BIT Numerical Mathematics (2020) 60:825–852
https://doi.org/10.1007/s10543-019-00798-9

A geometric Gauss–Newton method for least squares inverse eigenvalue problems

Teng-Teng Yao[1] · Zheng-Jian Bai[2] · Xiao-Qing Jin[3] · Zhi Zhao[4]

Received: 31 January 2019 / Accepted: 18 December 2019 / Published online: 9 January 2020

Abstract
This paper is concerned with the least squares inverse eigenvalue problem of reconstructing a linear parameterized real symmetric matrix from the prescribed partial eigenvalues in the sense of least squares, which was originally proposed by Chen and Chu (SIAM J Numer Anal 33:2417–2430, 1996). We provide a geometric Gauss–Newton method for solving the least squares inverse eigenvalue problem. The global and local convergence analysis of the proposed method is established under some assumptions. Also, a preconditioned conjugate gradient method with an efficient preconditioner is proposed for solving the geometric Gauss–Newton equation. Finally, some numerical tests, including an application in the inverse Sturm–Liouville problem, are reported to illustrate the efficiency of the proposed method.

Keywords Parameterized least squares inverse eigenvalue problem · Geometric Gauss–Newton method · Preconditioner

Mathematics Subject Classification 65F18 · 65F15 · 15A18 · 58C15

1 Introduction

An inverse eigenvalue problem (IEP) aims to reconstruct a structured matrix from the prescribed spectral data. Inverse eigenvalue problems (IEPs) arise in various applications such as structural dynamics, vibration, inverse Sturm–Liouville problem, control design, geophysics, nuclear spectroscopy and molecular spectroscopy, etc. For the

Communicated by Michiel E. Hochstenbach.

This work was supported by NSFC Nos. 11701514, 11671337, 11601112. The research of Z. J. Bai was partially supported by the Fundamental Research Funds for the Central Universities (No. 20720180008). The research of X. Q. Jin was supported by the research grant MYRG2016-00077-FST from University of Macau.

Extended author information available on the last page of the article

existence theory, numerical methods and applications of general IEPs, one may refer to [11,13,14,20,24,25,43] and references therein.

Recently, one may find some new applications and numerical methods of IEPs including the experiments for beaded strings [17], the inverse eigenvalue problem for graphs [5,6,31], the solvability condition and numerical algorithm for the parameterized generalized IEP [18], the weight optimization in a geodetic network [19], the IEP in wireless communications [40], the transcendental inverse eigenvalue problem in damage parameter estimation [36,37], the IEP for quantum channels [42], and the IEP in microelectromechanical systems [38], etc. In many applications, the number of distributed parameters in the required structured matrix is often smaller than the number of measured eigenvalues (see for instance [9,22,39]). Thus it is desirable that one can find a least square solution to a structured IEP. Sparked by this, in this paper, we consider the following parameterized least squares inverse eigenvalue problem, which was originally given by Chen and Chu [10].

PLSIEP I *Given $l+1$ real symmetric matrices $A_0, A_1, \ldots, A_l \in \mathbb{R}^{n\times n}$ and m real numbers $\lambda_1^* \le \lambda_2^* \le \cdots \le \lambda_m^*$ $(m \le n)$, find a vector $\mathbf{c} = (c_1, c_2, \ldots, c_l)^T \in \mathbb{R}^l$ and a permutation $\sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$ with $1 \le \sigma_1 < \sigma_2 < \cdots < \sigma_m \le n$ to minimize the function*

$$f(\mathbf{c}, \sigma) := \frac{1}{2}\sum_{i=1}^{m}(\lambda_{\sigma_i}(\mathbf{c}) - \lambda_i^*)^2,$$

where the real numbers $\lambda_1(\mathbf{c}) \le \lambda_2(\mathbf{c}) \le \cdots \le \lambda_n(\mathbf{c})$ are the eigenvalues of the matrix $A(\mathbf{c})$ defined by

$$A(\mathbf{c}) := A_0 + \sum_{i=1}^{l} c_i A_i.$$

This is a nonlinear least-squares problem, where $l \le m \le n$ and the cost function $f(\mathbf{c}, \sigma)$ is a function of a continuous variable $\mathbf{c}$ and a discrete variable σ. This is a special kind of mixed optimization problem, where the function $f(\mathbf{c}, \sigma)$ is non-differentiable when the permutation σ is changed. For the PLSIEP I, there exists an equivalent least-squares problem defined on a product manifold. Let $\mathscr{D}(m)$ and $\mathscr{O}(n)$ denote the set of all real diagonal matrices of order m and the set of all real $n \times n$ orthogonal matrices, respectively. Define $\Lambda_m^* := \operatorname{diag}(\lambda_1^*, \lambda_2^*, \ldots, \lambda_m^*)$, where $\operatorname{diag}(\mathbf{a})$ denotes a diagonal matrix with $\mathbf{a}$ on its diagonal. Given a matrix $\Lambda \in \mathscr{D}(n-m)$, $\operatorname{blkdiag}\left(\Lambda_m^*,\ \Lambda\right)$ denotes the block diagonal matrix obtained from Λ_m^* and Λ. Based on Theorem 3.2 in [10], the PLSIEP I is equivalent to the following problem.

PLSIEP II *Given $l+1$ real symmetric matrices $A_0, A_1, \ldots, A_l \in \mathbb{R}^{n\times n}$ and m real numbers $\lambda_1^* \le \lambda_2^* \le \cdots \le \lambda_m^*$ $(m \le n)$, find a vector $\mathbf{c} \in \mathbb{R}^l$, an orthogonal matrix $Q \in \mathscr{O}(n)$, and a diagonal matrix $\Lambda \in \mathscr{D}(n-m)$ to minimize the function*

$$h(\mathbf{c}, Q, \Lambda) := \frac{1}{2}\|A(\mathbf{c}) - Q\operatorname{blkdiag}\left(\Lambda_m^*,\ \Lambda\right)Q^T\|_F^2,$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

The PLSIEP II is a nonlinear least-squares problem defined on the product manifold $\mathbb{R}^l \times \mathscr{O}(n) \times \mathscr{D}(n-m)$. To solve the PLSIEP II, Chen and Chu [10] proposed a lift and projection (LP) method. This method is a modification of the alternating projection method to an affine space and a Riemannian manifold. To solve the PLSIEP I, Chen and Chu [10] proposed a hybrid method called the LP-Newton method. The idea is that an initial guess of $(\mathbf{c}, \sigma)$ is obtained by using the LP method to the PLSIEP II, then the Newton method is applied to the PLSIEP I by fixing the value of σ. As noted in [10], in spite of non-smoothness of f in σ, one may still adopt the Newton method for solving the PLSIEP I, where the Hessian matrix of f can be explicitly derived. However, as the separation of eigenvalues decreases, the Hessian matrix becomes increasingly ill-conditioned. Also, as noted in [23], when $l < m$, it is natural to consider the well-known techniques in [21, Chap. 10] (e.g., the Gauss–Newton method). Wang and Vong [41] proposed a Gauss–Newton-like method for a special case of $m = n$.

Optimization methods on smooth manifolds have been widely studied and applied to various kinds of areas such as numerical linear algebra and dynamical systems (see for instance [1–4,12,15,30,35] and references therein). Recently, some Riemannian optimization methods were proposed for solving nonlinear eigenvalue problems and inverse eigenvalue problems [44–47]. In this paper, we propose a geometric inexact Gauss–Newton method for solving the PLSIEP II. Absil et al. [2] proposed a Riemannian Gauss–Newton method for solving nonlinear least squares problems defined between Riemannian manifold and Euclidean space, where the convergence analysis was not discussed. Gratton et al. [27] gave some approximate Gauss–Newton methods for solving nonlinear least squares problems defined on Euclidean space. Sparked by [2,27], we present an efficient geometric inexact Gauss–Newton method for solving the PLSIEP II. The global convergence and local convergence rate are also derived under some assumptions. An effective preconditioner is proposed for solving the Riemannian Gauss–Newton equation via the conjugate gradient (CG) method [26]. Finally, some numerical experiments, including an application in the inverse Sturm–Liouville problem, are reported to show the efficiency of the proposed method for solving the PLSIEP II.

Throughout this paper, we use the following notation. The symbol A^T denotes the transpose of a matrix A. The symbol $\mathrm{Diag}(M) := \mathrm{diag}(m_{11}, m_{22}, \ldots, m_{nn})$ denotes a diagonal matrix containing the diagonal elements of an $n \times n$ matrix $M = [m_{ij}]$. Let $\mathbf{0}_{m\times n}$ be the $m \times n$ zero matrix and $\mathbf{e}_k$ be the kth column of the identity matrix I_n of order n. Let $\mathbb{R}^{n\times n}$ and $\mathbb{SR}^{n\times n}$ be the set of all n-by-n real matrices and the set of all n-by-n real symmetric matrices, respectively. Denote by $\mathrm{tr}(A)$ the trace of a square matrix A. For two matrices $A, B \in \mathbb{R}^{n\times n}$, the Lie Bracket $[A, B]$ of A and B is defined as the difference of the matrix products AB and BA, i.e., $[A, B] := AB - BA$ [32, p. 512]. Let $\mathrm{vec}(A)$ be the vectorization of a matrix A, i.e., a column vector obtained by stacking the columns of A on top of one another. For two finite-dimensional vector spaces $\mathscr{X}$ and $\mathscr{Y}$ equipped with a scalar inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\|\cdot\|$, let $\mathscr{A} : \mathscr{X} \to \mathscr{Y}$ be a linear operator and the adjoint of $\mathscr{A}$ be denoted by $\mathscr{A}^*$. The operator norm of $\mathscr{A}$ is defined by $|||\mathscr{A}||| := \sup\{\|\mathscr{A}x\| \mid x \in \mathscr{X} \text{ with } \|x\| = 1\}$.

The remainder of this paper is organized as follows. In Sect. 2 we propose a geometric inexact Gauss–Newton method for solving the PLSIEP II. In Sect. 3 we establish the global convergence and local convergence rate of the proposed approach under some assumptions. A preconditioner is also proposed for solving the geometric Gauss–Newton equation. Finally, we report some numerical tests in Sect. 4 and give some concluding remarks in Sect. 5.

2 Geometric inexact Gauss–Newton method

In this section, we present a geometric inexact Gauss–Newton method for solving the PLSIEP II. Define an affine subspace and an isospectral manifold by

$$\mathscr{L} := \left\{ A_0 + \sum_{i=1}^{l} c_i A_i \mid c_i \in \mathbb{R}, i = 1, 2, \ldots, l \right\},$$
$$\mathscr{M}(\Lambda_m^*) := \left\{ X = Q \,\mathrm{blkdiag}\left(\Lambda_m^*, \Lambda\right) Q^T \in \mathbb{SR}^{n\times n} \mid Q \in \mathscr{O}(n),\ \Lambda \in \mathscr{D}(n-m) \right\}.$$

We see that $\mathscr{M}(\Lambda_m^*)$ is the set of all real $n \times n$ symmetric matrices whose spectrum contains the m real numbers $\lambda_1^*, \lambda_2^*, \ldots, \lambda_m^*$. Thus, the PLSIEP II has a solution such that $h(\mathbf{c}, Q, \Lambda) = 0$ if and only if $\mathscr{L} \cap \mathscr{M}(\Lambda_m^*) \neq \emptyset$.

Denote

$$\mathscr{Z} := \mathbb{R}^l \times \mathscr{O}(n) \times \mathscr{D}(n-m) \quad \text{and} \quad \overline{\Lambda} := \mathrm{blkdiag}(\Lambda_m^*, \Lambda). \tag{2.1}$$

Let H be a nonlinear mapping between $\mathscr{Z}$ and $\mathbb{SR}^{n\times n}$ defined by

$$H(\mathbf{c}, Q, \Lambda) := A(\mathbf{c}) - Q\overline{\Lambda}Q^T, \tag{2.2}$$

for all $(\mathbf{c}, Q, \Lambda) \in \mathscr{Z}$. Then, the PLSIEP II can be written as the following minimization problem:

$$\begin{array}{ll} \min & h(\mathbf{c}, Q, \Lambda) := \frac{1}{2}\|H(\mathbf{c}, Q, \Lambda)\|_F^2 \\ \text{subject to (s.t.)} & (\mathbf{c}, Q, \Lambda) \in \mathscr{Z}. \end{array} \tag{2.3}$$

Sparked by the ideas in [2,27], we propose a geometric inexact Gauss–Newton method for solving Problem (2.3). We note that the dimension of the product manifold $\mathscr{Z}$ is given by

$$\dim(\mathscr{Z}) = l + \frac{n(n-1)}{2} + n - m.$$

If $l < m$, then

$$\dim(\mathscr{Z}) < \dim(\mathbb{SR}^{n\times n}).$$

Therefore, the nonlinear equation $H(\mathbf{c}, Q, \Lambda) = \mathbf{0}_{n\times n}$ is an over-determined matrix equation defined on the product manifold $\mathscr{Z}$.

