Newton method and two Newton-like methods for solving the symmetric PSIEP. Since then, considerable literatures have been devoted to solve the symmetric PSIEP by various Newton-like methods [1], inexact Newton methods [4] and inexact Newton-like methods [5,8,35]. Dai and Lancaster [16], Aishima [1], and Dalvand *et al*. [19,20] extended the Newton or Newton-like methods to the symmetric PGIEP. The Newton method based on the eigenvalues — i.e. on the nonlinear system (1.3), is highly effective. However, it is only applicable to the symmetric PQIEP.

Based on the locally smooth QR-decomposition with column pivoting for a matrix-valued function depending on several parameters [17], Xiang and Dai [38] formulated another nonlinear system equivalent to PQIEP

$$r(c)=\begin{pmatrix} r_{nn}^{(1)}(c) \\ r_{nn}^{(2)}(c) \\ \vdots \\ r_{nn}^{(2n)}(c) \end{pmatrix}=0, \tag{1.4}$$

where $r_{nn}^{(i)}(c)$ is obtained by the QR-decomposition with the column pivoting of $Q(\lambda_i,c)=\lambda_i^2A_2+\lambda_iA_1(c)+A_0(c)$, i.e.

$$Q(\lambda_i,c)\Pi_i(c)=Q_i(c)R_i(c), \quad i=1,2,\ldots,2n,$$

where $\Pi_i(c)$ is a permutation matrix, $Q_i(c)\in\mathscr{U}_n$, and

$$R_i(c)=\begin{pmatrix} R_{11}^{(i)}(c) & R_{12}^{(i)}(c) \\ 0 & r_{nn}^{(i)}(c) \end{pmatrix}, \quad R_{11}^{(i)}(c)\in\mathscr{R}_{n-1}.$$

Furthermore, they developed a Newton method for solving PQIEP. Based on the nonlinear system similar to (1.4), Dai [12] proposed a Newton method for solving the symmetric PGIEP. The Newton method based on the smooth QR-decomposition — i.e. on the nonlinear system (1.4), is applicable to both symmetric and asymmetric PQIEP. However, it is computationally expensive to compute mn QR-decompositions with column pivoting in each iteration.

Since a matrix is singular if and only if its smallest singular value is equal to zero, Xiang and Dai [39] reformulated the PQIEP as the following nonlinear system:

$$s(c)=\begin{pmatrix} \sigma_{\min}\left(\lambda_1^2A_2+\lambda_1A_1(c)+A_0(c)\right) \\ \sigma_{\min}\left(\lambda_2^2A_2+\lambda_2A_1(c)+A_0(c)\right) \\ \vdots \\ \sigma_{\min}\left(\lambda_{2n}^2A_2+\lambda_{2n}A_1(c)+A_0(c)\right) \end{pmatrix}=0, \tag{1.5}$$

and proposed a Newton and a Newton-like methods for PQIEP. Using a nonlinear system similar to (1.5), Xu [40] introduced a Newton and a Newton-like methods for PSIEP. Song and Dai [36] further extended the Newton method to solve PGIEP. A Newton and a Newton-like methods based on the smallest singular values — i.e. on the nonlinear system (1.5), demonstrate efficiency in solving both symmetric and asymmetric PQIEP [39].

Accordingly, we first transform the PPIEP into a multiparameter eigenvalue problem and then apply the theorem on the multiparameter eigenvalue problem to analyze the solvability of the PPIEP. Taking into account the advantages and disadvantages of the numerical methods mentioned above, we employ a Newton method based on the smooth QR-decomposition with column pivoting and a Newton and a Newton-like methods based on the smallest singular value to solve the PPIEP. These methods can handle both symmetric and asymmetric cases.

The remainder of the paper is organized as follows. In Section 2, we reduce the PPIEP to an equivalent multiparameter eigenvalue problem, and provide a sufficient condition, which guarantees the solvability of the PPIEP. In Section 3, we present a Newton method based on the smooth QR-decomposition with column pivoting for solving the PPIEP and its locally quadratic convergence. In Section 4, we derive a Newton and a Newton-like methods based on smallest singular value for solving the PPIEP and show the locally quadratic convergence of these methods. In Section 5, the performance of the methods is demonstrated through numerical experiments. Finally, a few conclusions are drawn in Section 6.

Throughout this paper, we assume that the given eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_{mn}$ are distinct and the solution to the PPIEP is always denoted by c^*. To ensure that the number of free parameters does not degenerate, we also assume that the mn matrices $[A_j^{(0)}, A_j^{(1)}, \cdots, A_j^{(m-1)}]$, $j = 1, 2, \ldots, mn$ are linearly independent in the space of $n \times mn$ matrices.

2. The Solvability of the PPIEP

In order to discuss the solvability of the PPIEP, we briefly recall the theory of the multiparameter eigenvalue problem [3]. Let $\mathbf{H}_r$ denote the Hilbert space $\mathbf{R}^{n_r}$ or $\mathbf{C}^{n_r}$ and $A_{r,s}$ be linear operators on $\mathbf{H}_r$, $r = 1, 2, \ldots, l$, $s = 0, 1, \ldots, l$. The multiparameter eigenvalue problem is to find scalars $\mu_0, \mu_1, \cdots, \mu_l$ that are not all zero and nonzero vectors $x_r \in \mathbf{C}^{n_r}$, $r = 1, 2, \ldots, l$ such that

$$\begin{aligned}
&\mu_0 A_{1,0} x_1 + \mu_1 A_{1,1} x_1 + \cdots + \mu_l A_{1,l} x_1 = 0,\\
&\mu_0 A_{2,0} x_2 + \mu_1 A_{2,1} x_2 + \cdots + \mu_l A_{2,l} x_2 = 0,\\
&\cdots\ \cdots\\
&\mu_0 A_{l,0} x_l + \mu_1 A_{l,1} x_l + \cdots + \mu_l A_{l,l} x_l = 0.
\end{aligned} \tag{2.1}$$

The $(l+1)$-tuple $\mu = (\mu_0, \mu_1, \cdots, \mu_l)$ and the vector $x = x_1 \otimes x_2 \otimes \cdots \otimes x_l$ are called the eigenvalue and the corresponding eigenvector, respectively. Obviously, $\mu = (\mu_0, \mu_1, \cdots, \mu_l)$ is an eigenvalue of (2.1) if and only if $\ker(\sum_{s=0}^{l} \mu_s A_{r,s}) \neq \{0\}$ or $\det(\sum_{j=0}^{l} \mu_s A_{r,s}) = 0$, $r = 1, 2, \ldots, l$.

Let $\mathbf{H} = \mathbf{H}_1 \otimes \mathbf{H}_2 \otimes \cdots \otimes \mathbf{H}_l$ be the tensor product space. For a decomposable tensor $x = x_1 \otimes x_2 \otimes \cdots \otimes x_l \in \mathbf{H}$, $A_{r,s}^{\dagger}$ denote the operators on $\mathbf{H}$ induced by $A_{r,s}$ and are defined as

$$\begin{aligned}
&A_{r,s}^{\dagger}(x_1 \otimes \cdots \otimes x_l)\\
&= x_1 \otimes \cdots \otimes x_{r-1} \otimes A_{r,s} x_r \otimes x_{r+1} \otimes \cdots \otimes x_l, \quad r = 1, 2, \ldots, l, \quad s = 0, 1, \ldots, l.
\end{aligned} \tag{2.2}$$

These induced operators possess a significant property of commutativity, meaning that they can be interchanged without affecting the outcome, i.e., $A_{r,s}^{\dagger} A_{k,s}^{\dagger} = A_{k,s}^{\dagger} A_{r,s}^{\dagger}$, $r \neq k$, $r, k = 1, 2, \ldots, l$, $s = 0, 1, \ldots, l$.

