BIT

Riemannian inexact Newton method for structured inverse eigenvalue and singular value problems

Chun-Yueh Chiang[1] · Matthew M. Lin[2] · Xiao-Qing Jin[3]

Abstract

Inverse eigenvalue and singular value problems have been widely discussed for decades. The well-known result is the Weyl-Horn condition, which presents the relations between the eigenvalues and singular values of an arbitrary matrix. This result by Weyl-Horn then leads to an interesting inverse problem, i.e., how to construct a matrix with desired eigenvalues and singular values. In this work, we do that and more. We propose an eclectic mix of techniques from differential geometry and the inexact Newton method for solving inverse eigenvalue and singular value problems as well as additional desired characteristics such as nonnegative entries, prescribed diagonal entries, and even predetermined entries. We show theoretically that our method converges globally and quadratically, and we provide numerical examples to demonstrate the robustness and accuracy of our proposed method.

✉ Matthew M. Lin
   mhlin@mail.ncku.edu.tw

   Chun-Yueh Chiang
   chiang@nfu.edu.tw

   Xiao-Qing Jin
   xqjin@umac.mo

[1]   Center for General Education, National Formosa University, Huwei 632, Taiwan

[2]   Department of Mathematics, National Cheng Kung University, Tainan 701, Taiwan

[3]   Department of Mathematics, University of Macau, Macao, China

Springer

1 Introduction

Let $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n| \geq 0$ and $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$ be the eigenvalues and singular values of a given $n \times n$ real matrix A. In [45] Weyl showed that sets of eigenvalues and singular values satisfy the following necessary condition:

$$\prod_{j=1}^{k} |\lambda_j| \leq \prod_{j=1}^{k} \sigma_j, \quad k = 1, \ldots, n-1, \tag{1.1a}$$

$$\prod_{j=1}^{n} |\lambda_j| = \prod_{j=1}^{n} \sigma_j. \tag{1.1b}$$

Moreover, Horn [29] proved that condition (1.1), called the Weyl-Horn condition, is also sufficient for constructing triangular matrices with prescribed eigenvalues and singular values. Research interest in inverse eigenvalue and singular value problems can be tracked back to the open problem raised by Higham in [28, Problem 26.3], as follows:

> Develop an efficient algorithm for computing a unit upper triangular $n \times n$ matrix with the prescribed singular values $\sigma_1, \ldots, \sigma_n$, where $\prod_{j=1}^{n} \sigma_j = 1$.

This problem, which was solved by Kosowski and Smoktunowicz [32], leads to the following interesting inverse eigenvalue and singular value problem (IESP):

> (**IESP**) Given two sets of numbers $\boldsymbol{\lambda} = \{\lambda_1, \ldots, \lambda_n\}$ and $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_n\}$ satisfying (1.1), find a real $n \times n$ matrix with eigenvalues $\boldsymbol{\lambda}$ and singular values $\boldsymbol{\sigma}$.

The following factors make the IESP difficult to solve:

– Often the desired matrices are real. This problem was solved by the authors of [9] with prescribed real eigenvalues and singular values. The method for finding a general real-valued matrix with prescribed complex–conjugate eigenvalues and singular values was also investigated in [33]. In this work, we take an alternative approach to tackle this problem and add further constraints.
– Often the desired matrices are structured. Corresponding to physical applications, the recovered matrices often preserve some common structure such as nonnegative entries or predetermined diagonal entries [8,46]. In this paper, specifically, we offer the condition of the existence of a nonnegative matrix provided that eigenvalues, singular values, and diagonal entries are given. Furthermore, solving the IESP with respect to the diagonal constraint is not enough because entries of the recovered matrices should preserve certain patterns, for example, non-negativity, which correspond to original observations. How to tackle this structured problem is the main thrust of this paper.

The IESP can be regarded as a natural generalization of the inverse eigenvalue problems, which is known for its a wide variety of applications such as the pole assignment problem [6,20,34], applied mechanics [15,18,19,25,38], and inverse Sturm–Liouville problem [3,24,26,37]. Thus applications of the IESP could be found in wireless communication [17,39,43] and quantum information science [21,30,46]. Research results advanced thus far for the IESP do not fully address the above scenarios. Often, given a set of data, the IESP is studied in parts. That is, there have been extensive investigations of the conditions for the existence of a matrix when the singular values and eigenvalues are provided (i.e., the Weyl-Horn condition [29,45]), when the singular values and main diagonal entries are provided (i.e., the Sing-Thompson condition [41,42]), or when the eigenvalues and main diagonal entries are provided (i.e., the Mirsky condition [36]). Also, the above conditions have given rise to numerical approaches, as found in [5,8,9,16,22,32,48].