Notice that $\mathscr{Z}$ is an embedded submanifold of the Euclidean space $\mathbb{R}^l \times \mathbb{R}^{n\times n} \times \mathscr{D}(n-m)$. One may equip $\mathscr{Z}$ with the induced Riemannian metric:

$$g_{(\mathbf{c},Q,\Lambda)}\big((\xi_1,\eta_1,\tau_1),(\xi_2,\eta_2,\tau_2)\big) := \mathrm{tr}(\xi_1^T\xi_2) + \mathrm{tr}(\eta_1^T\eta_2) + \mathrm{tr}(\tau_1^T\tau_2),$$

for $(\mathbf{c},Q,\Lambda)\in\mathscr{Z}$, and $(\xi_1,\eta_1,\tau_1),(\xi_2,\eta_2,\tau_2)\in T_{(\mathbf{c},Q,\Lambda)}\mathscr{Z}$, and its induced norm $\|\cdot\|$. The tangent space $T_{(\mathbf{c},Q,\Lambda)}\mathscr{Z}$ of $\mathscr{Z}$ at $(\mathbf{c},Q,\Lambda)$ is given by [2, p. 42]

$$T_{(\mathbf{c},Q,\Lambda)}\mathscr{Z} = \big\{(\mathbf{r},Q\Omega,U) \mid \Omega^T = -\Omega,\ \mathbf{r}\in\mathbb{R}^l, \Omega\in\mathbb{R}^{n\times n}, U\in\mathscr{D}(n-m)\big\}.$$

Hence, $(\mathscr{Z},g)$ is a Riemannian product manifold.

For simplification, we use the following notation:

$$X_k := (\mathbf{c}_k,Q_k,\Lambda_k)\in\mathscr{Z} \quad\text{and}\quad \Delta X_k := (\Delta\mathbf{c}_k,\Delta Q_k,\Delta\Lambda_k)\in T_{X_k}\mathscr{Z}.$$

A Riemannian Gauss–Newton method for solving Problem (2.3) can be stated as follows. Given the current iterate $X_k\in\mathscr{Z}$, solve the normal equation

$$(\mathrm{D}H(X_k))^*\circ\mathrm{D}H(X_k)[\Delta X_k] = -(\mathrm{D}H(X_k))^*[H(X_k)],$$

for $\Delta X_k\in T_{X_k}\mathscr{Z}$. Here,

$$\mathrm{D}H(X_k) : T_{X_k}\mathscr{Z}\to T_{H(X_k)}\mathbb{SR}^{n\times n}$$

is the Riemannian differential of H at the point X_k, which is given by

$$\mathrm{D}H(X_k)[\Delta X_k] = (A(\Delta\mathbf{c}_k)-A_0) + [Q_k\overline{\Lambda}_kQ_k^T, \Delta Q_kQ_k^T] - (Q_kP)\Delta\Lambda_k(Q_kP)^T, \tag{2.4}$$

where

$$\overline{\Lambda}_k := \mathrm{blkdiag}(\Lambda_m^*,\Lambda_k) \quad\text{and}\quad P := \begin{bmatrix}\mathbf{0}_{m\times(n-m)}\\ I_{(n-m)\times(n-m)}\end{bmatrix}. \tag{2.5}$$

With respect to the Riemannian metric g, the adjoint

$$(\mathrm{D}H(X_k))^* : T_{H(X_k)}\mathbb{SR}^{n\times n}\to T_{X_k}\mathscr{Z}$$

of $\mathrm{D}H(X_k)$ is given by

$$(\mathrm{D}H(X_k))^*[\Delta Z_k] = \big(\mathbf{v}(\Delta Z_k), [Q_k\overline{\Lambda}_kQ_k^T,\Delta Z_k]Q_k, -\mathrm{Diag}\big((Q_kP)^T\Delta Z_k(Q_kP)\big)\big), \tag{2.6}$$

where

$$\mathbf{v}(\Delta Z_k) := \big(\mathrm{tr}(A_1^T\Delta Z_k), \mathrm{tr}(A_2^T\Delta Z_k), \ldots, \mathrm{tr}(A_l^T\Delta Z_k)\big)^T. \tag{2.7}$$

For explicit derivation of (2.4) and (2.6), one may refer to Appendix A. Based on (2.6), the Riemannian gradient of h at a point $X := (\mathbf{c}, Q, \Lambda) \in \mathscr{Z}$ is determined by [2, p. 185]:

$$\begin{aligned}\operatorname{grad} h(X) = (\mathrm{D}H(X))^*[H(X)] = \big(\mathbf{v}\big(A(\mathbf{c}) - Q\overline{\Lambda}Q^T\big), [Q\overline{\Lambda}Q^T, A(\mathbf{c}) - Q\overline{\Lambda}Q^T]Q,\\ -\operatorname{Diag}\big((QP)^T(A(\mathbf{c}) - Q\overline{\Lambda}Q^T)(QP)\big)\big).\end{aligned} \tag{2.8}$$

Let ∇ denote the Riemannian connection of $\mathscr{Z}$. By using (8.31) in [2, p. 185] we obtain

$$\nabla^2 h(X)[\xi_X, \eta_X] = \langle \mathrm{D}H(X)[\xi_X], \mathrm{D}H(X)[\eta_X]\rangle + \langle H(X), \nabla^2 H(X)[\xi_X, \eta_X]\rangle, \tag{2.9}$$

for all $X := (\mathbf{c}, Q, \Lambda) \in \mathscr{Z}$ and $\xi_X, \eta_X \in T_X\mathscr{Z}$, where $\nabla^2 h$ is a (0, 2)-tensor field and $\nabla^2 H(X)[\cdot,\cdot] = \big[\nabla^2 H_{ij}(X)[\cdot,\cdot]\big] \in \mathbb{SR}^{n\times n}$ [2, p. 109]. The Riemannian Hessian Hess $h(X)$ at a point $X \in \mathscr{Z}$ is determined by

$$\nabla^2 h(X)[\xi_X, \eta_X] = \langle \operatorname{Hess} h(X)[\xi_X], \eta_X\rangle, \tag{2.10}$$

for all $\xi_X, \eta_X \in T_X\mathscr{Z}$. In particular, if X_* is a solution of the equation $H(X) = \mathbf{0}_{n\times n}$, then we can obtain

$$\operatorname{Hess} h(X_*) = (\mathrm{D}H(X_*))^* \circ \mathrm{D}H(X_*).$$

Based on the above discussion, a geometric inexact Gauss–Newton method for solving Problem (2.3) can be described as follows.

Algorithm 1 A geometric inexact Gauss–Newton method

Step 0 Choose an initial point $X_0 \in \mathscr{Z}$, $\beta, \eta_{\max} \in (0, 1)$, $\sigma \in (0, \frac{1}{2})$. Let $k := 0$.

Step 1 Apply the CG method to finding an approximate solution $\Delta X_k \in T_{X_k}\mathscr{Z}$ of

$$(\mathrm{D}H(X_k))^* \circ \mathrm{D}H(X_k)[\Delta X_k] = -\operatorname{grad} h(X_k) \tag{2.11}$$

such that

$$\|(\mathrm{D}H(X_k))^* \circ \mathrm{D}H(X_k)[\Delta X_k] + (\mathrm{D}H(X_k))^*[H(X_k)]\| \leq \eta_k \|\operatorname{grad} h(X_k)\| \tag{2.12}$$

and

$$\langle \operatorname{grad} h(X_k), \Delta X_k\rangle \leq -\eta_k \langle \Delta X_k, \Delta X_k\rangle, \tag{2.13}$$

where $\eta_k := \min\{\eta_{\max}, \|\operatorname{grad} h(X_k)\|\}$. If (2.12) and (2.13) are not attainable, then let

$$\Delta X_k := -\operatorname{grad} h(X_k).$$

Step 2 Let l_k be the smallest nonnegative integer l such that

$$h\big(R_{X_k}(\beta^l \Delta X_k)\big) - h(X_k) \le \sigma\beta^l \langle \operatorname{grad} h(X_k), \Delta X_k\rangle. \tag{2.14}$$

Set

$$X_{k+1} := R_{X_k}(\beta^{l_k} \Delta X_k).$$

Step 3 Replace k by $k+1$ and go to Step 1.

We point out that, in Step 2 of Algorithm 1, (2.14) is the Armijo condition (see [2, Definition 4.2.2]) and R is a retraction on $\mathscr{X}$, which takes the form of

$$R_{X_k}(\Delta X_k) = \big(\mathbf{c}_k + \Delta\mathbf{c}_k, R^o_{Q_k}(\Delta Q_k), \Lambda_k + \Delta\Lambda_k\big), \tag{2.15}$$

where R^o is a retraction on $\mathscr{O}(n)$, which may be chosen as [2, p. 58]:

$$R^o_Q(\eta_Q) = \mathrm{qf}(Q + \eta_Q), \quad \eta_Q \in T_Q\mathscr{O}(n).$$

Here, $\mathrm{qf}(A)$ denotes the Q factor of the QR decomposition of an invertible matrix $A \in \mathbb{R}^{n\times n}$ as $A = Q\widehat{R}$, where Q belongs to $\mathscr{O}(n)$ and $\widehat{R}$ is an upper triangular matrix with strictly positive diagonal elements. For the retraction R defined by (2.15), there exist two scalars $\nu > 0$ and $\mu_\nu > 0$ such that [2, p. 149]

$$\nu\|\Delta X\| \ge \mathrm{dist}\big(X, R_X(\Delta X)\big), \tag{2.16}$$

for all $X \in \mathscr{X}$ and

$$\Delta X \in T_X\mathscr{X} \quad \text{with } \|\Delta X\| \le \mu_\nu, \tag{2.17}$$

where "dist" means the Riemannian distance on the Riemannian product manifold $(\mathscr{X}, g)$ [2, p. 46]. Of course, one may choose other retractions on $\mathscr{O}(n)$ via the polar decomposition, Givens rotation, Cayley transform, exponential mapping, or singular value decomposition (see for instance [2, p. 58] and [45]).

3 Convergence analysis

In this section, we establish the global convergence and local convergence rate of Algorithm 1.

3.1 Global convergence

For the global convergence of Algorithm 1, we have the following result. The proof follows from Theorem 4.1 in [45]. Thus we omit it here.

Theorem 1 *Any accumulation point X_* of the sequence $\{X_k\}$ (if exists) generated by Algorithm 1 is a stationary point of the cost function h defined in Problem* (2.3).

Remark 1 In Step 2 of Algorithm 1, the Armijo condition (2.14) guarantees that the objective function h is monotone decreasing. As in [2, Corollary 4.3.2], the sequence $\{X_k\}$ generated by Algorithm 1 must have an accumulation point if the level set $\mathscr{L} := \{X \in \mathscr{Z} \mid h(X) \leq h(X_0)\}$ is bounded. In particular, if $m = n$ and the matrices $A_1, A_2, \ldots, A_l$ are linearly independent, then it is easy to check that the level set $\mathscr{L}$ is bounded for any initial point $X_0 \in \mathscr{Z}$ (see for instance [47, Lemma 3.1]).

The search directions $\{\Delta X_k\}$ generated by Algorithm 1 have the following property. The proof is similar to that of Lemma 4.3 in [45]. Thus we omit it here.

Lemma 1 *Let X_* be an accumulation point of the sequence $\{X_k\}$ generated by Algorithm 1. If $\mathrm{D}H(X_*) : T_{X_*}\mathscr{Z} \to T_{H(X_*)}\mathbb{SR}^{n\times n}$ is injective, then there exist five positive constants $\bar{\rho}, \kappa_0, \kappa_1, d_1, d_2 > 0$ such that for all $X_k \in B_{\bar{\rho}}(X_*)$, the linear operator $(\mathrm{D}H(X_k))^* \circ \mathrm{D}H(X_k) : T_{H(X_*)}\mathbb{SR}^{n\times n} \to T_{H(X_*)}\mathbb{SR}^{n\times n}$ is nonsingular,*

$$|||(\mathrm{D}H(X_k))^* \circ \mathrm{D}H(X_k)||| \leq \kappa_0, \qquad |||\big((\mathrm{D}H(X_k))^* \circ \mathrm{D}H(X_k)\big)^{-1}||| \leq \kappa_1, \tag{3.1}$$

and

$$d_1 \,\|\mathrm{grad}\, h(X_k)\| \leq \|\Delta X_k\| \leq d_2 \,\|\mathrm{grad}\, h(X_k)\|, \tag{3.2}$$

where $B_{\bar{\rho}}(X_*) := \{X \in \mathscr{Z} \mid \mathrm{dist}(X, X_*) < \bar{\rho}\}$.

Next, we discuss the global convergence of Algorithm 1. We need the following assumption.

Assumption 1 Suppose the differential $\mathrm{D}H(X_*) : T_{X_*}\mathscr{Z} \to T_{H(X_*)}\mathbb{SR}^{n\times n}$ is injective and X_* is an isolated local minimizer of h defined in Problem (2.3), where X_* is an accumulation point of the sequence $\{X_k\}$ generated by Algorithm 1.

For the global convergence of Algorithm 1 related to an isolated local minima of h, we have the following result. The proof follows from [8, Proposition 1.2.5].