Solving the multiparameter eigenvalue problem (2.1) requires the use of operator-valued determinants defined as follows:

$$\Delta_s = (-1)^s \det \begin{pmatrix} A_{1,0}^{\dagger} & \cdots & \widehat{A}_{1,s}^{\dagger} & \cdots & A_{1,l}^{\dagger} \\ \vdots & & \vdots & & \vdots \\ A_{l,0}^{\dagger} & \cdots & \widehat{A}_{l,s}^{\dagger} & \cdots & A_{l,l}^{\dagger} \end{pmatrix}, \quad s = 0, 1, \ldots, l, \tag{2.3}$$

where the caret (†) indicates omission, which means that Δ_s obtained by deleting the $(s+1)$-th column. The right-hand side of (2.3) can be expanded in the usual manner for determinants, in which the products of the entries are interpreted as the composites of the operators concerned.

As shown in [3], assuming $\Delta = \sum_{s=0}^{m} \gamma_s \Delta_s$ is nonsingular for some scalars $\gamma_0, \gamma_1, \cdots, \gamma_l$, the problem (2.1) is equivalent to the following joint eigenvalue problem:

$$\Delta^{-1}\Delta_s x = \mu_s x, \quad s = 0, 1, \ldots, l. \tag{2.4}$$

Theorem 2.1 (cf. Atkinson [3]). *If there are scalars $\gamma_0, \gamma_1, \cdots, \gamma_l$ such that $\Delta = \sum_{s=0}^{m} \gamma_s \Delta_s$ is nonsingular, then for an eigenvalue $\mu = (\mu_0, \mu_1, \cdots, \mu_l)$ of the problem* (2.1)*, we have*

$$\sum_{s=0}^{l} \gamma_s \mu_s \neq 0.$$

Up to a scaling factor, the eigenvalues of the problem (2.1) *are the simultaneous eigenvalues of the joint eigenvalue problem* (2.4)*. Moreover, all operators $\Delta^{-1}\Delta_s$, $s = 1, 2, \ldots, l$ mutually commute — i.e.*

$$\Delta_r \Delta^{-1} \Delta_s = \Delta_s \Delta^{-1} \Delta_r, \quad r, s = 1, 2, \ldots, l.$$

Now we discuss the solvability of the PPIEP. According to the definition of the PPIEP, if the PPIEP has a solution $c \in \mathbf{C}^{mn}$, then there are nonzero vectors $x_i \in \mathbf{C}^n$ such that

$$\left[\lambda_i^m A_m + \lambda_i^{m-1} A_{m-1}(c) + \cdots + \lambda_i A_1(c) + A_0(c)\right] x_i = 0, \quad i = 1, 2, \ldots, mn. \tag{2.5}$$

Let

$$\begin{aligned} A_{i,0} &= \lambda_i^m A_m + \lambda_i^{m-1} A_0^{(m-1)} + \cdots + A_0^{(0)}, \\ A_{i,j} &= \lambda_i^{m-1} A_j^{(m-1)} + \lambda_i^{m-2} A_j^{(m-2)} + \cdots + A_j^{(0)}, \end{aligned} \quad i, j = 1, 2, \ldots, mn. \tag{2.6}$$

It is straightforward to verify that (2.5) is equivalent to the multiparameter eigenvalue problem

$$\begin{aligned} &A_{1,0} x_1 + c_1 A_{1,1} x_1 + \cdots + c_{mn} A_{1,mn} x_1 = 0, \\ &A_{2,0} x_2 + c_1 A_{2,1} x_2 + \cdots + c_{mn} A_{2,mn} x_2 = 0, \\ &\cdots\ \cdots \\ &A_{mn,0} x_{mn} + c_1 A_{mn,1} x_{mn} + \cdots + c_{mn} A_{mn,mn} x_{mn} = 0, \end{aligned} \tag{2.7}$$

where $c \in \mathbf{C}^{mn}$ and $x = x_1 \otimes x_2 \otimes \cdots \otimes x_{mn}$ are the eigenvalue and corresponding eigenvector of (2.7). So, the following result is obtained.

Theorem 2.2. *If the given eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_{mn}$ are distinct, then $c = (c_1, c_2, \cdots, c_{mn})^T$ is a solution of the PPIEP if and only if it is an eigenvalue of the multiparameter eigenvalue problem* (2.7).

Theorem 2.2 demonstrates that the analysis of the solvability of the PPIEP can be reduced to the analysis of the solvability of the multiparameter eigenvalue problem (2.7).

According to the definition (2.2) and the properties of the Kronecker product [13], we represent the operators $A^{\dagger}_{i,j}$ induced by $A_{i,j}$ in (2.6) as

$$A^{\dagger}_{i,j} = I \otimes \cdots \otimes I \otimes A_{i,j} \otimes I \otimes \cdots \otimes I, \quad i = 1,2,\ldots,mn, \quad j = 0,1,\ldots,mn.$$

Besides, the operator-valued determinants of the system (2.7) are defined as follows:

$$T_j = (-1)^j \det \begin{pmatrix} A^{\dagger}_{1,0} & \cdots & \widehat{A}^{\dagger}_{1,j} & \cdots & A^{\dagger}_{1,mn} \\ \vdots & & \vdots & & \vdots \\ A^{\dagger}_{mn,0} & \cdots & \widehat{A}^{\dagger}_{mn,j} & \cdots & A^{\dagger}_{mn,mn} \end{pmatrix}, \quad j = 0,1,\ldots,mn,$$

where $\widehat{A}^{\dagger}_{i,j}$ means the omission of $A^{\dagger}_{i,j}$, $i = 1,2,\ldots,mn$.

Since the coefficients of $A_{i,0}$, $i = 1,2,\ldots,mn$ in (2.7) are all equal to 1, the approach of [3] and Theorem 2.1 lead to the following theorem.

Theorem 2.3. *If the given eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_{mn}$ are distinct, then the solutions of the PPIEP coincide with the simultaneous eigenvalues of the following joint eigenvalue problem:*

$$(T_1 - c_1 T_0)x = 0, \quad (T_2 - c_2 T_0)x = 0, \quad \cdots, \quad (T_{mn} - c_{mn} T_0)x = 0.$$

From Theorems 2.1 and 2.3, we can derive a sufficient condition for the solvability of the PPIEP.

Theorem 2.4. *If the given eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_{mn}$ are distinct and T_0 is invertible, then the PPIEP has at least one solution.*

Proof. The proof of this result is quite similar to that of [15, Theorem 2.4] and is omitted here. □

3. Newton Method Based on Smooth *QR*-Decomposition

We briefly recall some conclusions on the smooth QR-decomposition [17, 30], which are helpful in our derivation. Let $B(c) = (b_{st}(c)) \in \mathbf{C}^{n\times n}$ be a twice continuously differentiable matrix-valued function defined on an open connected domain $\mathbf{D} \subseteq \mathbf{C}^l$. The twice differentiability of $B(c)$ with respect to $c = (c_1, c_2, \cdots, c_l)^T$ means that for any $c^{(0)} = (c_1^{(0)}, c_2^{(0)}, \cdots, c_l^{(0)})^T \in \mathbf{D}$ the partial derivatives $\partial b_{st}(c)/\partial c_i$, $s,t = 1,\ldots,n$, $i = 1,\ldots,l$ exist and

$$B(c) = B(c^{(0)}) + \sum_{i=1}^{l} \frac{\partial B(c^{(0)})}{\partial c_i}\left(c_i - c_i^{(0)}\right) + \mathscr{O}\left(\|c - c^{(0)}\|_2^2\right),$$

where

$$\frac{\partial B(c^{(0)})}{\partial c_i} = \left(\left.\frac{\partial b_{st}(c)}{\partial c_i}\right|_{c=c^{(0)}} \right) \in \mathbf{C}^{n\times n}.$$

The following results are related to the existence of a locally smooth QR-decomposition of $B(c)$.