Our work here not only provides a way to solve the IESP but also immediately use the solution to address the IESP with the desired structure, for example, nonnegative entries and fixed diagonal elements. For one relatively close result, see [46], where the authors consider a new type of IESP that requires that all three constraints, i.e., eigenvalues, singular values, and diagonal entries, be satisfied simultaneously. Theoretically, Wu and Chu generalize the classical Mirsky, Sing-Thompson, and Weyl-Horn conditions and provide one sufficient condition for the existence of a matrix with prescribed eigenvalues, singular values, and diagonal entries when $n \geq 3$. Numerically, Wu and Chu establish a dynamic system for constructing such a matrix, in which real eigenvalues are given. In this work, we solve an IESP with complex conjugate eigenvalues and with entries fixed at certain locations. Note that, in general, the solution of the IESP is not unique or difficult to find once structured requirements are added. To solve an IESP with some specific feature, we combine techniques from differential geometry and for solving nonlinear equations.

We organize this paper as follows. In Sect. 2, we propose the use of the Riemannian inexact Newton method for solving an IESP with complex conjugate eigenvalues. In Sect. 3, we show that the convergence is quadratic. In Sect. 4, we demonstrate the application of our technique to an IESP with a specific structure that includes nonnegative or predetermined entries to show the robustness and efficiency of our proposed approaches. The concluding remarks are given in Sect. 5.

2 Riemannian inexact Newton method

In this section, we explain how the Riemannian inexact Newton method can be applied to the IESP. The problem of optimizing a function on a matrix manifold has received much attention in the scientific and engineering fields due to its peculiarity and capacity. Its applications include, but are not limited to, the study of eigenvalue problems [1,2,7,10,12–14,46,47,49–51], matrix low rank approximation [4,27], and nonlinear matrix equations [11,44]. Numerical methods for solving problems involving matrix manifolds rely on interdisciplinary inputs from differential geometry, optimization theory, and gradient flows.

To begin, let $\mathscr{O}(n) \subset \mathbb{R}^{n \times n}$ be the group of $n \times n$ real orthogonal matrices, and let $\boldsymbol{\lambda} = \{\lambda_1, \ldots, \lambda_n\}$ and $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_n\}$ be the eigenvalues and singular values of an $n \times n$ matrix. We assume without loss of generality that:

$$\lambda_{2i-1} = \alpha_i + \beta_i \sqrt{-1}, \quad \lambda_{2i} = \alpha_i - \beta_i \sqrt{-1}, \quad i = 1, \ldots, k,$$
$$\lambda_i \in \mathbb{R}, \quad i = 2k+1, \ldots, n,$$

where $\alpha_i, \beta_i \in \mathbb{R}$ with $\beta_i \neq 0$ for $i = 1, \ldots, k$, and we define the corresponding block diagonal matrix

$$\Lambda = \text{diag} \left\{ \begin{bmatrix} \alpha_1 & \beta_1 \\ -\beta_1 & \alpha_1 \end{bmatrix}, \ldots, \begin{bmatrix} \alpha_k & \beta_k \\ -\beta_k & \alpha_k \end{bmatrix}, \lambda_{2k+1}, \ldots, \lambda_n \right\}$$

and the diagonal matrix

$$\Sigma = \text{diag} \{\sigma_1, \ldots, \sigma_n\}.$$

Then the IESP, based on the real Schur decomposition, is equivalent to finding matrices $U, V, Q \in \mathscr{O}(n)$, and

$$W \in \mathscr{W}(n) := \{W \in \mathbb{R}^{n \times n} \mid W_{i,j} = 0 \text{ if } \Lambda_{i,j} \neq 0 \text{ or } i \geq j, \text{ for } 1 \leq i, j \leq n\},$$

which satisfy the following equation:

$$F(U, V, Q, W) = U \Sigma V^\top - Q(\Lambda + W)Q^\top = \mathbf{0}. \tag{2.1}$$