Theorem 2 *Let X_* be an accumulation point of the sequence $\{X_k\}$ generated by Algorithm 1. Suppose that Assumption 1 holds, then $\{X_k\}$ converges to X_*.*

Proof By assumption, X_* is an isolated local minimizer of h. Thus there exists a parameter $\hat{\rho} > 0$ such that X_* is the only stationary point of h in the neighborhood $B_{\hat{\rho}}(X_*)$ and

$$h(X) > h(X_*), \quad \forall X \neq X_*,\ X \in B_{\hat{\rho}}(X_*).$$

Since h is continuously differentiable and X_* is a stationary point of h, i.e., $\mathrm{grad}\ h(X_*) = 0_{X_*}$, we obtain

$$\lim_{\mathrm{dist}(X,X_*)\to 0} \mathrm{grad}\, h(X) = 0_{X_*}. \tag{3.3}$$

By using (3.3), the norm $\|\mathrm{grad}\, h(X)\|$ can be as small as possible if X is sufficiently close to X_*. Thus there exists a scalar $0 < \rho < \min\{\hat{\rho}, \bar{\rho}\}$ such that

$$h\big(X_*\big) \leq h\big(X\big) \quad \text{and} \quad d_2\|\mathrm{grad}\, h(X)\| < \mu_\nu, \quad \forall X \in B_\rho(X_*), \tag{3.4}$$

where μ_ν is defined in (2.17). By assumption, the differential $\mathrm{D}H(X_*)$ is injective. Thus, by Lemma 1 we know that (3.2) holds for $X_k \in B_\rho(X_*)$ since $\rho \leq \bar{\rho}$.

Let

$$\phi(t) := \min_{\{X \mid t \leq \mathrm{dist}(X, X_*) \leq \rho\}} \big\{h(X) - h(X_*)\big\}, \quad \forall t \in [0, \rho].$$

We note that ϕ is a monotonically nondecreasing function of t and thus $\phi(t) > 0$ for all $t \in (0, \rho]$. Using (3.3), for any $\varepsilon \in (0, \rho]$, there exists a constant $r \in (0, \varepsilon]$ such that

$$\mathrm{dist}(X, X_*) + \nu d_2 \|\mathrm{grad}\, h(X)\| < \varepsilon, \tag{3.5}$$

for all $X \in B_r(X_*)$, where ν is defined in (2.16). Define the open set

$$S := \big\{X \mid \mathrm{dist}(X, X_*) < \varepsilon,\ h(X) < h(X_*) + \phi(r)\big\}.$$

It is easy to see that $S \subset B_\rho(X_*)$ since $\varepsilon \in (0, \rho]$.

We claim that if $X_k \in S$ for some k, then $X_{k+1} \in S$. Indeed, by using the definitions of ϕ and S, if $X_k \in S$, then

$$\phi\big(\mathrm{dist}(X_k, X_*)\big) \leq h(X_k) - h(X_*) < \phi(r). \tag{3.6}$$

Since ϕ is monotonically nondecreasing we have $\mathrm{dist}(X_k, X_*) < r$. This, together with (3.5), yields

$$\mathrm{dist}(X_k, X_*) + \nu d_2 \|\mathrm{grad}\, h(X_k)\| < \varepsilon. \tag{3.7}$$

Thus, (3.2) holds for X_k since $X_k \in S \subset B_\rho(X_*)$. Using (2.16), (3.2), and (3.4) we have

$$\begin{aligned}
\mathrm{dist}(X_{k+1}, X_*) &\leq \mathrm{dist}(X_k, X_*) + \mathrm{dist}\big(X_{k+1}, X_k\big) \\
&= \mathrm{dist}(X_k, X_*) + \mathrm{dist}\big(R_{X_k}(\rho^{l_k} \Delta X_k), X_k\big) \\
&\leq \mathrm{dist}(X_k, X_*) + \nu \rho^{l_k} \|\Delta X_k\| \leq \mathrm{dist}(X_k, X_*) + \nu \|\Delta X_k\| \\
&\leq \mathrm{dist}(X_k, X_*) + \nu d_2 \|\mathrm{grad}\, h(X_k)\|.
\end{aligned} \tag{3.8}$$

Since $h(X_{k+1}) \leq h(X_k)$, it follows from (3.6), (3.7), and (3.8) that

$$\mathrm{dist}(X_{k+1}, X_*) < \varepsilon, \qquad h(X_{k+1}) - h(X_*) < \phi(r).$$

By the above inequalities and the definition of the set S we have $X_{k+1} \in S$. This completes the proof of the claim. Also, By using induction to the claim we have $X_i \in S$ for all $i \geq k$ if $X_k \in S$ for some k.

Since X_* is an accumulation point of the sequence $\{X_k\}$, there exists a subsequence $\{X_{k_j}\}$ such that $\lim_{j \to \infty} X_{k_j} = X_*$. Thus there exists an integer $k_{\bar{l}}$ such that $X_{k_{\bar{l}}} \in S$. By using the claim we have $X_k \in S$ for all $k \geq k_{\bar{l}}$. Since $h(X_{k+1}) < h(X_k)$ for all $k \geq k_{\bar{l}}$ and $\lim_{k_j \to \infty} h(X_{k_j}) = h(X_*)$, it follows that

$$\lim_{k \to \infty} h(X_k) = h(X_*). \tag{3.9}$$

Using (3.6), $X_k \in S$ for all $k \geq k_{\bar{l}}$, and (3.9) we have $\lim_{k\to\infty} \phi\big(\operatorname{dist}(X_k, X_*)\big) = 0$. Since ϕ is monotone nondecreasing, it follows that $\lim_{k\to\infty} \operatorname{dist}(X_k, X_*) = 0$ and thus $X_k \to X_*$. The proof is complete. □

3.2 Convergence rate

We discuss the local convergence rate of Algorithm 1. The pullbacks of H and h are defined as $\widehat{H} := H \circ R$ and $\widehat{h} := h \circ R$, where R is the retraction defined in (2.15). In addition, we use $\widehat{H}_X := H \circ R_X$ and $\widehat{h}_X := h \circ R_X$ to denote the restrictions of $\widehat{H}$ and $\widehat{h}$ to the tangent space $T_X\mathscr{Z}$.

Since $R_X(0_X) = X$ [2, Definition 4.1.1], the value of H at X is equal to the value of its pullback $\widehat{H}_X$ at 0_X, i.e.,

$$H(X) = H(R_X(0_X)) = \widehat{H}_X(0_X). \tag{3.10}$$

It follows from (2.3) and (3.10) that

$$h(X) = \widehat{h}_X(0_X). \tag{3.11}$$

Let $\mathrm{Id}_{T_X\mathscr{Z}}$ denote the identity mapping on $T_X\mathscr{Z}$. Then one has $\mathrm{D}R_X(0_X) = \mathrm{Id}_{T_X\mathscr{Z}}$ [2, (4.2)]. Thus the differential of H at X is equal to the differential of its pull back $\widehat{H}_X$ at 0_X, i.e.,

$$\mathrm{D}\widehat{H}_X(0_X) = \mathrm{D}(H \circ R_X)(0_X) = \mathrm{D}H(R_X(0_X)) \circ \mathrm{D}R_X(0_X) = \mathrm{D}H(X), \tag{3.12}$$

for all $X \in \mathscr{Z}$. Using (3.12) we have

$$(\mathrm{D}\widehat{H}_X(0_X))^* = (\mathrm{D}H(X))^*, \tag{3.13}$$

for all $X \in \mathscr{Z}$. For the Riemannian gradient of h and the gradient of its pull back $\widehat{h}$, we have [2, p. 56]

$$\operatorname{grad} h(X) = \operatorname{grad} \widehat{h}_X(0_X), \tag{3.14}$$

for all $X \in \mathscr{Z}$.

Here, we need the following result on mean value inequality for differentiable mapping defined between normed vector spaces.

Lemma 2 [16, Corollary 3.3] *Let V and W be normed vector spaces, U an open subset of V and $F : U \to W$ differentiable on U. Given two vectors $\mathbf{a}, \mathbf{b} \in U$, if the segment between $\mathbf{a}$ and $\mathbf{b}$ is contained in U, then we have*

$$\begin{aligned}&\|F(\mathbf{b}) - F(\mathbf{a}) - \mathrm{D}F(\mathbf{a})[\mathbf{b}-\mathbf{a}]\|_W \\ &\quad \leq \sup_{\theta\in[0,1]} |||\mathrm{D}F(\mathbf{a}+\theta(\mathbf{b}-\mathbf{a})) - \mathrm{D}F(\mathbf{a})||| \cdot \|\mathbf{b}-\mathbf{a}\|_V,\end{aligned}$$

where $\|\cdot\|_V$ and $\|\cdot\|_W$ denote the norms of V and W, respectively.

For the stepsize β^{l_k} in (2.14), we have the following result [33].

Lemma 3 *Let X_* be an accumulation point of the sequence $\{X_k\}$ generated by Algorithm 1. Suppose that Assumption 1 holds and $\|H(X_*)\|_F$ is sufficiently small, then for k sufficiently large, $l_k = 0$ satisfies (2.14).*

Proof By hypothesis, Theorem 1, and Lemma 2 we have

$$\lim_{k\to\infty} X_k = X_* \quad \text{and} \quad \lim_{k\to\infty} \operatorname{grad} h(X_k) = \operatorname{grad} h(X_*) = 0_{X_*}. \tag{3.15}$$

By the definition of η_k in Algorithm 1 and (3.15) we have

$$\lim_{k\to\infty} \eta_k = \lim_{k\to\infty} \min\{\eta_{\max}, \|\operatorname{grad} h(X_k)\|\} = \lim_{k\to\infty} \|\operatorname{grad} h(X_k)\| = 0. \tag{3.16}$$

Since $\mathrm{D}H(X_*)$ is injective, it follows from (3.1) and (3.16) that the conditions (2.12) and (2.13) are satisfied for all k sufficiently large. Thus ΔX_k is an approximate solution to (2.11) for all k sufficiently large. Let ΔX_k^{GN} denote the exact solution to (2.11). Then we have

$$(\mathrm{D}H(X_k))^* \circ \mathrm{D}H(X_k)[\Delta X_k - \Delta X_k^{GN}] = \operatorname{grad} h(X_k) + (\mathrm{D}H(X_k))^* \circ \mathrm{D}H(X_k)[\Delta X_k]. \tag{3.17}$$

From (2.11), (3.10), (3.12), and (3.13), it follows that

$$(\mathrm{D}\widehat{H}_{X_k}(0_{X_k}))^*[\widehat{H}_{X_k}(0_{X_k})] + (\mathrm{D}\widehat{H}_{X_k}(0_{X_k}))^* \circ \mathrm{D}\widehat{H}_{X_k}(0_{X_k})[\Delta X_k^{GN}] = 0_{X_k}. \tag{3.18}$$

Using the definition of least-squares solution, (3.10), and (3.12) we find

$$\Delta X_k^{GN} := \operatorname*{argmin}_{\xi_{X_k} \in T_{X_k}\mathscr{X}} \|\widehat{H}_{X_k}(0_{X_k}) + \mathrm{D}\widehat{H}_{X_k}(0_{X_k})[\xi_{X_k}]\|_F.$$

This implies that

$$\begin{aligned} &\|\widehat{H}_{X_k}(0_{X_k}) + \mathrm{D}\widehat{H}_{X_k}(0_{X_k})[\Delta X_k^{GN}]\|_F \\ &\quad \le \|\widehat{H}_{X_k}(0_{X_k}) + \mathrm{D}\widehat{H}_{X_k}(0_{X_k})[0_{X_k}]\|_F = \|\widehat{H}_{X_k}(0_{X_k})\|_F. \end{aligned} \tag{3.19}$$

By hypothesis, $\mathrm{D}H(X_*)$ is injective. Thus, by using Lemma 1, (2.12), (3.1), and (3.17), we have for all k sufficiently large,

$$\begin{aligned} &\|\Delta X_k - \Delta X_k^{GN}\| \\ &\quad = \left\| \big((\mathrm{D}H(X_k))^* \circ \mathrm{D}H(X_k)\big)^{-1} \big[\operatorname{grad} h(X_k) + (\mathrm{D}H(X_k))^* \circ \mathrm{D}H(X_k)[\Delta X_k]\big] \right\| \\ &\quad \le \left|\!\left|\!\left| \big((\mathrm{D}H(X_k))^* \circ \mathrm{D}H(X_k)\big)^{-1} \right|\!\right|\!\right| \cdot \left\| \operatorname{grad} h(X_k) + (\mathrm{D}H(X_k))^* \circ \mathrm{D}H(X_k)[\Delta X_k] \right\| \\ &\quad \le \kappa_1 \eta_k \|\operatorname{grad} h(X_k)\| \le \kappa_1 \|\operatorname{grad} h(X_k)\|^2 \le \tfrac{\kappa_1}{d_1^2} \|\Delta X_k\|^2. \end{aligned} \tag{3.20}$$

In addition, the adjoint $(\mathrm{D}\widehat{H}_X)^*$ of $\mathrm{D}H_X$ is Lipschitz-continuous at 0_X uniformly in a neighborhood of X_*. That is, there exist three scalars $\kappa_2, \delta_1, \delta_2 > 0$ such that

$$|||(\mathrm{D}\widehat{H}_X(\xi_X))^* - (\mathrm{D}\widehat{H}_X(0_X))^*||| \le \kappa_2\|\xi_X\|, \tag{3.21}$$

for all $X \in B_{\delta_1}(X_*)$ and $\xi_X \in B_{\delta_2}(0_X)$. Let

$$G(X_k) := \widehat{H}_{X_k}(\varDelta X_k) - \widehat{H}_{X_k}(0_{X_k}) - \mathrm{D}\widehat{H}_{X_k}(0_{X_k})[\varDelta X_k]. \tag{3.22}$$

By using Lemma 2, (3.21), and (3.22), we have

$$\|G(X_k)\|_F \le \sup_{\theta\in[0,1]} |||\mathrm{D}\widehat{H}_{X_k}(\theta\varDelta X_k) - \mathrm{D}\widehat{H}_{X_k}(0_{X_k})||| \cdot \|\varDelta X_k\| \le \kappa_2\|\varDelta X_k\|^2, \tag{3.23}$$

for all k sufficiently large.