Theorem 3.1 (cf. Xiang and Dai [38]). *Let $B(c) \in \mathbf{C}^{n\times n}$ be twice continuously differentiable at $c^{(0)} \in \mathbf{D} \subseteq \mathbf{C}^l$. Assume that $\mathrm{rank}(B(c^{(0)})) \geq n-1$, $\Pi \in \mathbf{C}^{n\times n}$ is a permutation matrix such that the first $n-1$ columns of $B(c^{(0)})\Pi$ are linearly independent, and $B(c^{(0)})\Pi$ has a QR-decomposition*

$$B(c^{(0)})\Pi = Q_0 R_0,$$

where $Q_0 \in \mathscr{U}_n$,

$$R_0 = \begin{pmatrix} R_{11} & R_{12} \\ 0 & r_{nn} \end{pmatrix},$$

and $R_{11} \in \mathscr{R}_{n-1}$ is a nonsingular matrix. Then there exists a neighborhood $\mathscr{N}(c^{(0)}) \subset \mathbf{D}$ of $c^{(0)}$ such that for all $c \in \mathscr{N}(c^{(0)})$, the matrix-valued function $B(c)\Pi$ has the QR-decomposition

$$B(c)\Pi = Q(c)R(c),$$

where $Q(c) \in \mathscr{U}_n$,

$$R(c) = \begin{pmatrix} R_{11}(c) & R_{12}(c) \\ 0 & r_{nn}(c) \end{pmatrix},$$

and $R_{11}(c) \in \mathscr{R}_{n-1}$. Moreover, this QR-decomposition has the following properties:

(1) $Q(c^{(0)}) = Q_0$ and $R(c^{(0)}) = R_0$.

(2) All elements of $Q(c)$ and $R(c)$ are continuous in $\mathscr{N}(c^{(0)})$.

(3) The diagonal elements of $R(c)$ are continuously differentiable at $c^{(0)}$, and

$$r_{nn}(c) = r_{nn} + e_n^T Q_0^H \sum_{i=1}^{l} \left.\frac{\partial B(c)}{\partial c_i}\right|_{c=c^{(0)}} \Pi\left(e_n - I^{(n-1)} R_{11}^{-1} R_{12}\right)\left(c_i - c_i^{(0)}\right) + \mathscr{O}\left(\|c - c^{(0)}\|_2^2\right).$$

Now we reformulate the PPIEP. Since $\{A_q(c)\}_{q=0}^{m-1}$ are the affine families (1.1) and all $\lambda_1, \lambda_2, \cdots, \lambda_{mn}$ are distinct, the functions

$$P(\lambda_i, c) = \lambda_i^m A_m + \sum_{q=0}^{m-1} \lambda_i^q A_q(c), \quad i = 1, 2, \ldots, mn$$

are twice continuously differentiable in $\mathbf{C}^{mn}$, and $\mathrm{rank}[P(\lambda_i, c^*)] = n-1$, $i = 1, 2, \ldots, mn$. It follows from Theorem 3.1 that there exists a neighborhood $\mathscr{N}(c^*) \subset \mathbf{C}^{mn}$ of c^* such that for all $c \in \mathscr{N}(c^*)$ the matrix-valued functions $P(\lambda_i, c)\Pi$ have QR-decompositions with the column pivoting — i.e.

$$P(\lambda_i, c)\Pi_i(c) = Q_i(c)R_i(c), \quad i = 1, 2, \ldots, mn,$$

where $\Pi_i(c)$ is a permutation matrix, $Q_i(c) \in \mathscr{U}_n$,

$$R_i(c) = \begin{pmatrix} R_{11}^{(i)}(c) & R_{12}^{(i)}(c) \\ 0 & r_{nn}^{(i)}(c) \end{pmatrix}, \quad R_{11}^{(i)}(c) \in \mathscr{R}_{n-1}.$$

Assume that $\mathscr{N}(c^*)$ is sufficiently small, so that for each i the permutation matrices $\Pi_i(c)$, $i=1,2,\ldots,mn$ are constant. If the column pivoting is performed such that $R_{11}^{(i)}(c)$, $i=1,2,\ldots,mn$ are nonsingular and

$$\left|e_1^T R_i(c)e_1\right| \geq \left|e_2^T R_i(c)e_2\right| \geq \cdots \geq \left|e_n^T R_i(c)e_n\right| = \left|r_{nn}^{(i)}(c)\right|, \quad i=1,2,\ldots,mn,$$

then the polynomial eigenvalue problem $[\lambda_i^m A_m + \sum_{q=0}^{m-1}\lambda_i^q A_q(c)]x=0$ has distinct eigenvalues $\lambda_1,\lambda_2,\cdots,\lambda_{mn}$ if and only if

$$r_{nn}^{(i)}(c)=0, \quad i=1,2,\ldots,mn.$$

Thus solving the PPIEP is equivalent to solving the nonlinear system (3.1) below.

Problem 3.1. Solve the nonlinear system

$$f(c)=\begin{pmatrix} r_{nn}^{(1)}(c) \\ r_{nn}^{(2)}(c) \\ \vdots \\ r_{nn}^{(mn)}(c) \end{pmatrix}=0. \tag{3.1}$$

Let $c^{(k)} \in \mathscr{N}(c^*)$ represent the current iterate to the solution c^* of the nonlinear system (3.1). It follows from Theorem 3.1 that the functions $r_{nn}^{(i)}(c)$, $i=1,2,\ldots,mn$ are continuously differentiable at $c^{(k)}$ and

$$r_{nn}^{(i)}(c)=r_{nn}^{(i)}(c^{(k)})+\sum_{j=1}^{mn}\frac{\partial r_{nn}^{(i)}(c^{(k)})}{\partial c_j}\left(c_j-c_j^{(k)}\right)+\mathscr{O}\left(\|c_j-c_j^{(k)}\|_2^2\right), \quad j=1,2,\ldots,mn,$$

where

$$\begin{aligned}\frac{\partial r_{nn}^{(i)}(c^{(k)})}{\partial c_j} &= e_n^T Q_i(c^{(k)})^H\left(\sum_{q=0}^{m-1}\lambda_i^q A_j^{(q)}\right)\Pi_i(c^{(k)}) \\ &\quad \times\left[e_n - I^{(n-1)}\left(R_{11}^{(i)}(c^{(k)})\right)^{-1}R_{12}^{(i)}(c^{(k)})\right].\end{aligned} \tag{3.2}$$

According to (3.2), the Newton method [33] can be applied to the nonlinear system (3.1). Note that each iteration has the form

$$J_f(c^{(k)})(c^{(k+1)}-c^{(k)})=-f(c^{(k)}), \tag{3.3}$$

where

$$J_f(c)=\left(\frac{\partial r_{nn}^{(i)}(c)}{\partial c_j}\right)$$

is the Jacobian matrix of the nonlinear system (3.1).