In the rest of the proof, for simplification, we drop the subindex k. Based on (3.11), (3.14), and (3.22), we have

$$\begin{aligned}
h\big(R_X(\varDelta X)\big) &= \widehat{h}_X(\varDelta X) = \tfrac{1}{2}\|\widehat{H}_X(\varDelta X)\|_F^2 \\
&= \tfrac{1}{2}\|\widehat{H}_X(0_X) + \mathrm{D}\widehat{H}_X(0_X)[\varDelta X] + G(X)\|_F^2 \\
&= \tfrac{1}{2}\|\widehat{H}_X(0_X) + \mathrm{D}\widehat{H}_X(0_X)[\varDelta X]\|_F^2 \\
&\quad + \big\langle \widehat{H}_X(0_X) + \mathrm{D}\widehat{H}_X(0_X)[\varDelta X], G(X)\big\rangle + \tfrac{1}{2}\|G(X)\|_F^2 \\
&= \tfrac{1}{2}\|\widehat{H}_X(0_X)\|_F^2 + \big\langle \widehat{H}_X(0_X), \mathrm{D}\widehat{H}_X(0_X)[\varDelta X]\big\rangle + \tfrac{1}{2}\|G(X)\|_F^2 \\
&\quad + \tfrac{1}{2}\big\langle \mathrm{D}\widehat{H}_X(0_X)[\varDelta X], \mathrm{D}\widehat{H}_X(0_X)[\varDelta X_k]\big\rangle \\
&\quad + \big\langle \widehat{H}_X(0_X) + \mathrm{D}\widehat{H}_X(0_X)[\varDelta X], G(X)\big\rangle \\
&= h(X) + \tfrac{1}{2}\big\langle \operatorname{grad} h(X), \varDelta X\big\rangle + \tfrac{1}{2}\big\langle (\mathrm{D}\widehat{H}_X(0_X))^*[\widehat{H}_X(0_X)], \varDelta X\big\rangle \\
&\quad + \tfrac{1}{2}\big\langle (\mathrm{D}\widehat{H}_X(0_X))^*\mathrm{D}\widehat{H}_X(0_X)[\varDelta X], \varDelta X\big\rangle \\
&\quad + \big\langle \widehat{H}_X(0) + \mathrm{D}\widehat{H}_X(0_X)[\varDelta X], G(X)\big\rangle + \tfrac{1}{2}\|G(X)\|_F^2.
\end{aligned}$$

Using (3.1), (3.10), (3.18), (3.19), (3.20), (3.23), and the above equality, we have for all k sufficiently large,

$$\begin{aligned}
&h\big(R_X(\varDelta X)\big) - h(X) - \tfrac{1}{2}\big\langle \operatorname{grad} h(X), \varDelta X\big\rangle \\
&\quad = \tfrac{1}{2}\big\langle (\mathrm{D}\widehat{H}_X(0_X))^*[\widehat{H}_X(0_X)], \varDelta X\big\rangle + \tfrac{1}{2}\big\langle (\mathrm{D}\widehat{H}_X(0_X))^* \circ \mathrm{D}\widehat{H}_X(0_X)[\varDelta X], \varDelta X\big\rangle \\
&\qquad + \big\langle \widehat{H}_X(0_X) + \mathrm{D}\widehat{H}_X(0_X)[\varDelta X], G(X)\big\rangle + \tfrac{1}{2}\|G(X)\|_F^2 \\
&\quad = \tfrac{1}{2}\big\langle (\mathrm{D}\widehat{H}_X(0_X))^*[\widehat{H}_X(0_X)] + (\mathrm{D}\widehat{H}_X(0_X))^* \circ \mathrm{D}\widehat{H}_X(0_X)[\varDelta X^{GN}], \varDelta X\big\rangle \\
&\qquad + \tfrac{1}{2}\big\langle (\mathrm{D}\widehat{H}_X(0_X))^* \circ \mathrm{D}\widehat{H}_X(0_X)[\varDelta X - \varDelta X^{GN}], \varDelta X\big\rangle \\
&\qquad + \big\langle \widehat{H}_X(0_X) + \mathrm{D}\widehat{H}_X(0_X)[\varDelta X^{GN}], G(X)\big\rangle + \tfrac{1}{2}\|G(X)\|_F^2 \\
&\qquad + \big\langle \mathrm{D}\widehat{H}_X(0_X)[\varDelta X - \varDelta X^{GN}], G(X)\big\rangle \\
&\quad \le 0 + \tfrac{1}{2}|||(\mathrm{D}\widehat{H}_X(0_X))^* \circ \mathrm{D}\widehat{H}_X(0_X)||| \cdot \|\varDelta X - \varDelta X^{GN}]\| \cdot \|\varDelta X\|
\end{aligned}$$

$$
\begin{aligned}
&+\|\widehat{H}_X(0_X)+\mathrm{D}\widehat{H}_X(0_X)[\Delta X^{GN}]\|\cdot\|G(X)\|_F+\tfrac{1}{2}\|G(X)\|_F^2\\
&+|||\mathrm{D}\widehat{H}_X(0_X)|||\cdot\|\Delta X-\Delta X^{GN}]\|\cdot\|G(X)\|_F\\
\le{}&\kappa_2\|H(X)\|_F\cdot\|\Delta X\|^2+\tfrac{1}{2}\tfrac{\kappa_1\kappa_0}{d_1^2}\|\Delta X\|^3+\left(\tfrac{1}{2}\kappa_2^2+\tfrac{\kappa_1\kappa_2}{d_1^2}|||\mathrm{D}\widehat{H}_X(0_X)|||\right)\|\Delta X\|^4.
\end{aligned}
$$

Based on (3.15), if $\|H(X_*)\|_F$ is sufficiently small, then $\|H(X_k)\|_F$ is small enough for all k sufficiently large. By using the above inequality, (2.14) holds with $l_k = 0$ for all k sufficiently large. This completes the proof. □

We now establish the local convergence rate of Algorithm 1.

Theorem 3 *Let X_* be an accumulation point of the sequence $\{X_k\}$ generated by Algorithm 1. Suppose that Assumption 1 holds and $\|H(X_*)\|_F$ is sufficiently small, then the sequence $\{X_k\}$ converges to X_* linearly. Furthermore, if $H(X_*) = \mathbf{0}_{n\times n}$, then the sequence $\{X_k\}$ converges to X_* quadratically.*

Proof By hypothesis and using Lemma 3 we have $X_k \to X_*$ and $X_{k+1} = R_{X_k}(\Delta X_k)$ for all k sufficiently large. Using Lemma 7.4.8 and Lemma 7.4.9 in [2], there exist three scalars $\tau_0, \tau_1, \tau_2 > 0$ such that for all k sufficiently large,

$$
\begin{cases}
\tau_0\mathrm{dist}(X_k, X_*) \le \|\mathrm{grad}\, h(X_k)\| \le \tau_1\mathrm{dist}(X_k, X_*),\\
\|\mathrm{grad}\, h(X_{k+1})\| = \|\mathrm{grad}\, h\big(R_{X_k}(\Delta X_k)\big)\| \le \tau_2\|\mathrm{grad}\,\widehat{h}_{X_k}(\Delta X_k)\|.
\end{cases}
\tag{3.24}
$$

By using Taylor's formula we have for all k sufficiently large,

$$
\begin{aligned}
\mathrm{grad}\,\widehat{h}_{X_k}(\Delta X_k) ={}& \mathrm{grad}\,\widehat{h}_{X_k}(0_{X_k}) + (\mathrm{D}\widehat{H}_{X_k}(0_{X_k}))^*\circ\mathrm{D}\widehat{H}_{X_k}(0_{X_k})[\Delta X_k]\\
&+\mathrm{Hess}\,\widehat{h}_{X_k}(0_{X_k})[\Delta X_k] - (\mathrm{D}\widehat{H}_{X_k}(0_{X_k}))^*\circ\mathrm{D}\widehat{H}_{X_k}(0_{X_k})[\Delta X_k]\\
&+\int_0^1\big(\mathrm{Hess}\,\widehat{h}_{X_k}(t\Delta X_k) - \mathrm{Hess}\,\widehat{h}_{X_k}(0_{X_k})\big)[\Delta X_k]\mathrm{d}t.
\end{aligned}
\tag{3.25}
$$

Since H is twice continuously differentiable, it follows from (2.9) and (2.10) that there exist two scalars $\kappa_3 > 0$ and $\delta_3 > 0$ such that for all $X \in B_{\delta_3}(X_*)$,

$$
|||\mathrm{Hess}\,\widehat{h}_{X_k}(0_{X_k}) - (\mathrm{D}\widehat{H}_{X_k}(0_{X_k}))^*\circ\mathrm{D}\widehat{H}_{X_k}(0_{X_k})||| \le \kappa_3\|H(X_k)\|_F. \tag{3.26}
$$

Furthermore, the Hessian operator Hess $\widehat{h}_X$ is Lipschitz-continuous at 0_X uniformly in a neighborhood of X_*, i.e., there exist three scalars $\kappa_4 > 0$, $\delta_4 > 0$, and $\delta_5 > 0$, such that for all $X \in B_{\delta_4}(X_*)$ and $\xi_X \in B_{\delta_5}(0_X)$, it holds that

$$
|||\mathrm{Hess}\,\widehat{h}_X(\xi_X) - \mathrm{Hess}\,\widehat{h}_X(0_X)||| \le \kappa_4\|\xi_X\|. \tag{3.27}
$$

In addition, H is Lipschitz-continuous in a neighborhood of X_*, i.e., there exist two constants $L > 0$ and $\delta_6 > 0$ such that for all $X, Y \in B_{\delta_6}(X_*)$,

$$
\|H(X) - H(Y)\|_F \le L\mathrm{dist}(X, Y). \tag{3.28}
$$

From Lemma 1, (2.12), (3.24), (3.25), (3.26), and (3.27), we have for k sufficiently large,

$$\begin{aligned}
\tfrac{\tau_0}{\tau_2}\operatorname{dist}(X_{k+1},X_*) &\le \|\operatorname{grad}\widehat{h}_{X_k}(\varDelta X_k)\|\\
&\le \left\|\operatorname{grad}\widehat{h}_{X_k}(0_{X_k}) + (\mathrm{D}\widehat{H}_{X_k}(0_{X_k}))^*\circ \mathrm{D}\widehat{H}_{X_k}(0_{X_k})[\varDelta X_k]\right\|\\
&\quad + \left\|\operatorname{Hess}\widehat{h}_{X_k}(0_{X_k})[\varDelta X_k] - (\mathrm{D}\widehat{H}_{X_k}(0_{X_k}))^*\circ \mathrm{D}\widehat{H}_{X_k}(0_{X_k})[\varDelta X_k]\right\|\\
&\quad + \left\|\int_0^1 \big(\operatorname{Hess}\widehat{h}_{X_k}(t\varDelta X_k) - \operatorname{Hess}\widehat{h}_{X_k}(0_{X_k})\big)[\varDelta X_k]dt\right\|\\
&\le \|\operatorname{grad} h(X_k) + (\mathrm{D}\widehat{H}_{X_k}(0_{X_k}))^*\circ \mathrm{D}\widehat{H}_{X_k}(0_{X_k})[\varDelta X_k]\|\\
&\quad + \kappa_3\|H(X_k)\|_F\cdot\|\varDelta X_k\| + \kappa_4\|\varDelta X_k\|^2\\
&\le \eta_k\|\operatorname{grad} h(X_k)\| + \kappa_3 d_2\|H(X_k)\|_F\cdot\|\operatorname{grad} h(X_k)\| + \kappa_4 d_2^2\|\operatorname{grad} h(X_k)\|^2\\
&\le \kappa_3 d_2\tau_1\|H(X_k)\|_F\operatorname{dist}(X_k,X_*) + (1+\kappa_4 d_2^2)\|\operatorname{grad} h(X_k)\|^2\\
&\le \kappa_3 d_2\tau_1\|H(X_k)\|_F\operatorname{dist}(X_k,X_*) + (1+\kappa_4 d_2^2)\tau_1^2\big(\operatorname{dist}(X_k,X_*)\big)^2. \qquad (3.29)
\end{aligned}$$

Thus,

$$\begin{aligned}
\operatorname{dist}(X_{k+1},X_*) &\le \tfrac{\tau_1\tau_2}{\tau_0}\kappa_3 d_2\|H(X_k)\|_F\operatorname{dist}(X_k,X_*) + \tfrac{\tau_1^2\tau_2}{\tau_0}(1+\kappa_4 d_2^2)\big(\operatorname{dist}(X_k,X_*)\big)^2\\
&= c_1\|H(X_k)\|_F\operatorname{dist}(X_k,X_*) + c_2\big(\operatorname{dist}(X_k,X_*)\big)^2,
\end{aligned}$$

where $c_1 := \frac{\tau_1\tau_2}{\tau_0}\kappa_3 d_2$ and $c_2 := \frac{\tau_1^2\tau_2}{\tau_0}(1+\kappa_4 d_2^2)$. If $\|H(X_*)\|_F$ is sufficiently small, then $\|H(X_k)\|_F$ is small enough such that $c_1\|H(X_k)\|_F < 1$ for all k sufficiently large. Thus if $\|H(X_*)\|$ is sufficiently small, then$\{X_k\}$ converges to X_* linearly.