Summarizing, we obtain Algorithm 3.1 for solving the PPIEP. As is shown in [12, Lemma 4.1], the iterates $c^{(k)}$ generated by Algorithm 3.1 remain unchanged with different QR-decompositions of $P(\lambda_i,c^{(k)})\Pi_i(c^{(k)})$, $i=1,2,\ldots,mn$. The convergence of Algorithm 3.1 can be described by Theorem 3.2.

Algorithm 3.1 Newton Method Based on Smooth QR-Decomposition.

Input: Matrices A_m, $\{A_j^{(q)}\}_{j=0}^{mn}$, $q=0,1,\ldots,m-1$, eigenvalues $\{\lambda_i\}_{i=1}^{mn}$, and an initial guess $c^{(0)}$.
Output: Computed solution $c^{(k+1)}$.
for $k=0,1,2,\ldots$ until convergence **do**
  Compute
$$P(\lambda_i,c^{(k)})=\lambda_i^m A_m+\sum_{q=0}^{m-1}\lambda_i^q A_q(c^{(k)}),\quad i=1,2,\ldots,mn.$$
  Compute QR-decomposition of $P(\lambda_i,c^{(k)})$, $i=1,2,\ldots,mn$ with column pivoting
$$P(\lambda_i,c^{(k)})\Pi_i(c^{(k)})=Q_i(c^{(k)})R_i(c^{(k)}),\quad i=1,2,\ldots,mn,$$
  where $\Pi_i(c^{(k)})$ is a permutation matrix, $Q_i(c^{(k)})\in\mathscr{U}_n$,
$$R_i(c^{(k)})=\begin{pmatrix} R_{11}^{(i)}(c^{(k)}) & R_{12}^{(i)}(c^{(k)})\\ 0 & r_{nn}^{(i)}(c^{(k)})\end{pmatrix},\quad R_{11}^{(i)}(c^{(k)})\in\mathscr{R}_{n-1}.$$
  Form the vector $f(c^{(k)})$ by Eq. (3.1).
  if $\|f(c^{(k)})\|$ is small enough **then**
    Stop;
  else
    Form the Jacobian matrix $J_f(c^{(k)})$ by Eq. (3.2).
  end if
  Compute $c^{(k+1)}$ by solving Eq. (3.3).
end for

Theorem 3.2. *Let c^* be a solution of the PPIEP and $\Pi_i(c^{(k)})=\Pi_i(c^*)$, $i=1,2,\ldots,mn$ used in Algorithm* 3.1 *be independent of k for sufficiently small $\|c^*-c^{(k)}\|_2$. If the Jacobian matrix $J_f(c^*)$ corresponding to the QR-decompositions of $(\lambda_i^m A_m+\sum_{q=0}^{m-1}\lambda_i^q A_q(c^{(k)}))\Pi_i(c^*)$, $i=1,2,\ldots,mn$ is nonsingular, then the sequence $\{c^{(k)}\}$ generated by Algorithm* 3.1 *converges locally quadratically to c^*.*

Proof. It is similar to that of [12, Theorem 4.1] and is omitted here. □

4. Newton and Newton-like Methods Based on Smallest Singular Value

In light of the properties of the smallest singular value, we develop two additional approaches for solving the PPIEP. Let $\sigma_n^{(i)}(c)=\sigma_{\min}(P(\lambda_i,c))$ be the smallest singular value of the polynomial pencil
$$P(\lambda_i,c)=\lambda_i^m A_m+\sum_{q=0}^{m-1}\lambda_i^q A_q(c).$$

Since $\{A_q(c)\}_{q=0}^{m-1}$ are affine families (1.1), $P(\lambda_i, c)$ is an analytic matrix-valued function of $c \in \mathbf{R}^{mn}$. If $\{\lambda_i\}_{i=1}^{mn}$ are given distinct eigenvalues and c^* is a solution of the PPIEP, the smallest singular value $\sigma_n^{(i)}(c^*)$ of $P(\lambda_i, c^*)$ is simple. Thus, the polynomial pencil $P(\lambda_i, c)$ has the eigenvalues $\{\lambda_i\}_{i=1}^{mn}$ if and only if

$$\sigma_n^{(i)}(c) = 0, \quad i = 1, 2, \ldots, mn.$$

Accordingly, in some neighborhood of c^*, we can reformulate the PPIEP as follows.

Problem 4.1. Solve the nonlinear system

$$g(c) = \begin{pmatrix} \sigma_n^{(1)}(c) \\ \sigma_n^{(2)}(c) \\ \vdots \\ \sigma_n^{(mn)}(c) \end{pmatrix} = 0. \tag{4.1}$$

In order to apply the Newton method to the nonlinear system (4.1), we use Sun's Theorem [37] to calculate the partial derivatives of $g(c)$ with respect to $c_1, c_2, \cdots, c_{mn}$.

Theorem 4.1 (cf. Sun [37]). *Let $p = (p_1, p_2, \cdots, p_l)^T \in \mathbf{R}^l$ and $E(p) \in \mathbf{C}^{m\times n}$. Suppose that* $\mathrm{Re}[E(p)]$ *and* $\mathrm{Im}[E(p)]$ *are real analytic matrix-valued functions of p in some neighborhood $\mathscr{N}(p^{(0)}) \subseteq \mathbf{R}^l$ of $p^{(0)}$. If σ is a simple non-zero singular value of $E(p^{(0)})$, $u \in \mathbf{C}^m$ and $v \in \mathbf{C}^n$ are associated unit left and right singular vectors, respectively, then:*

(1) There is a simple singular value $\sigma(p)$ of $E(p)$ which is a real analytic function of p in some neighborhood $\mathscr{N}_1(p^{(0)})$ of $p^{(0)}$, and

$$\sigma(p^{(0)}) = \sigma, \quad \frac{\partial \sigma(p^{(0)})}{\partial p_j} = \mathrm{Re}\left[u^H \frac{\partial E(p^{(0)})}{\partial p_j} v\right],$$

where

$$\frac{\partial \sigma(p^{(0)})}{\partial p_j} = \left.\frac{\partial \sigma(p)}{\partial p_j}\right|_{p=p^{(0)}}, \quad \frac{\partial E(p^{(0)})}{\partial p_j} = \left.\frac{\partial E(p)}{\partial p_j}\right|_{p=p^{(0)}}.$$

(2) The unit left singular vector $u(p)$ and the unit right singular vector $v(p)$ of $E(p)$ corresponding to $\sigma(p)$ may be defined so that $\mathrm{Re}[u(p)], \mathrm{Im}[u(p)], \mathrm{Re}[v(p)]$, *and* $\mathrm{Im}[v(p)]$ *are real analytic functions of p in $\mathscr{N}_1(p^{(0)})$, and $u(p^{(0)}) = u, v(p^{(0)}) = v$.*