If $H(X_*) = \mathbf{0}_{n\times n}$, then we have from (3.28) for all k sufficiently large,

$$\|H(X_k)\|_F = \|H(X_k) - H(X_*)\|_F \le L\operatorname{dist}(X_k,X_*). \qquad (3.30)$$

Using (3.29) and (3.30), we have

$$\operatorname{dist}(X_{k+1},X_*) \le \tfrac{\tau_1\tau_2}{\tau_0}\big(\kappa_3 d_2 L + (1+\kappa_4 d_2^2)\tau_1\big)\big(\operatorname{dist}(X_k,X_*)\big)^2.$$

Therefore, if $H(X_*) = \mathbf{0}_{n\times n}$, then $\{X_k\}$ converges to X_* quadratically. This completes the proof. □

As a direct consequence of (3.24) and (3.29), we have the following result.

Corollary 1 *Let X_* be an accumulation point of the sequence $\{X_k\}$ generated by Algorithm 1. Suppose the assumptions in Theorem 3 are satisfied. Then there exist two constants $\mu_1,\mu_2 > 0$ such that for all k sufficiently large,*

$$\|\operatorname{grad} h(X_{k+1})\| \le \mu_1\|H(X_k)\|_F\|\operatorname{grad} h(X_k)\| + \mu_2\|\operatorname{grad} h(X_k)\|^2.$$

Furthermore, if $H(X_*) = \mathbf{0}_{n\times n}$, *then there exists a scalar* $\bar{\nu} > 0$ *such that for all k sufficiently large,*

$$\|\mathrm{grad}\, h(X_{k+1})\| \le \bar{\nu}\|\mathrm{grad}\, h(X_k)\|^2.$$

3.3 Injectivity condition

We provide the injectivity condition of $\mathrm{D}H(X_*)$, where $X_* = (\mathbf{c}_*, Q_*, \Lambda_*)$ is an accumulation point of the sequence $\{X_k\}$ generated by Algorithm 1. Based on (2.4), $\mathrm{D}H(X_*)$ is injective if and only if the following matrix equation

$$\begin{cases} (A(\Delta\mathbf{c}) - A_0) + Q_*\overline{\Lambda}_*Q_*^T\Delta Q Q_*^T - \Delta Q\overline{\Lambda}_*Q_*^T - (Q_*P)\Delta\Lambda(Q_*P)^T = \mathbf{0}_{n\times n}, \\ \text{s.t.} \quad (\Delta\mathbf{c}, \Delta Q, \Delta\Lambda) \in T_{X_*}\mathscr{Z}, \end{cases} \tag{3.31}$$

has a unique solution $(\Delta\mathbf{c}, \Delta Q, \Delta\Lambda) = (\mathbf{0}_l, \mathbf{0}_{n\times n}, \mathbf{0}_{(n-m)\times(n-m)}) \in T_{X_*}\mathscr{Z}$, where $\overline{\Lambda}_* := \mathrm{blkdiag}\left(\Lambda_m^*, \Lambda_*\right)$ and P is defined in (2.5).

For $W \in \mathbb{R}^{n\times n}$ define $\widehat{\mathrm{vec}}(W) \in \mathbb{R}^{\frac{n(n-1)}{2}}$ by

$$\widehat{\mathrm{vec}}(W)_{\frac{(j-1)(j-2)}{2}+i} := W_{ij}, \quad 1 \le i < j \le n.$$

This shows that $\widehat{\mathrm{vec}}(W)$ is a column vector obtained by stacking the strictly upper triangular part of W. For $\mathbf{w} \in \mathbb{R}^{\frac{n(n-1)}{2}}$ define $\widehat{\mathrm{skew}}(\mathbf{w}) \in \mathbb{R}^{n\times n}$ by

$$\widehat{\mathrm{vec}}\Big(\widehat{\mathrm{skew}}(\mathbf{w})\Big) := \mathbf{w}, \qquad \widehat{\mathrm{vec}}\Big(\big(\widehat{\mathrm{skew}}(\mathbf{w})\big)^T\Big) := -\mathbf{w},$$

and

$$\big(\widehat{\mathrm{skew}}(\mathbf{w})\big)_{ii} = 0, \quad i = 1, 2, \ldots, n.$$

We observe that $\widehat{\mathrm{skew}}(\mathbf{w})$ is a skew-symmetric matrix constructed from $\mathbf{w}$. Therefore, $\widehat{\mathrm{vec}}$ and $\widehat{\mathrm{skew}}$ are a pair of inverse operators. In addition, there exists a matrix $\widehat{P} \in \mathbb{R}^{n^2\times\frac{n(n-1)}{2}}$ such that

$$\mathrm{vec}\big(\widehat{\mathrm{skew}}(\mathbf{w})\big) = \widehat{P}\mathbf{w} \tag{3.32}$$

for all $\mathbf{w} \in \mathbb{R}^{\frac{n(n-1)}{2}}$. Since $\Delta Q \in T_{Q_*}\mathscr{O}(n)$, there exists a skew-symmetric matrix $\Delta\Omega \in \mathbb{R}^{n\times n}$ such that $\Delta Q = Q_*\Delta\Omega$. For $\Delta\Omega \in \mathbb{R}^{n\times n}$, it follows from (3.32) that there exists a vector $\Delta\mathbf{v} \in \mathbb{R}^{\frac{n(n-1)}{2}}$ such that $\mathrm{vec}\big(\Delta\Omega\big) = \widehat{P}\Delta\mathbf{v}$. Thus, we have

$$\mathrm{vec}(\Delta Q) = \mathrm{vec}(Q_*\Delta\Omega) = (I_n \otimes Q_*)\mathrm{vec}(\Delta\Omega) = (I_n \otimes Q_*)\widehat{P}\Delta\mathbf{v}, \tag{3.33}$$

where "⊗" means the Kronecker product [7, Definition 7.1.2]. Let $\widehat{A}$ be an $n^2 \times l$ matrix defined by

$$\widehat{A} := \big[\mathrm{vec}(A_1), \mathrm{vec}(A_2), \ldots, \mathrm{vec}(A_l)\big] \in \mathbb{R}^{n^2\times l}. \tag{3.34}$$

Since $\varDelta\varLambda \in \mathscr{D}(n-m)$, there exists a matrix $G \in \mathbb{R}^{(n-m)^2\times(n-m)}$ and a vector $\varDelta\mathbf{w} \in \mathbb{R}^{n-m}$ such that

$$\mathrm{vec}(\varDelta\varLambda) = G\varDelta\mathbf{w}. \tag{3.35}$$

Based on (3.33), (3.34), and (3.35), the vectorization of the matrix equation (3.31) is given by

$$\left[\widehat{A},\ (Q_*\otimes Q_*)(I_n\otimes\overline{\varLambda}-\overline{\varLambda}\otimes I_n)\widehat{P},\ (Q_*P)\otimes(Q_*P)G\right]\begin{bmatrix}\varDelta\mathbf{c}\\ \varDelta\mathbf{v}\\ \varDelta\mathbf{w}\end{bmatrix} = \mathbf{0}_{n^2}.$$

Based on the above analysis, we have the following injectivity condition of $\mathrm{D}H(X_*)$.

Theorem 4 *Let $X_* = (\mathbf{c}_*, Q_*, \varLambda_*)$ be an accumulation point of the sequence $\{X_k\}$ generated by Algorithm 1. Then $\mathrm{D}H(X_*)$ is injective if and only if the following matrix*

$$\left[\widehat{A},\ (Q_*\otimes Q_*)(I_n\otimes\overline{\varLambda}_*-\overline{\varLambda}_*\otimes I_n)\widehat{P},\ (Q_*P)\otimes(Q_*P)G\right]$$

is of full rank.

3.4 Preconditioning technique

We propose a preconditioner for solving (2.11). Here we adapt a centered preconditioner [34, p. 279]. For the CG method, instead of solving (2.11), we solve the following preconditioned linear system

$$\begin{cases}(\mathrm{D}H(X_k))^*\circ M_k^{-1}\circ\mathrm{D}H(X_k)[\varDelta X_k] = -(\mathrm{D}H(X_k))^*\circ M_k^{-1}[H(X_k)],\\ \text{s.t.}\quad \varDelta X_k\in T_{(\mathbf{c},Q,\varLambda)}\mathscr{Z},\end{cases}$$

where $M_k : T_{H(X_k)}\mathbb{SR}^{n\times n} \to T_{H(X_k)}\mathbb{SR}^{n\times n}$ is a self-adjoint and positive definite linear operator.

According to (2.4) and (2.6), for $\varDelta Z_k \in T_{H(X_k)}\mathbb{SR}^{n\times n}$, we have

$$\begin{aligned}&\big(\mathrm{D}H(X_k)\circ(\mathrm{D}H(X_k))^* + \hat{t}\,\mathrm{Id}_{T_{H(X_k)}\mathbb{SR}^{n\times n}}\big)[\varDelta Z_k]\\ &\quad= (A(\mathbf{v}(\varDelta Z_k))-A_0) + \big[Q_k\overline{\varLambda}_kQ_k^T, [Q_k\overline{\varLambda}_kQ_k^T, \varDelta Z_k]\big]\\ &\qquad+ Q_kP\,\mathrm{Diag}\big(P^TQ_k^T\varDelta Z_kQ_kP\big)P^TQ_k^T + \hat{t}\varDelta Z_k,\end{aligned} \tag{3.36}$$

where $\hat{t} > 0$ is a given constant, and $\mathrm{Id}_{T_{H(X_k)}\mathbb{SR}^{n\times n}}$ means the identity mapping on $T_{H(X_k)}\mathbb{SR}^{n\times n}$. Replacing the matrix $\mathrm{Diag}\big(P^TQ_k^T\varDelta Z_kQ_kP\big)$ in (3.36) by $P^TQ_k^T\varDelta Z_kQ_kP$, we can construct a centered preconditioner M_k in the following form:

$$
\begin{aligned}
M_k[\Delta Z_k] := (A(\mathbf{v}(\Delta Z_k)) - A_0) + (Q_k\overline{\Lambda}_k^2 Q_k^T)\Delta Z_k - 2(Q_k\overline{\Lambda}_k Q_k^T)\Delta Z_k(Q_k\overline{\Lambda}_k Q_k^T)\\
+\ \Delta Z_k(Q_k\overline{\Lambda}_k^2 Q_k^T) + Q_k P P^T Q_k^T \Delta Z_k Q_k P P^T Q_k^T + \hat{t}\Delta Z_k, \quad (3.37)
\end{aligned}
$$

for all $\Delta Z_k \in T_{H(X_k)}\mathbb{SR}^{n\times n}$. Based on (3.36) and (3.37), the constructed preconditioner M_k is an approximation of $\mathrm{D}H(X_k) \circ (\mathrm{D}H(X_k))^* + \hat{t}\mathrm{Id}_{T_{H(X_k)}\mathbb{SR}^{n\times n}}$.

Given four matrices $A, B, C, D \in \mathbb{R}^{n\times n}$, we have the following properties of the Kronecker product [7, pp. 400–401]:

$$
\begin{cases}
\mathrm{vec}(ABC) = (C^T \otimes A)\mathrm{vec}(B), \quad (A \otimes B)(C \otimes D) = (AC) \otimes (BD),\\
(A + B) \otimes C = A \otimes C + B \otimes C, \quad C \otimes (A + B) = C \otimes A + C \otimes B,\\
(A \otimes B) \otimes C = A \otimes (B \otimes C).
\end{cases} \quad (3.38)
$$

Using (3.34) we find

$$
\widehat{A}\widehat{A}^T = \sum_{i=1}^{l} \mathrm{vec}(A_i)\mathrm{vec}(A_i)^T. \quad (3.39)
$$