Assume that the current iterate $c^{(k)} \in \mathbf{R}^{mn}$ is sufficiently close to the solution $c^* \in \mathbf{R}^{mn}$ of the nonlinear system (4.1). Theorem 4.1 states that if the smallest singular value $\sigma_n^{(i)}(c^{(k)}) \neq 0$ of $P(\lambda_i, c^{(k)})$, then there exists a neighborhood $\mathscr{N}(c^{(k)}) \subseteq \mathbf{R}^{mn}$ of c^* such that $\sigma_n^{(i)}(c)$ is analytic and

$$\begin{aligned}\frac{\partial \sigma_n^{(i)}(c^{(k)})}{\partial c_j} &= \mathrm{Re}\left[u_n^{(i)}(c^{(k)})^H \frac{\partial P(\lambda_i, c^{(k)})}{\partial c_j} v_n^{(i)}(c^{(k)})\right] \\ &= \mathrm{Re}\left[u_n^{(i)}(c^{(k)})^H \left(\sum_{q=0}^{m-1} \lambda_i^q A_j^{(q)}\right) v_n^{(i)}(c^{(k)})\right], \quad i, j = 1, 2, \ldots, mn,\end{aligned} \tag{4.2}$$

where $u_n^{(i)}(c^{(k)})$ and $v_n^{(i)}(c^{(k)})$ are the unit left and right singular vectors associated with $\sigma_n^{(i)}(c^{(k)})$, respectively. Thus the Jacobian matrix of $g(c)$ is

$$J_g(c)=\left(\frac{\partial\sigma_n^{(i)}(c)}{\partial c_j}\right),$$

and the step of the Newton method is defined as

$$J_g(c^{(k)})(c^{(k+1)}-c^{(k)})=-g(c^{(k)}). \tag{4.3}$$

Consequently, the Newton method for solving the PPIEP has the form.

Algorithm 4.1 Newton Method Based on the Smallest Singular Value.

1: **Input**: Matrices A_m, $\{A_j^{(q)}\}_{q=0}^{mn}$, $q=0,1,\dots,m-1$, given eigenvalues $\{\lambda_i\}_{i=1}^{mn}$, and an initial guess $c^{(0)}$.
2: **Output**: Computed solution $c^{(k+1)}$.
3: **for** $k=0,1,2,\dots$ until convergence **do**
4: Compute
$$P(\lambda_i,c^{(k)})=\lambda_i^m A_m+\sum_{q=0}^{m-1}\lambda_i^q A_q(c^{(k)}),\quad i=1,2,\dots,mn.$$
5: Compute the smallest singular values $\sigma_n^{(i)}(c^{(k)})$ and the corresponding unit left and right singular vectors $u_n^{(i)}(c^{(k)})$ and $v_n^{(i)}(c^{(k)})$ of $P(\lambda_i,c^{(k)})$, $i=1,2,\dots,mn$.
6: Form the vector $g(c^{(k)})$ by Eq. (4.1).
7: **if** $\|g(c^{(k)})\|$ is small enough **then**
8: Stop;
9: **else**
10: Form the Jacobian matrix $J_g(c^{(k)})$ by Eq. (4.2).
11: **end if**
12: Compute $c^{(k+1)}$ by solving linear system Eq. (4.3).
13: **end for**

It follows from Theorem 4.1 that the smallest singular values $\{\sigma_n^{(i)}(c)\}_{i=1}^{mn}$ of $\{P(\lambda_i,c)\}_{i=1}^{mn}$ near $c=c^*$ are smooth dependent. Combined with the assumption that the Jacobian matrix of $g(c)$ at $c=c^*$ is non-singular, the results of [33] can be employed to investigate the convergence of Algorithm 4.1.

Theorem 4.2. *Let the given eigenvalues $\{\lambda_i\}_{i=1}^{mn}$ be distinct and c^* be a solution of the PPIEP. If the Jacobian matrix $J_g(c^*)$ corresponding to $g(c)$ is nonsingular, then there exists a neighborhood $\mathcal{N}(c^*)$ of c^* such that for all $c^{(0)}\in\mathcal{N}(c^*)$, Algorithm* 4.1 *is locally quadratic convergent.*

Proof. We can immediately obtain the result by applying the well-known theorem concerning the convergence of Newton method in [33]. $\square$

In each iteration of Algorithm 4.1, Step 5 involves the computation of mn smallest singular values and the corresponding unit left and right singular vectors by singular value decomposition. This process is time-consuming, as it requires $21mn^4$ flops [25]. In order to reduce the computational cost, we propose the utilization of one step of inverse iteration to approximate the smallest singular values and the associated unit left and right singular vectors of $P(\lambda_i, c^{(k)})$.

Let $\sigma_i^{(k)}$ and $u_i^{(k)}$, $v_i^{(k)}$ be approximations of the smallest singular value $\sigma_n^{(i)}(c^{(k)})$ and the associated unit left, right singular vectors $u_n^{(i)}(c^{(k)})$, $v_n^{(i)}(c^{(k)})$ of $P(\lambda_i, c^{(k)})$, respectively. We update these approximations by one step of inverse iteration — i.e.

$$P(\lambda_i, c^{(k)})w = u_i^{(k-1)}, \tag{4.4}$$

$$v_i^{(k)} = w/\|w\|, \tag{4.5}$$

$$P(\lambda_i, c^{(k)})^H y = v_i^{(k)}, \tag{4.6}$$

$$\sigma_i^{(k)} = 1/\|y\|, \tag{4.7}$$

$$u_i^{(k)} = \sigma_i^{(k)} y, \tag{4.8}$$

where $u_i^{(0)}$ is an initial guess.

The approximate computational cost of (4.4)-(4.8) is $(2/3)n^3$ flops. This means that the application of one step of inverse iteration can significantly reduce the computation cost ($21mn^4$ flops) of Step 5 in Algorithm 4.1. In particular, each iteration can reduce it by $(61/3)mn^4$ flops. Similar to [39], we choose the unit left singular vector $u_n^{(i)}(c^{(0)})$ associated with the smallest singular value $\sigma_n^{(i)}(c^{(0)})$ of $P(\lambda_i, c^{(0)})$ as our initial guess $u_i^{(0)}$. Let

$$g_k = \left[\sigma_1^{(k)}, \sigma_2^{(k)}, \cdots, \sigma_{mn}^{(k)}\right]^T, \tag{4.9}$$

and

$$[J_k]_{ij} = \mathrm{Re}\left[\left(u_i^{(k)}\right)^H \left(\sum_{q=0}^{m-1} \lambda_i^q A_j^{(q)}\right) v_i^{(k)}\right]. \tag{4.10}$$

In order to obtain a new estimate $c^{(k+1)}$, we solve the equation

$$J_k(c^{(k+1)} - c^{(k)}) = -g_k. \tag{4.11}$$

Accordingly, the Newton-like method for solving the PPIEP has the form.

Algorithm 4.2 Newton-Like Method Based on the Smallest Singular Value.

1: **Input**: Matrices A_m, $\{A_j^{(q)}\}_{q=0}^{mn} \in \mathbf{R}^{n\times n}$, $q = 0, 1, \ldots, m-1$, given eigenvalues $\{\lambda_i\}_{i=1}^{mn} \in \mathbf{C}$, and an initial guess $c^{(0)} \in \mathbf{C}^{mn}$.
2: **Output**: Computed solution $c^{(k+1)}$.
3: **if** $k = 0$ **then**

Implement Algorithm 4.1 once to obtain $c^{(1)}$, and set

$$\sigma_i^{(0)} = \sigma_n^{(i)}(c^{(0)}), \quad u_i^{(0)} = u_n^{(i)}(c^{(0)}), \quad v_i^{(0)} = v_n^{(i)}(c^{(0)}), \quad i = 1, 2, \ldots, mn.$$

end if
for $k = 1, 2, \ldots$ until convergence **do**
Compute approximate smallest singular values $\sigma_i^{(k)}$ and the corresponding unit left and right singular vectors $u_i^{(k)}$ and $v_i^{(k)}$ of $P(\lambda_i, c^{(k)})$, $i = 1, 2, \ldots, mn$ by the inverse iteration (4.4)-(4.8).
Form the vector g_k by Eq. (4.9).
if $\|g_k\|$ is small enough **then**
Stop;
else
Form the Jacobian matrix J_k by Eq. (4.10).
end if
Compute $c^{(k+1)}$ by solving linear system (4.11).
end for

The convergence of Algorithm 4.2 is elucidated by the subsequent theorem.