From (2.7), (3.37), (3.38), (3.39) we have for any $\Delta Z_k \in T_{H(X_k)}\mathbb{SR}^{n\times n}$,

$$
\begin{aligned}
&\mathrm{vec}\big(M_k[\Delta Z_k]\big)\\
&\quad= \mathrm{vec}\Big(\sum_{i=1}^{l} \mathrm{tr}(A_i^T\Delta Z_k)\cdot A_i\Big) + \big((Q_k I_n Q_k^T) \otimes (Q_k\overline{\Lambda}_k^2 Q_k^T)\big)\mathrm{vec}(\Delta Z_k)\\
&\qquad - 2\big((Q_k\overline{\Lambda}_k Q_k^T)^T \otimes (Q_k\overline{\Lambda}_k Q_k^T)\big)\mathrm{vec}(\Delta Z_k)\\
&\qquad + \big((Q_k\overline{\Lambda}_k^2 Q_k^T)^T \otimes (Q_k I_n Q_k^T)\big)\mathrm{vec}(\Delta Z_k)\\
&\qquad + \big((Q_k P P^T Q_k^T)^T \otimes Q_k P P^T Q_k^T\big)\mathrm{vec}(\Delta Z_k) + \hat{t} I_{n^2}\mathrm{vec}(\Delta Z_k)\\
&\quad= \sum_{i=1}^{l} \big(\mathrm{vec}(A_i)\mathrm{vec}(A_i)^T\big)\mathrm{vec}(\Delta Z_k)\\
&\qquad + \Big((Q_k \otimes Q_k)\big(I_n \otimes \overline{\Lambda}_k^2 - 2\overline{\Lambda}_k \otimes \overline{\Lambda}_k + \overline{\Lambda}_k^2 \otimes I_n\big)(Q_k^T \otimes Q_k^T)\Big)\mathrm{vec}(\Delta Z_k)\\
&\qquad + \big((Q_k \otimes Q_k)(P P^T \otimes P P^T)(Q_k^T \otimes Q_k^T)\big)\mathrm{vec}(\Delta Z_k)\\
&\qquad + \big((Q_k \otimes Q_k)(\hat{t} I_{n^2})(Q_k^T \otimes Q_k^T)\big)\mathrm{vec}(\Delta Z_k)\\
&\quad= (Q_k \otimes Q_k)\big((I_n \otimes \overline{\Lambda}_k - \overline{\Lambda}_k \otimes I_n)^2 + (P P^T) \otimes (P P^T)\\
&\qquad + \hat{t} I_{n^2}\big)(Q_k^T \otimes Q_k^T)\mathrm{vec}(\Delta Z_k) + \widehat{A}\widehat{A}^T\mathrm{vec}(\Delta Z_k), \quad (3.40)
\end{aligned}
$$

where the equality $I_n = Q_k I_n Q_k^T$ is used. Let

$$
\widehat{B}_k := (Q_k \otimes Q_k)\big((I_n \otimes \overline{\Lambda}_k - \overline{\Lambda}_k \otimes I_n)^2 + (P P^T) \otimes (P P^T) + \hat{t} I_{n^2}\big)(Q_k^T \otimes Q_k^T) \quad (3.41)
$$

and

$$
\widehat{M}_k := \widehat{B}_k + \widehat{A}\widehat{A}^T. \quad (3.42)
$$

From (3.40) and (3.42) we have for $\varDelta Z_k \in T_{H(X_k)}\mathbb{SR}^{n\times n}$,

$$\mathrm{vec}\big(M_k[\varDelta Z_k]\big) = \widehat{M}_k \mathrm{vec}(\varDelta Z_k) = \big(\widehat{B}_k + \widehat{A}\widehat{A}^T\big)\mathrm{vec}(\varDelta Z_k). \tag{3.43}$$

Given two nonsingular matrices $C_1 \in \mathbb{R}^{n\times n}$ and $C_2 \in \mathbb{R}^{m\times m}$, we have

$$(C_1 \otimes C_2)^{-1} = C_1^{-1} \otimes C_2^{-1}. \tag{3.44}$$

By using the definition of the matrix P in (2.4) we have

$$PP^T = \mathrm{blkdiag}(0_m, I_{n-m}).$$

Thus the matrix $(I_n \otimes \overline{\varLambda}_k - \overline{\varLambda}_k \otimes I_n)^2 + (PP^T)\otimes(PP^T) + \hat{t} I_{n^2}$ is a diagonal matrix with positive diagonal entries. Then it follows that the matrix $\widehat{B}_k$ is invertible. Using (3.41), and (3.44) we have

$$\widehat{B}_k^{-1} = (Q_k \otimes Q_k)\big((I_n \otimes \overline{\varLambda}_k - \overline{\varLambda}_k \otimes I_n)^2 + (PP^T)\otimes(PP^T) + \hat{t} I_{n^2}\big)^{-1}(Q_k^T \otimes Q_k^T).$$

Thus $\widehat{B}_k^{-1}$ can be easily computed.

By (3.39), the matrix $\widehat{A}\widehat{A}^T$ is a low rank matrix, i.e., $\mathrm{rank}(\widehat{A}\widehat{A}^T) \le l$. By assumption, $l < m \le n < n^2$, thus $\widehat{M}_k$ is a low rank perturbation of $\widehat{B}_k$. By using the Sherman–Morrison–Woodbury formula [28] we find

$$\widehat{M}_k^{-1} = \big(\widehat{B}_k + \widehat{A}\widehat{A}^T\big)^{-1} = \widehat{B}_k^{-1} - \widehat{B}_k^{-1}\widehat{A}\big(I_l + \widehat{A}^T\widehat{B}_k^{-1}\widehat{A}\big)^{-1}\widehat{A}^T\widehat{B}_k^{-1}. \tag{3.45}$$

We conclude from (3.45) that, for any vector $\mathbf{x} \in \mathbb{R}^{n^2}$, the matrix-vector product $\widehat{M}_k^{-1}\mathbf{x}$ can be computed efficiently, where the main computational cost is to calculate the inverse of $(I_l + \widehat{A}^T\widehat{B}_k^{-1}\widehat{A}) \in \mathbb{R}^{l\times l}$.

4 Numerical experiments

In this section we report the numerical performance of Algorithm 1 for solving the PLSIEP II. All the numerical tests are carried out by using `MATLAB` 7.1 running on a workstation with a Intel Xeon CPU E5-2687W at 3.10 GHz and 32 GB of RAM. To illustrate the efficiency of our algorithm, we compare Algorithm 1 with the LP method in [10].

In our numerical tests, we set $\beta = 0.5$, $\eta_{\max} = 0.01$, $\sigma = 10^{-4}$, and $\hat{t} = 10^{-5}$. The largest number of iterations in Algorithm 1 and the LP method is set to be 30000, and the largest number of iterations in the CG method is set to be n^3. Let '`CT.`', '`IT.`', '`NF.`', '`NCG.`', '`Res.`', '`grad.`', and '`err-c.`' denote the averaged total computing time in seconds, the averaged number of outer Gauss–Newton iterations or LP iterations, the averaged number of function evaluations, the averaged total number of inner CG iterations, the averaged residual $\|H(X_k)\|_F$, the averaged residual $\|\mathrm{grad}\, h(X_k)\|$, and the averaged relative error $\|\mathbf{c}_k - \widehat{\mathbf{c}}\|_\infty/\|\widehat{\mathbf{c}}\|_\infty$ at the final iterates of the corresponding algorithms, accordingly. Here, $\|\cdot\|_\infty$ means the vector ∞-norm.

The stopping criterion for Algorithm 1 and the LP method is set to be

$$\|\mathrm{grad}\, h(X_k)\| < \zeta,$$

where $\zeta > 0$ is the prescribed tolerance.

We consider the following two examples.

Example 1 We consider the Sturm–Liouville problem of the form:

$$-\frac{d^2y}{dx^2} + q(x)y = \lambda y, \quad 0 \leq x \leq \pi, \tag{4.1}$$

where q is a real, square-integrable function and the following Dirichlet boundary conditions are imposed

$$y(0) = y(\pi) = 0.$$

By using the Rayleigh–Ritz method in [29], the Rayleigh quotient of (4.1) is given by

$$R\big(y(x)\big) = \frac{\int_0^\pi \big((y'(x))^2 + q(x)y(x)^2\big)\mathrm{d}x}{\int_0^\pi y(x)^2\mathrm{d}x}.$$

Suppose that $y(x) = \sum_{j=1}^n w_j \sin(jx)$. By simple calculation, we have

$$R\big(y(x)\big) = \frac{\sum_{i=1}^n \sum_{j=1}^n i \cdot j \cdot w_i \cdot w_j \cdot \delta_j^i + \frac{2}{\pi} \cdot \sum_{i=1}^n \sum_{j=1}^n w_i \cdot w_j \int_0^\pi q(x)\sin(ix)\sin(jx)\mathrm{d}x}{\sum_{i=1}^n \sum_{j=1}^n w_i \cdot w_j \cdot \delta_j^i},$$

i.e.,

$$R\big(y(x)\big) = \frac{\mathbf{w}^T A\mathbf{w}}{\mathbf{w}^T\mathbf{w}},$$

where $\mathbf{w} := (w_1, w_2, \ldots, w_n)^T$ and the entries of the symmetric matrix $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ are given by

$$\begin{aligned} a_{ij} &= i \cdot j \cdot \delta_j^i + \frac{2}{\pi} \cdot \int_0^\pi q(x)\sin(ix)\sin(jx)\mathrm{d}x \\ &= i \cdot j \cdot \delta_j^i + \frac{2}{\pi}\int_0^\pi q(x)\frac{\cos\big((i-j)x\big) - \cos\big((i+j)x\big)}{2}\mathrm{d}x, \end{aligned}$$

for $i, j = 1, 2, \ldots, n$. If $q(x) = 2\sum_{k=1}^{l} c_k \cos(2kx)$, then one has

$$a_{ij} = i \cdot j \cdot \delta_j^i + \sum_{k=1}^{l} c_k \cdot \left(\delta_{|i-j|}^{2k} - \delta_{i+j}^{2k}\right), \quad i, j = 1, 2, \ldots, n.$$

Let T_k $(k = 1, 2, \ldots, n-1)$ and H_k $(k = 1, 2, \ldots, 2n-1)$ be $n \times n$ real matrices generated by the MATLAB built-in functions toeplitz and hankel:

$$T_k = \texttt{toeplitz}(\mathbf{e}_{k+1}), \quad k = 1, 2, \ldots, n-1$$

and

$$H_k = \begin{cases} \texttt{hankel}(\mathbf{e}_k, \mathbf{0}_n), & k = 1, 2, \ldots, n, \\ \texttt{hankel}(\mathbf{0}_n, \mathbf{e}_{k-n+1}), & k = n+1, n+2, \ldots, 2n-1. \end{cases}$$

Define

$$A_0 = \operatorname{diag}(1, 2^2, 3^2, \ldots, n^2), \quad A_k = \begin{cases} T_{2k} - H_{2k-1}, & 1 \le k \le \min\left\{l, \frac{n-1}{2}\right\}, \\ -H_{2k-1}, & \min\left\{l, \frac{n-1}{2}\right\} < k \le l. \end{cases}$$

Then

$$A = A_0 + \sum_{k=1}^{l} c_k A_k \equiv A(\mathbf{c}).$$

To estimate the first l Fourier coefficients of the potential $q(x)$ defined by [29]

$$q(x) = \sum_{k=1}^{\infty} \frac{192}{\pi^4} \frac{1}{k^4} \cos(2kx),$$

we consider the PLSIEP with above $\{A_k\}$ and the n eigenvalues of $A(\widehat{\mathbf{c}})$ as the prescribed spectrum for varying $n = m$ and l, where the entries of $\widehat{\mathbf{c}}$ are given by

$$\hat{c}_k = \frac{192}{\pi^4} \frac{1}{k^4}, \quad k = 1, 2, \ldots, l.$$

Example 2 We consider the PLSIEP with varying n, l, and m. Let $\widehat{\mathbf{c}} \in \mathbb{R}^l$ be a random vector and $A_0, A_1, \ldots, A_l$ be $n \times n$ random symmetric matrices, which are generated by the MATLAB built-in function randn:

$$\widehat{\mathbf{c}} := \texttt{randn}(l, 1), \quad B_k := \texttt{randn}(n, n), \quad A_k = \frac{1}{2}(B_k + B_k^T), \quad k = 0, 1, \ldots, l.$$

We choose the m smallest eigenvalues of $A(\widehat{\mathbf{c}})$ as the prescribed partial spectrum.

Table 1 Comparison results for Example 1

Algorithm	(n, l, m)	`CT.`	`IT.`	`NF.`	`NCG.`	`Res.`	`grad.`	`err-c.`
LP	(30, 15, 30)	0.54 s	834	835		3.55×10^{-8}	9.98×10^{-7}	1.64×10^{-8}
	(50, 25, 50)	1.77 s	1498	1499		2.42×10^{-8}	9.98×10^{-7}	1.12×10^{-8}
	(100, 40, 100)	11.21 s	3253	3254		1.23×10^{-8}	9.91×10^{-7}	5.23×10^{-9}
	(150, 25, 150)	01 m 12 s	10019	10020		2.37×10^{-10}	9.99×10^{-7}	2.67×10^{-11}
	(200, 30, 200)	05 m 52 s	30000*	30001		5.75×10^{-10}	5.03×10^{-6}	4.83×10^{-11}
Algorithm 1 with CG	(30, 15, 30)	0.12 s	6	7	350	3.14×10^{-11}	1.92×10^{-10}	9.98×10^{-12}
	(50, 25, 50)	0.86 s	6	7	1133	5.37×10^{-12}	2.62×10^{-11}	1.21×10^{-12}
	(100, 40, 100)	12.30 s	6	7	4713	4.12×10^{-11}	3.03×10^{-10}	8.53×10^{-12}
	(150, 25, 150)	50.05 s	6	7	10233	1.34×10^{-10}	2.09×10^{-9}	2.47×10^{-11}
	(200, 30, 200)	02 m 23 s	6	7	18284	3.07×10^{-10}	1.18×10^{-8}	4.71×10^{-11}
Algorithm 1 with PCG	(30, 15, 30)	0.00 s	5	6	1.2	1.17×10^{-12}	2.78×10^{-12}	4.82×10^{-13}
	(50, 25, 50)	0.00 s	5	6	1.2	4.61×10^{-12}	9.72×10^{-12}	1.73×10^{-12}
	(100, 40, 100)	0.09 s	5	6	1.2	4.00×10^{-11}	7.42×10^{-11}	1.59×10^{-11}
	(150, 25, 150)	0.13 s	5	6	1.2	1.36×10^{-10}	1.48×10^{-10}	2.31×10^{-11}
	(200, 30, 200)	0.22 s	5	6	1.2	3.11×10^{-10}	3.31×10^{-10}	4.02×10^{-11}