Theorem 4.3. *Let the given eigenvalues $\{\lambda_i\}_{i=1}^{mn}$ be distinct and $c^* \in \mathbf{R}^{mn}$ be a solution of the PPIEP. If the Jacobian matrix $J(c^*)$ is nonsingular, then Algorithm* 4.2 *generates a well-defined sequence $\{c^{(k)}\}$ for which $c^{(k)} \to c^*$, and the convergence is locally quadratic.*

Proof. The proof is similar to [39, Theorem 3.2] and is omitted here. □

5. Numerical Experiments

We present several numerical experiments to illustrate the effectiveness of the proposed methods for solving the PPIEP. An initial guess $c^{(0)} \in \mathbf{C}^{mn}$ can be randomly generated or directly selected based on practical applications or desired solutions. To demonstrate the locally quadratic convergence after a limited number of iterations, we select initial approximations that are in close proximity to the exact solution c^* of the PPIEP.

The stop criteria of Algorithms 3.1, 4.1, 4.2 are

$$\|f(c^{(k)})\| \le 10^{-6}, \quad \|g(c^{(k)})\| \le 10^{-6}, \quad \|g^{(k)}\| \le 10^{-6},$$

where $\|f(c^{(k)})\|$, $\|g(c^{(k)})\|$ and $\|g^{(k)}\|$ are defined as in (3.1), (4.1) and (4.9). In the tables, CPU denotes the CPU time (in seconds) for computing an approximate solution for the PPIEP.

We perform the tests in MATLAB R2016b with double precision arithmetic on an Intel Core 2.9 GHz PC with 8GB memory under Windows 10 system.

Example 5.1. The numerical solution of vibration problems by the dynamic element method leads to cubic eigenvalue problems [32]

$$\lambda^3 F_3 + \lambda^2 F_2 + \lambda F_1 + F_0 = 0,$$

where $F_i^T = F_i$, $i = 0, 1, 2, 3$. Thus we construct a symmetric parameterized cubic inverse eigenvalue problem

$$\lambda^3 A_3 + \lambda^2 A_2(c) + \lambda A_1(c) + A_0(c) = 0,$$

where $A_q(c) = \sum_{j=1}^{9} c_j A_j^{(q)}$, $q = 0, 1, 2$ and A_3, $\{A_j^{(q)}\}_{j=1}^{9} \in \mathbf{R}^{n\times n}$ are symmetric matrices.

Let $m = 3$, $n = 3$,

$$A_3 = \begin{pmatrix} 5 & 2 & 1 \\ 2 & 7 & 5 \\ 1 & 5 & 6 \end{pmatrix}, \quad A_0^{(0)} = \begin{pmatrix} 4 & 2 & 5 \\ 2 & 8 & 3 \\ 5 & 3 & 2 \end{pmatrix},$$

$$A_0^{(1)} = \begin{pmatrix} 4 & 3 & 3 \\ 3 & 2 & 4 \\ 3 & 4 & 2 \end{pmatrix}, \quad A_0^{(2)} = \begin{pmatrix} 4 & 3 & 7 \\ 3 & 8 & 4 \\ 7 & 4 & 4 \end{pmatrix},$$

$$A_1^{(0)} = A_4^{(1)} = A_7^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$A_2^{(0)} = A_5^{(1)} = A_8^{(2)} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix},$$

$$A_3^{(0)} = A_6^{(1)} = A_9^{(2)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix},$$

and the remaining matrices be zero matrices. The given eigenvalues are $-3.4, -1.4, 1.3, -0.2 \pm 0.6i, \pm 1.0i, 0.3 \pm 1.0i$, where $i^2 = -1$. With the starting value

$$c^{(0)} = (1, 1, 1, 1, 1, 1, 1, 1, 1)^T,$$

the sequences $\{c^{(k)}\}$ generated by Algorithms 3.1, 4.1 and 4.2 converge to the exact solution

$$c^* = (1.7631, 0.2767, 1.6279, 0.3296, 0.1041, 0.2119, 0.2843, 0.3467, 0.5191)^T.$$

The numerical results presented in Table 1 show that for symmetric PPIEP, all three methods are effective and locally quadratic convergent.

Example 5.2. Consider now a parameterized cubic inverse eigenvalue problem with non-symmetric matrices. Let $m = 3$, $n = 3$,

$$A_3 = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 0 \\ 3 & 2 & 3 \end{pmatrix}, \quad A_1^{(0)} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$A_2^{(0)} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_3^{(0)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix},$$

$$A_4^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \quad A_5^{(1)} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \\ 0 & 1 & 0 \end{pmatrix}, \quad A_6^{(1)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \end{pmatrix},$$

$$A_7^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad A_8^{(2)} = \begin{pmatrix} 0 & 0 & 0 \\ 2 & 0 & 1 \\ 0 & 2 & 0 \end{pmatrix}, \quad A_9^{(2)} = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

and remaining matrices be zero matrices. The target eigenvalues are $0.7643, -0.5370, -1.2278, 1.3867 \pm 0.3248i, -0.0046 \pm 0.9147i, -0.5818 \pm 0.3594i$. We choose the initial guess as

$$c^{(0)} = (1.2, 1.2, 1.2, 1.1, 1.3, 1.4, 1.4, 1.3, 1.1)^T.$$

Applying Algorithms 3.1, 4.1 and 4.2, we obtain exact solution

$$c^* = (1, 1, 1, 1, 1, 1, 1, 1, 1)^T.$$

Numerical results are displayed in Table 2. It is clear that the proposed methods exhibit effectiveness in handling the asymmetric PPIEP, while also preserving the locally quadratic convergence.

Table 1: Convergence of algorithms for symmetric PPIEP.

Iteration	Algorithm 3.1		Algorithm 4.1		Algorithm 4.2	
k	$\|c^{(k)} - c^*\|$	$\|f(c^{(k)})\|$	$\|c^{(k)} - c^*\|$	$\|g(c^{(k)})\|$	$\|c^{(k)} - c^*\|$	$\|g_k\|$
0	2.13e-00	5.38e-00	2.13e-00	4.48e-00	2.13e-00	4.48e-00
1	7.06e-01	1.14e-00	8.38e-01	1.73e-00	8.38e-01	1.76e-00
2	5.09e-02	1.25e-01	9.17e-02	1.77e-01	4.14e-02	9.63e-02
3	4.25e-04	7.47e-04	4.70e-04	9.81e-04	9.86e-05	2.03e-04
4	2.33e-08	3.37e-08	3.93e-08	4.31e-08	1.83e-08	2.15e-09
CPU	0.0471		0.0363		0.0272	

Table 2: Convergence of algorithms for non-symmetric PPIEP.

Iteration	Algorithm 3.1		Algorithm 4.1		Algorithm 4.2	
k	$\|c^{(k)} - c^*\|$	$\|f(c^{(k)})\|$	$\|c^{(k)} - c^*\|$	$\|g(c^{(k)})\|$	$\|c^{(k)} - c^*\|$	$\|g_k\|$
0	8.00e-01	1.76e-00	8.01e-00	1.39e-00	8.01e-00	1.39e-00
1	2.44e-01	2.33e-01	1.46e-01	1.19e-01	1.46e-01	1.19e-01
2	4.16e-02	4.02e-02	5.57e-03	6.89e-03	5.69e-03	7.08e-03
3	1.14e-03	1.06e-03	1.73e-05	1.41e-05	2.50e-05	1.63e-05
4	8.16e-07	7.64e-07	2.11e-10	8.61e-11	3.91e-10	1.80e-10
CPU(s)	0.0481		0.0319		0.0278	

Example 5.3. Consider various high-order PPIEPs. Matrices A_m, $A_q = (a_{st}^{(q)}) \in \mathbf{R}^{n\times n}$, $q = 0,1,\ldots,m-1$ are randomly generated by the MATLAB function `rand`, whose elements uniformly distributed over $[-1.0, 1.0]$. Let

$$\begin{aligned}
A_{qn+j}^{(q)} &= a_{jj}^{(q)} e_j e_j^T + \sum_{i=j+1}^{n} \left(a_{ji}^{(q)} e_j e_i^T + a_{ij}^{(q)} e_i e_j^T \right), \\
A_{qn+n}^{(q)} &= a_{nn}^{(q)} e_n e_n^T, \quad q = 0,1,\ldots,m-1, \quad j = 1,2,\ldots,n-1,
\end{aligned}$$

and remaining matrices be zero matrices. We first compute the eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_{mn}$ of the polynomial eigenvalue problem

$$\left(\lambda^m A_m + \sum_{q=0}^{m-1} \lambda^q A_q(c) \right) x = 0$$

at $c^* = (1,1,\cdots,1) \in \mathbf{R}^{mn}$, and then recompute the exact solution c^* of the PPIEP by using calculated eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_{mn}$. The initial value $c^{(0)}$ is obtained by perturbing c^*

$$c^{(0)} = c^* + \mu r_c,$$

where $\mu = 0.01$ is the disturbance parameter, $r_c \in \mathbf{R}^{mn}$ is a vector whose elements are generated randomly and distributed uniformly within $[-1.0, 1.0]$.

We implement Algorithms 3.1, 4.1 and 4.2 and provide the numerical results in Table 3. In the table, 'Its' represents the number of iterations, while $\|c^{(Its)} - c^*\|$ denotes the final residual norm. Table 3 shows that Algorithm 3.1 requires the least iteration steps and CPU time.

For the sake of brevity, we take $n = 100$, $m = 4,5,6,7$ as examples and use Fig. 1 to show the locally quadratic convergence of the proposed methods for high-order PPIEPs. From Fig. 1, it easy to see that Algorithms 3.1, 4.1 and 4.2 still have locally quadratic convergence.

Table 3: Numerical results for high-order PPIEPs.

Order		Algorithm 3.1			Algorithm 4.1			Algorithm 4.2		
m	n	Its	CPU	$\|\|c^{(Its)} - c^*\|\|$	Its	CPU	$\|\|c^{(Its)} - c^*\|\|$	Its	CPU	$\|\|c^{(Its)} - c^*\|\|$
4	50	4	22.04	1.63e-10	3	11.32	5.91e-07	3	10.49	5.91e-07
	100	5	369.31	2.53e-10	4	208.01	3.87e-08	4	198.79	3.87e-08
5	50	5	48.50	1.20e-08	4	28.64	2.31e-06	4	26.98	2.31e-06
	100	6	839.72	4.59e-12	4	421.63	1.10e-07	4	408.62	1.10e-07
6	50	5	76.15	3.51e-07	5	59.01	4.04e-07	5	55.43	4.01e-07
	100	6	1368.86	1.12e-07	5	901.42	2.22e-06	5	869.01	2.16e-06
7	50	6	151.91	6.82e-12	5	95.84	1.38e-08	5	87.32	1.39e-08
	100	8	3380.79	1.43e-11	5	1638.90	1.45e-11	5	1418.65	1.35e-11

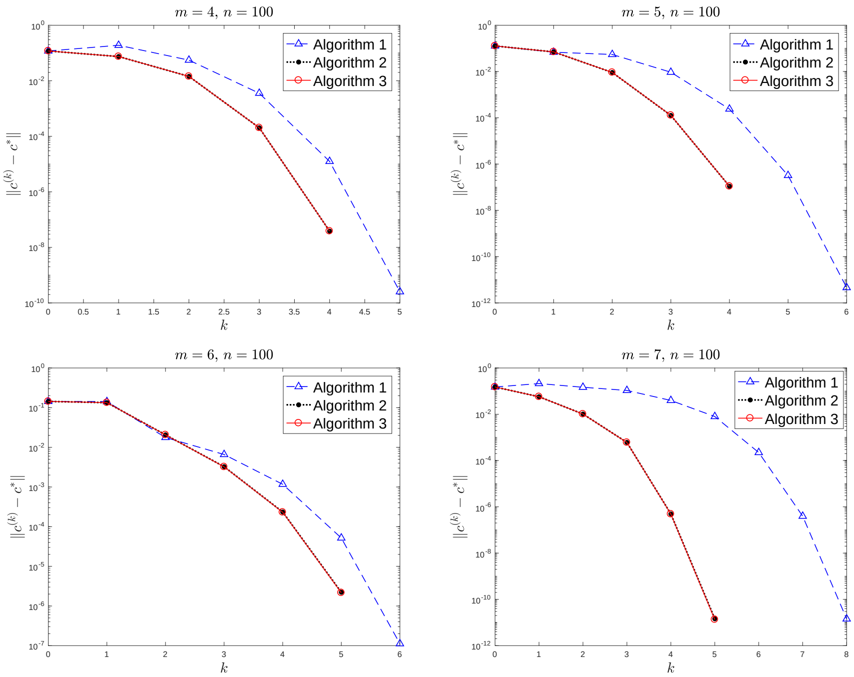

Figure 1: Convergence of the proposed methods for high-order PPIEPs.

6. Conclusion

This study focuses on two key aspects of PPIEPs — viz. the solvability and computability. Using the theory of multiparameter eigenvalue problem, we propose a sufficient condition for the existence of a solution to the PPIEP. In order to solve PPIEP, we propose the Newton method based on the smooth QR-decomposition with column pivoting, and the Newton and Newton-like methods based on the smallest singular values. If the given eigenvalues are distinct, the PPIEP has a solution c^*, the Jacobian matrix $J(c^*)$ is nonsingular, and all methods are locally quadratic convergent. Numerical results demonstrate that our proposed numerical methods work well for solving the PPIEP. In addition to the algorithms proposed in this paper, the Newton method based on the smooth LU-decomposition with complete pivoting proposed and analyzed in [15] can also be extended to solve both symmetric and asymmetric PPIEP. However, it is computationally expensive to compute mn LU-decompositions with complete pivoting in each iteration, see [14] for more details. This leads to lower efficiency of the algorithm based on smooth LU-decomposition compared to the algorithms proposed in this paper. Through a substantial number of experiments, we found that the computational cost of these methods are dominated by the process of constructing the Jacobian matrix during each iteration. The efficiency can be improved through the utilization of the quasi-Newton method. Furthermore, our methods may be extended to solve the parameterized polynomial inverse eigenvalue problems of analytical matrix-valued functions $\{A_q(c)\}_{q=0}^m$ that depend on the parameter $c \in \mathbf{R}^{mn}$ or $c \in \mathbf{C}^{mn}$.