Table 2 Comparison results for Example 2

Algorithm	(n, l, m)	CT.	IT.	NF.	NCG.	Res.	grad.	err-c.
LP	(20, 10, 18)	0.55 s	1502	1503		7.08×10^{-9}	9.95×10^{-9}	1.97×10^{-9}
	(40, 20, 32)	3.76 s	4773	4774		7.80×10^{-9}	9.99×10^{-9}	1.57×10^{-9}
	(60, 30, 52)	11.66 s	6902	6903		5.08×10^{-9}	9.99×10^{-9}	4.55×10^{-10}
	(70, 40, 62)	20.94 s	11295	11296		5.31×10^{-9}	9.99×10^{-9}	5.08×10^{-10}
	(80, 50, 72)	38.49 s	17056	17057		5.59×10^{-9}	9.99×10^{-9}	5.79×10^{-10}
	(90, 60, 82)	01 m 4 s	22584	22585		5.57×10^{-9}	9.99×10^{-9}	4.51×10^{-10}
Algorithm 1 with CG	(20, 10, 18)	0.04 s	3.2	4.2	266	2.58×10^{-9}	7.76×10^{-10}	6.97×10^{-10}
	(40, 20, 32)	0.36 s	3.6	4.6	997	4.22×10^{-10}	1.95×10^{-9}	2.46×10^{-11}
	(60, 30, 52)	1.49 s	5.6	6.6	1439	1.92×10^{-12}	3.54×10^{-11}	3.45×10^{-14}
	(70, 40, 62)	2.27 s	6.0	7.0	1693	4.75×10^{-12}	3.84×10^{-11}	2.73×10^{-13}
	(80, 50, 72)	4.22 s	6.2	7.2	2267	6.77×10^{-11}	3.30×10^{-10}	2.79×10^{-12}
	(90, 60, 82)	8.47 s	6.8	7.8	3236	3.80×10^{-11}	4.89×10^{-10}	1.55×10^{-12}
Algorithm 1 with PCG	(20, 10, 18)	0.00 s	2.4	3.4	2.2	1.12×10^{-10}	1.46×10^{-9}	3.41×10^{-12}
	(40, 20, 32)	0.01 s	3.0	4.0	6.2	2.67×10^{-13}	3.84×10^{-12}	4.69×10^{-15}
	(60, 30, 52)	0.03 s	4.0	5.0	4.2	3.07×10^{-13}	1.09×10^{-11}	3.64×10^{-15}
	(70, 40, 62)	0.04 s	4.0	5.0	3.8	4.57×10^{-13}	2.20×10^{-11}	6.82×10^{-15}
	(80, 50, 72)	0.06 s	4.0	5.0	3.7	5.53×10^{-13}	3.24×10^{-11}	7.43×10^{-15}
	(90, 60, 82)	0.08 s	4.0	5.0	3.5	5.27×10^{-12}	1.03×10^{-10}	1.10×10^{-14}

Table 3 Numerical results for Example 1

(n, l, m)	CT. (s)	IT.	NF.	NCG.	Res.	grad.	err-c.
(100, 50, 100)	0.09	5	6	1.2	3.24×10^{-11}	5.79×10^{-11}	1.49×10^{-11}
(200, 60, 200)	0.38	5	6	1.2	2.46×10^{-10}	3.42×10^{-10}	6.45×10^{-11}
(300, 70, 300)	1.04	5	6	1.2	5.73×10^{-10}	9.52×10^{-10}	1.61×10^{-10}
(400, 75, 400)	2.36	5	6	1.2	1.78×10^{-9}	1.87×10^{-9}	4.00×10^{-10}
(500, 80, 500)	3.82	5	6	1.2	2.97×10^{-9}	3.14×10^{-9}	4.01×10^{-10}
(600, 85, 600)	5.86	5	6	1.2	4.76×10^{-9}	4.74×10^{-9}	9.18×10^{-10}
(800, 90, 800)	12.03	5	6	1.2	1.12×10^{-8}	9.54×10^{-9}	2.29×10^{-9}

Table 4 Numerical results for Example 2

(n, l, m)	CT. (s)	IT.	NF.	NCG.	Res.	grad.	err-c.
(100, 60, 80)	0.09	3.0	4.0	27	3.04×10^{-11}	4.17×10^{-10}	3.73×10^{-14}
(200, 120, 160)	0.66	3.0	4.0	33	6.86×10^{-11}	1.85×10^{-9}	4.00×10^{-14}
(300, 160, 200)	3.36	3.7	4.7	64	2.14×10^{-11}	1.43×10^{-9}	3.71×10^{-14}
(400, 220, 280)	10.70	4.0	5.0	65	9.42×10^{-12}	2.51×10^{-9}	4.19×10^{-14}
(500, 340, 400)	22.92	4.0	5.0	48	2.48×10^{-11}	5.64×10^{-9}	6.29×10^{-14}
(600, 420, 480)	46.97	4.2	5.2	54	2.13×10^{-11}	9.58×10^{-9}	7.95×10^{-14}

For Algorithm 1 and the LP method in [10], the starting points are generated by the MATLAB built-in function eig:

$$\left[Q_0, \widetilde{\Lambda}\right] = \texttt{eig}\,(A(\mathbf{c}_0), \text{'real'}), \quad \Lambda_0 = \widetilde{\Lambda}(m+1:n).$$

For Example 1, $\mathbf{c}_0$ is set to be a zero vector. For Example 2, $\mathbf{c}_0$ is formed by chopping the components of $\widehat{\mathbf{c}}$ to two decimal places for $n < 100$ and to three decimal places for $n \geq 100$.

We first apply the LP method and Algorithm 1 to Example 1 with $\zeta = 10^{-6}$. Table 1 displays numerical results for Example 1, where the symbol "*" means the largest number of iterations is reached. We observe that Algorithm 1 with PCG works much better than the LP method in terms of computing time.

Next, we apply the LP method and Algorithm 1 to Example 2 with $\zeta = 10^{-8}$. For comparison purposes, we repeat our experiments over 10 different problems. Table 2 shows numerical results for Example 2. We observe from Table 2 that Algorithm 1 is more effective than the LP method in terms of computing time. We also see that the proposed preconditioner can reduce the number of inner CG iterations effectively.

To further illustrate the efficiency of Algorithm 1, we apply Algorithm 1 with the proposed preconditioner to Examples 1 and 2 for varying n, l, m with $\zeta = 10^{-8}$. The corresponding numerical results are displayed in Tables 3 and 4. In Fig. 1, we give the convergence history of Algorithm 1 for two tests. Figure 1 depicts the logarithm of the residual versus the number of outer iterations. We see from Tables 3 and 4 that Algorithm 1 with the proposed preconditioner works very efficient for different values

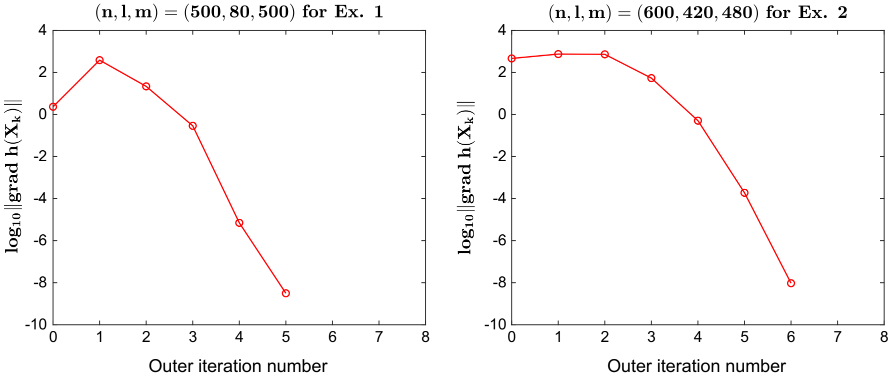

Fig. 1 Convergence history of two tests

of n, l, m. Finally, the linear or quadratic convergence of Algorithm 1 is observed from Fig. 1, which agrees with our prediction.

5 Conclusions

In this paper, we focus on the parameterized least squares inverse eigenvalue problem where the number of parameters is smaller than the number of given eigenvalues. We have proposed a geometric Gauss–Newton method for solving the problem. We have established the global and linear or quadratic convergence of the proposed method under the assumption that the Riemannian differential $\mathrm{D}H(\cdot)$ is injective and the residual $\|H(\cdot)\|_F$ is sufficiently small or equal to zero at an accumulation point X_* of the sequence $\{X_k\}$ generated by our method.

In each outer iteration of the proposed method, the Riemannian Gauss–Newton equation is solved approximately by the CG method. However, the Riemannian Gauss–Newton equation is often ill-conditioned. To improve the efficiency of our method, we have constructed a preconditioner by investigating the special structure of the Riemannian Gauss–Newton equation. Numerical results show that the proposed preconditioner can reduce the number of inner CG iterations and the computational time very effectively. Thus our method can be applied to solve large-scale problems.

Finally, we should point out that the proposed method may not be effective as expected if the differential $\mathrm{D}H(X_*)$ is not injective or the residual $\|H(X_*)\|_F$ is not small enough. This needs further study.

Acknowledgements We are very grateful to the editor and the referees for their valuable comments and suggestions, which have considerably improved this paper.

Appendix A

In this appendix, we deduce (2.4) and (2.6). To derive the Riemannian differential of $H : \mathscr{X} \to \mathbb{SR}^{n\times n}$ defined by (2.2), we consider the following extended mapping

$$\widetilde{H} : \mathbb{R}^l \times \mathbb{R}^{n\times n} \times \mathscr{D}(n-m) \to \mathbb{SR}^{n\times n}, \tag{A.1}$$

which is defined by

$$\widetilde{H}(\mathbf{c}, Q, \Lambda) := A(\mathbf{c}) - Q\overline{\Lambda}Q^T,$$

for all $(\mathbf{c}, Q, \Lambda) \in \mathbb{R}^l \times \mathbb{R}^{n\times n} \times \mathscr{D}(n-m)$, where $\overline{\Lambda}$ is defined as in (2.1). Thus the map H is the restriction of $\widetilde{H}$ from the Euclidean space $\mathbb{R}^l \times \mathbb{R}^{n\times n} \times \mathscr{D}(n-m)$ to the Riemannian product manifold $\mathscr{Z}$, i.e., $H = \widetilde{H}|_{\mathscr{Z}}$.

Similar to [2, (3.17)], for any $(\Delta\mathbf{c}, \Delta Q, \Delta\Lambda) \in T_{(\mathbf{c},Q,\Lambda)}\mathscr{Z}$, one has

$$\mathrm{D}H(\mathbf{c}, Q, \Lambda)[(\Delta\mathbf{c}, \Delta Q, \Delta\Lambda)] = \mathrm{D}\widetilde{H}(\mathbf{c}, Q, \Lambda)[(\Delta\mathbf{c}, \Delta Q, \Delta\Lambda)]. \tag{A.2}$$

For a tangent vector $(\Delta\mathbf{c}, \Delta Q, \Delta\Lambda) \in T_{(\mathbf{c},Q,\Lambda)}\mathscr{Z}$, the matrix $\Delta Q Q^T$ is skew-symmetric [2, (3.26)], i.e.,

$$(\Delta Q Q^T)^T = -\Delta Q Q^T. \tag{A.3}$$

By the definition of P in (2.5) we have

$$\mathrm{blkdiag}(\mathbf{0}_m, \Delta\Lambda) = P\Delta\Lambda P^T. \tag{A.4}$$

Using (2.1), (2.5), (A.1), (A.3), and (A.4) we have

$$\begin{aligned}
&\widetilde{H}(\mathbf{c} + t\Delta\mathbf{c}, Q + t\Delta Q, \Lambda + t\Delta\Lambda)\\
&\quad = A(\mathbf{c} + t\Delta\mathbf{c}) - (Q + t\Delta Q)\,\mathrm{blkdiag}\big(\Lambda_m^*,\ \Lambda + t\Delta\Lambda\big)(Q + t\Delta Q)^T\\
&\quad = A(\mathbf{c} + t\Delta\mathbf{c}) - (Q + t\Delta Q)\overline{\Lambda}(Q + t\Delta Q)^T\\
&\qquad - (Q + t\Delta Q)\,\mathrm{blkdiag}\big(\mathbf{0}_m,\ t\Delta\Lambda\big)(Q + t\Delta Q)^T\\
&\quad = \widetilde{H}(\mathbf{c}, Q, \Lambda) + t(A(\Delta\mathbf{c}) - A_0) + t[Q\overline{\Lambda}Q^T, \Delta Q Q^T]\\
&\qquad - t(QP)\Delta\Lambda(QP)^T + O(t^2),
\end{aligned}$$

where $t \in \mathbb{R}$. Based on Proposition 2.5 in [16] and the above equality, we have

$$\begin{aligned}
&\mathrm{D}\widetilde{H}(\mathbf{c}, Q, \Lambda)[(\Delta\mathbf{c}, \Delta Q, \Delta\Lambda)]\\
&\quad = \lim_{t\to 0} \frac{\widetilde{H}(\mathbf{c} + t\Delta\mathbf{c}, Q + t\Delta Q, \Lambda + t\Delta\Lambda) - \widetilde{H}(\mathbf{c}, Q, \Lambda)}{t}\\
&\quad = (A(\Delta\mathbf{c}) - A_0) + [Q\overline{\Lambda}Q^T, \Delta Q Q^T] - (QP)\Delta\Lambda(QP)^T.
\end{aligned}$$

This, together with (A.2), yields

$$\begin{aligned}
&\mathrm{D}H(\mathbf{c}, Q, \Lambda)[(\Delta\mathbf{c}, \Delta Q, \Delta\Lambda)]\\
&\quad = (A(\Delta\mathbf{c}) - A_0) + [Q\overline{\Lambda}Q^T, \Delta Q Q^T] - (QP)\Delta\Lambda(QP)^T,
\end{aligned} \tag{A.5}$$

for all $(\mathbf{c}, Q, \Lambda) \in \mathscr{Z}$ and $(\Delta\mathbf{c}, \Delta Q, \Delta\Lambda) \in T_{(\mathbf{c},Q,\Lambda)}\mathscr{Z}$, and thus (2.4) holds.