Acknowledgments

The authors are very much indebted to anonymous referees for their useful comments and suggestions and to the Co-Editor of EAJAM, Prof. Victor Didenko for his careful editing, which improved the presentation of the paper.

This work is supported by the Nanjing Institute of Technology Research Foundation under Grant No. YKJ202355.

References

[1] K. Aishima, *A quadratically convergent algorithm for inverse eigenvalue problems with multiple eigenvalues*, Linear Algebra Appl. **549**, 30–52 (2018).
[2] A.L. Andrew, K.W.E. Chu and P. Lancaster, *Derivatives of eigenvalues and eigenvectors of matrix functions*, SIAM J. Matrix Anal. Appl. **14**, 903–926 (1993).
[3] F.V. Atkinson, *Multiparameter Eigenvalue Problems. Volume I: Matrices and Compact Operators*, Academic Press (1972).
[4] Z.-J. Bai, *Inexact Newton methods for inverse eigenvalue problems*, Appl. Math. Comput. **172**, 682–689 (2006).
[5] Z.-J. Bai, R.H. Chan and B. Morini, *An inexact Cayley transform method for inverse eigenvalue problems*, Inverse Probl. **20**, 1675–1689 (2004).
[6] F.W. Biegler-König, *A Newton iteration process for inverse eigenvalue problems*, Numer. Math. **37**, 349–354 (1981).
[7] Z. Bohte, *Numerical solution of the inverse algebraic eigenvalue problem*, Computer J. **10**, 385–388 (1968).
[8] R.H. Chan, H.L. Chung and S.-F. Xu, *The inexact Newton-like method for inverse eigenvalue problem*, BIT **43**, 7–20 (2003).
[9] M.T. Chu and G.H. Golub, *Inverse Eigenvalue Problems: Theory, Algorithms, and Applications*, Oxford University Press (2005).
[10] N. Cottin, *Dynamic model updating – a multiparameter eigenvalue problem*, Mech. Syst. Signal Pr. **15**, 649–665 (2001).
[11] N. Cottin and J. Reetz, *Accuracy of multiparameter eigenvalues used for dynamic model updating with measured natural frequencies only*, Mech. Syst. Signal Pr. **20**, 65–77 (2006).
[12] H. Dai, *An algorithm for symmetric generalized inverse eigenvalue problems*, Linear Algebra Appl. **296**, 79–98 (1999).
[13] H. Dai, *The Theory of Matrices (in Chinese)*, Science Press (2001).
[14] H. Dai and Z.-Z. Bai, *On smooth LU decompositions with applications to solutions of nonlinear eigenvalue problems*, J. Comput. Math. **28**, 745–766 (2010).
[15] H. Dai, Z.-Z. Bai and Y. Wei, *On the solvability condition and numerical algorithm for the parameterized generalized inverse eigenvalue problem*, SIAM J. Matrix Anal. Appl. **36**, 707–726 (2015).
[16] H. Dai and P. Lancaster, *Newton's method for a generalized inverse eigenvalue problem*, Numer. Linear Algebra Appl. **4**, 1–21 (1997).
[17] H. Dai and P. Lancaster, *Numerical methods for finding multiple eigenvalues of matrices depending on parameters*, Numer. Math. **76**, 189–208 (1997).
[18] Z. Dalvand and M. Hajarian, *Solving generalized inverse eigenvalue problems via L-BFGS-B method*, Inverse Probl. in Sci. En. **28**, 1719–1746 (2020).

[19] Z. Dalvand and M. Hajarian, *Newton-like and inexact Newton-like methods for a parameterized generalized inverse eigenvalue problem*, Math. Meth. Appl. Sci. **44**, 4217–4234 (2021).

[20] Z. Dalvand, M. Hajarian and J.E. Roman, *An extension of the Cayley transform method for a parameterized generalized inverse eigenvalue problem*, Numer. Linear Algebra Appl. **27**, e2327 (2020).

[21] B.N. Datta and V. Sokolov, *A solution of the affine quadratic inverse eigenvalue problem*, Linear Algebra Appl. **434**, 1745–1760 (2011).

[22] S. Elhay and Y.M. Ram, *An affine inverse eigenvalue problem*, Inverse Probl. **18**, 455–466 (2002).

[23] S. Elhay and Y.M. Ram, *Quadratic pencil pole assignment by affine summs*, ANZIAM J. **45**, 592–603 (2004).

[24] S. Friedland, J. Nocedal and M.L. Overton, *The formulation and analysis of numerical methods for inverse eigenvalue problems*, SIAM J. Numer. Anal. **24**, 634–667 (1987).

[25] G.H. Golub and V.C.F. Loan, *Matrix Computation*, The Johns Hopkins University Press (1996).

[26] X.-Z. Ji, *On matrix inverse eigenvalue problems*, Inverse Probl. **14**, 275–285 (1998).

[27] K.T. Joseph, *Inverse eigenvalue problem in structural design*, AIAA J. **30**, 2890–2896 (1992).

[28] H.W. Kuhn, *The Hungarian method for the assignment problem*, Naval Res. Logist. Quart. **2**, 83–97 (1955).

[29] P. Lancaster, *Algorithms for lambda-matrices*, Numer. Math. **6**, 388–394 (1964).

[30] R.-C. Li, *QR-decomposition and nonlinear eigenvalue problems (in Chinese)*, Math. Numer. Sinica **11**, 374–385 (1989).

[31] W. Ma, *A backward error for the symmetric generalized inverse eigenvalue problem*, Linear Algebra Appl. **464**, 90–99 (2015).

[32] V. Mehrmann and D. Watkins, *Polynomial eigenvalue problems with Hamiltonian structure*, Electron. Trans. Numer. Anal. **13**, 106–118 (2002).

[33] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press (1970).

[34] R.L. Parker and K.A. Whaler, *Numerical methods for establishing solutions to the inverse problem of electromagnetic induction*, J. Geophys. Res. Solid Earth **86**, 9574–9584 (1981).

[35] W.-P. Shen and C. Li, *An Ulm-like Cayley transform method for inverse eigenvalue problems*, Taiwanese J. Math. **16**, 367–386 (2012).

[36] Q. Song and H. Dai, *Numerical methods for solving generalized inverse eigenvalue problem*, Journal of Nanjing University of Aeronautics and Astronautics **31**, 679–685 (1999).

[37] J.-G. Sun, *A note on simple non-zero singular values*, J. Comput. Math. **6**, 258–266 (1988).

[38] M.-L. Xiang and H. Dai, *On parameterized quadratic inverse eigenvalue problem*, East Asian J. Appl. Math. **12**, 185–200 (2022).

[39] M.-L. Xiang and H. Dai, *Smallest singular value based Newton-like methods for solving quadratic inverse eigenvalue problem*, East Asian J. Appl. Math. **12**, 333–352 (2022).

[40] S.-F. Xu, *A smallest singular value method for solving inverse eigenvalue problems*, J. Comput. Math. **14**, 23–31 (1996).

[41] S.-F. Xu, *An Introduction to Inverse Algebraic Eigenvalue Problems*, Peking University Press (1998).

[42] S.-Q. Zhou and H. Dai, *The Algebraic Inverse Eigenvalue Problem (in Chinese)*, Henan Science and Technology Press (1991).