Let $(\mathbf{c}, Q, \Lambda) \in \mathscr{Z}$. For the Riemannian differential $\mathrm{D}H(\mathbf{c}, Q, \Lambda)$ and its adjoint $(\mathrm{D}H(\mathbf{c}, Q, \Lambda))^*$ with respect to the Riemannian metric g, one has [2, p. 185],

$$\begin{aligned} &\mathrm{tr}\big(\Delta Z^T \mathrm{D}H(\mathbf{c}, Q, \Lambda)[(\Delta\mathbf{c}, \Delta Q, \Delta\Lambda)]\big) \\ &\quad = g_{(\mathbf{c},Q,\Lambda)}\big((\mathrm{D}H(\mathbf{c}, Q, \Lambda))^*[\Delta Z], (\Delta\mathbf{c}, \Delta Q, \Delta\Lambda)\big), \end{aligned} \tag{A.6}$$

for any $(\Delta\mathbf{c}, \Delta Q, \Delta\Lambda) \in T_{(\mathbf{c},Q,\Lambda)}\mathscr{Z}$ and $\Delta Z \in T_{H(\mathbf{c},Q,\Lambda)}\mathbb{SR}^{n\times n}$. Using (3.36) in [2], the definition of the linear operator $\mathbf{v}$ in (2.7), (A.5), and (A.6) we have

$$\begin{aligned} &\mathrm{tr}\big(\Delta Z\, \mathrm{D}H(\mathbf{c}, Q, \Lambda)[(\Delta\mathbf{c}, \Delta Q, \Delta\Lambda)]\big) \\ &\quad = \mathrm{tr}\big(\Delta Z(A(\Delta\mathbf{c}) - A_0)\big) - \mathrm{tr}\big(\Delta Z Q\overline{\Lambda}(\Delta Q)^T\big) \\ &\qquad - \mathrm{tr}\big(\Delta Z(\Delta Q)\overline{\Lambda}Q^T\big) - \mathrm{tr}\big(\Delta Z(QP)\Delta\Lambda(QP)^T\big) \\ &\quad = \mathrm{tr}\Big(\Delta Z \sum_{i=1}^{m}(\Delta\mathbf{c})_i A_i\Big) - 2\mathrm{tr}\big((\Delta Z Q\overline{\Lambda})^T \Delta Q\big) - \mathrm{tr}\big((QP)^T \Delta Z(QP)\Delta\Lambda\big) \\ &\quad = \sum_{i=1}^{m}(\Delta \mathrm{c})_i \mathrm{tr}(A_i^T \Delta Z) - 2\mathrm{tr}\big(\big(Q\,\mathrm{skew}(Q^T \Delta Z Q\overline{\Lambda})\big)^T \Delta Q\big) \\ &\qquad - \mathrm{tr}\big((QP)^T \Delta Z(QP)\Delta\Lambda\big) \\ &\quad = \big(\mathbf{v}(\Delta Z)\big)^T \Delta\mathbf{c} + \mathrm{tr}\big(([Q\overline{\Lambda}Q^T, \Delta Z]Q)^T \Delta Q\big) - \mathrm{tr}\big(\mathrm{Diag}\big((QP)^T \Delta Z(QP)\big)\Delta\Lambda\big) \\ &\quad = g_{(\mathbf{c},Q,\Lambda)}\big(\mathbf{v}(\Delta Z), [Q\overline{\Lambda}Q^T, \Delta Z]Q, -\mathrm{Diag}\big((QP)^T \Delta Z(QP)\big), (\Delta\mathbf{c}, \Delta Q, \Delta\Lambda)\big) \\ &\quad = g_{(\mathbf{c},Q,\Lambda)}\big((\mathrm{D}H(\mathbf{c}, Q, \Lambda))^*[\Delta Z], (\Delta\mathbf{c}, \Delta Q, \Delta\Lambda)\big), \end{aligned}$$

where $\mathrm{skew}(A) := \frac{1}{2}(A - A^T)$. Thus,

$$(\mathrm{D}H(\mathbf{c}, Q, \Lambda))^*[\Delta Z] = \big(\mathbf{v}(\Delta Z), [Q\overline{\Lambda}Q^T, \Delta Z]Q, -\mathrm{Diag}\big((QP)^T \Delta Z(QP)\big)\big)$$

for all $\Delta Z \in T_{H(\mathbf{c},Q,\Lambda)}\mathbb{SR}^{n\times n}$. This proves (2.6).

References

1. Absil, P.-A., Baker, C.G., Gallivan, K.A.: Trust-region methods on Riemannian manifolds. Found. Comput. Math. **7**, 303–330 (2007)
2. Absil, P.-A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2008)
3. Absil, P.-A., Malick, J.: Projection-like retractions on matrix manifolds. SIAM J. Optim. **22**, 135–158 (2012)
4. Adler, R.L., Dedieu, J.-P., Margulies, J.Y., Martens, M., Shub, M.: Newton's method on Riemannian manifolds and a geometric model for the human spine. IMA J. Numer. Anal. **22**, 359–390 (2002)
5. Barrett, W., Butler, S., Fallat, S.M., Hall, H.T., Hogben, L., Lin, J.C.-H., Shader, B.L., Young, M.: The inverse eigenvalue problem of a graph: multiplicities and minors. arXiv:1708.00064 (2017)
6. Barrett, W., Nelson, C.G., Sinkovic, J.H., Yang, T.: The combinatorial inverse eigenvalue problem II: all cases for small graphs. Electron. J. Linear Algebra **27**, 742–778 (2014)
7. Bernstein, D.S.: Matrix Mathematics-Theory, Facts, and Formulas, 2nd edn. Princeton University Press, Princeton (2009)

8. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont (1999)
9. Brussaard, P.J., Glaudemans, P.W.M.: Shell Model Applications in Nuclear Spectroscopy. North-Holland Publishing Company, Amsterdam (1977)
10. Chen, X.Z., Chu, M.T.: On the least-squares solution of inverse eigenvalue problems. SIAM J. Numer. Anal. **33**, 2417–2430 (1996)
11. Chu, M.T.: Inverse eigenvalue problems. SIAM Rev. **40**, 1–39 (1998)
12. Chu, M.T., Driessel, K.R.: Constructing symmetric nonnegative matrices with prescribed eigenvalues by differential equations. SIAM J. Math. Anal. **22**, 1372–1387 (1991)
13. Chu, M.T., Golub, G.H.: Structured inverse eigenvalue problems. Acta Numer. **11**, 1–71 (2002)
14. Chu, M.T., Golub, G.H.: Inverse Eigenvalue Problems: Theory, Algorithms, and Applications. Oxford University Press, Oxford (2005)
15. Chu, M.T., Guo, Q.: A numerical method for the inverse stochastic spectrum problem. SIAM J. Matrix Anal. Appl. **19**, 1027–1039 (1998)
16. Coleman, R.: Calculus on Normed Vector Spaces. Springer, New York (2012)
17. Cox, S.J., Embree, M., Hokanson, J.M.: One can hear the composition of a string: experiments with an inverse eigenvalue problem. SIAM Rev. **54**, 157–178 (2012)
18. Dai, H., Bai, Z.Z., Wei, Y.: On the solvability condition and numerical algorithm for the parameterized generalized inverse eigenvalue problem. SIAM J. Matrix Anal. Appl. **36**, 707–726 (2015)
19. Dalmolin, Q., Oliveira, R.: Inverse eigenvalue problem applied to weight optimisation in a geodetic network. Surv. Rev. **43**, 187–198 (2011)
20. Datta, B.N.: Numerical Methods for Linear Control Systems: Design and Analysis. Elsevier Academic Press, London (2003)
21. Dennis, J.E., Schnabel, R.B.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Prentice-Hall, Englewood Cliffs (1983)
22. Friedland, S.: The reconstruction of a symmetric matrix from the spectral data. J. Math. Anal. Appl. **71**, 412–422 (1979)
23. Friedland, S., Nocedal, J., Overton, M.L.: The formulation and analysis of numerical methods for inverse eigenvalue problems. SIAM J. Numer. Anal. **24**, 634–667 (1987)
24. Friswell, M.I., Mottershead, J.E.: Finite Element Model Updating in Structural Dynamics. Kluwer Academic Publishers, Dordrecht (1995)
25. Gladwell, G.M.L.: Inverse Problems in Vibration. Kluwer Academic Publishers, Dordrecht (2004)
26. Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th edn. Johns Hopkins University Press, Baltimore (2013)
27. Gratton, S., Lawless, A.S., Nichols, N.K.: Approximate Gauss–Newton methods for nonlinear least squares problems. SIAM J. Optim. **18**, 106–132 (2007)
28. Hager, W.W.: Updating the inverse of a matrix. SIAM Rev. **31**, 221–239 (1989)
29. Hald, O.H.: The inverse Sturm–Liouville problem and the Rayleigh–Ritz method. Math. Comput. **32**, 687–705 (1978)
30. Helmke, U., Moore, J.B.: Optimization and Dynamical Systems. Springer, London (1994)
31. Hogben, L.: Spectral graph theory and the inverse eigenvalue problem of a graph. Electron. J. Linear Algebra **14**, 12–31 (2005)
32. Horn, R.A., Johnson, C.R.: Topics in Matrix Analysis. Cambridge University Press, Cambridge (1994)
33. Ring, W., Wirth, B.: Optimization methods on Riemannian manifolds and their application to shape space. SIAM J. Optim. **22**, 596–627 (2012)
34. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM, Philadelphia (2003)
35. Smith, S.T.: Optimization techniques on Riemannian manifolds. In: Bloch, A. (ed.) Hamiltonian and Gradient Flows, Algorithms and Control. Fields Institute Communications, vol. 3, pp. 113–136. AMS, Providence (1994)
36. Singh, K.V.: Transcendental inverse eigenvalue problems in damage parameter estimation. Mech. Syst. Signal Process. **23**, 1870–1883 (2009)
37. Singh, K.V., Ram, Y.M.: A transcendental inverse eigenvalue problem associated with longitudinal vibrations in rods. AIAA J. **44**, 317–332 (2006)
38. Tao, G.W., Zhang, H.M., Chang, H.L., Choubey, B.: Inverse eigenvalue sensing in coupled micro/nano system. J. Microelectromech. Syst. **27**, 886–895 (2018)
39. Toman, S., Pliva, J.: Multiplicity of solutions of the inverse secular problem. J. Mol. Spectrosc. **21**, 362–371 (1966)

40. Tropp, J.A., Dhillon, I.S., Heath, R.W.: Finite-step algorithms for constructing optimal CDMA signature sequences. IEEE Trans. Inf. Theory **50**, 2916–2921 (2004)
41. Wang, Z.B., Vong, S.W.: A Gauss-Newton-like method for inverse eigenvalue problems. Int. J. Comput. Math. **90**, 1435–1447 (2013)
42. Wolf, M.M., Perez-Garcia, D.: The inverse eigenvalue problem for quantum channels. arXiv:1005.4545 (2010)
43. Xu, S.F.: An Introduction to Inverse Algebraic Eigenvalue Problems. Peking University Press, Beijing (1998)
44. Yao, T.T., Bai, Z.J., Zhao, Z., Ching, W.K.: A Riemannian Fletcher–Reeves conjugate gradient method for doubly stochastic inverse eigenvalue problems. SIAM J. Matrix Anal. Appl. **37**, 215–234 (2016)
45. Zhao, Z., Bai, Z.J., Jin, X.Q.: A Riemannian Newton algorithm for nonlinear eigenvalue problems. SIAM J. Matrix Anal. Appl. **36**, 752–774 (2015)
46. Zhao, Z., Bai, Z.J., Jin, X.Q.: A Riemannian inexact Newton-CG method for constructing a nonnegative matrix with prescribed realizable spectrum. Numer. Math. **140**, 827–855 (2018)
47. Zhao, Z., Jin, X.Q., Bai, Z.J.: A geometric nonlinear conjugate gradient method for stochastic inverse eigenvalue problems. SIAM J. Numer. Anal. **54**, 2015–2035 (2016)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Teng-Teng Yao[1] · Zheng-Jian Bai[2] · Xiao-Qing Jin[3] · Zhi Zhao[4]

✉ Zheng-Jian Bai
zjbai@xmu.edu.cn

Teng-Teng Yao
yaotengteng718@163.com

Xiao-Qing Jin
xqjin@umac.edu.mo

Zhi Zhao
zhaozhi231@163.com

[1] Department of Mathematics, School of Sciences, Zhejiang University of Science and Technology, Hangzhou 310023, People's Republic of China

[2] School of Mathematical Sciences and Fujian Provincial Key Laboratory on Mathematical Modeling and High Performance Scientific Computing, Xiamen University, Xiamen 361005, People's Republic of China

[3] Department of Mathematics, University of Macau, Macao, People's Republic of China

[4] Department of Mathematics, School of Sciences, Hangzhou Dianzi University, Hangzhou 310018, People's Republic of China