Here, we may assume without loss of generality that Q is an identity matrix and simplify Eq. (2.1) as follows:

$$F(U, V, W) = U \Sigma V^\top - (\Lambda + W) = \mathbf{0}. \tag{2.2}$$

Let $X = (U, V, W) \in \mathscr{O}(n) \times \mathscr{O}(n) \times \mathscr{W}(n)$. Upon using Eq. (2.2), we can see that we might solve the IESP by

$$\text{finding } X \in \mathscr{O}(n) \times \mathscr{O}(n) \times \mathscr{W}(n) \text{ such that } F(X) = \mathbf{0}, \tag{2.3}$$

where $F : \mathscr{O}(n) \times \mathscr{O}(n) \times \mathscr{W}(n) \to \mathbb{R}^{n \times n}$ is continuously differentiable. By making an initial guess, X_0, one immediate way to solve Eq. (2.3) is to apply the Newton method and generate a sequence of iterates by solving

$$DF(X_k)[\Delta X_k] = -F(X_k), \tag{2.4}$$

for $\Delta X_k \in T_{X_k}(\mathscr{O}(n) \times \mathscr{O}(n) \times \mathscr{W}(n))$ and set

$$X_{k+1} = R_{X_k}(\Delta X_k),$$

where $DF(X_k)$ represents the differential of F at X_k and R is a retraction on $\mathscr{O}(n) \times$ $\mathscr{O}(n) \times \mathscr{W}(n)$. Since Eq. (2.4) is an underdetermined system, it may have more than one solution. Let $DF(X_k)^*$ be the adjoint operator of $DF(X_k)$. In our calculation, we choose the solution ΔX_k with the minimum norm by letting [35, Chap. 6]

$$\Delta X_k = DF(X_k)^*[\Delta Z_k], \tag{2.5}$$

where $\Delta Z_k \in T_{F(X_k)}(\mathbb{R}^{n \times n})$ is a solution for

$$\big(DF(X_k) \circ DF(X_k)^*\big)[\Delta Z_k] = -F(X_k). \tag{2.6}$$

Note that the notation $\circ$ represents the composition of two operators $DF(X_k)$ and $DF(X_k)^*$. This implies that the operator $DF(X_k) \circ DF(X_k)^*$ is symmetric and positive semidefinite. If, as is the general case, the operator $DF(X_k) \circ DF(X_k)^*$ from $T_{F(X_k)}(\mathbb{R}^{n \times n})$ to $\mathbb{R}^{n \times n}$ is invertible, we can compute the optimal solution in (2.5).

Note that solving for the solution of Eq. (2.6) could be unnecessary and computationally time-consuming, and that the linear model given by Eq. (2.6) is large-scale or the resulting iteration X_k is far from the solution of condition (2.3) [40]. By analogy with the classical Newton method [23], we adopt the "inexact" Newton method on Riemannian manifolds, i.e., without solving Eq. (2.6) exactly, we repeatedly apply the conjugate gradient (CG) method to find $\Delta Z_k \in T_{F(X_k)}(\mathbb{R}^{n \times n})$, such that:

$$\|(DF(X_k) \circ DF(X_k)^*)[\Delta Z_k] + F(X_k)\| \le \eta_k \|F(X_k)\|, \tag{2.7}$$

for some constant $\eta_k \in [0, 1)$, is satisfied. Then, we update X_k corresponding to ΔZ_k until the stopping criterion is satisfied. Here, the notation $\|\cdot\|$ is the Frobenius norm. Note that in our calculation, the elements in the product space $\mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n}$ are computed using the standard Frobenius inner product:

$$\langle (A_1, A_2, A_3), (B_1, B_2, B_3) \rangle_F := \langle A_1, B_1 \rangle + \langle A_2, B_2 \rangle + \langle A_3, B_3 \rangle, \tag{2.8}$$

where $\langle A, B \rangle := \text{trace}(AB^\top)$ for any $A, B \in \mathbb{R}^{n \times n}$ and the induced norm $\|X\|_F = \sqrt{\langle X, X \rangle_F}$ (or, simply, $\langle X, X \rangle$ and $\|X\|$ without the risk of confusion) for any $X \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n}$.

Then, the mapping $DF(X_k)$ at $\Delta X_k = (\Delta U_k, \Delta V_k, \Delta W_k) \in T_{X_k}(\mathscr{O}(n) \times \mathscr{O}(n) \times \mathscr{W}(n))$ is given by:

$$DF(X_k)[\Delta X_k] = \Delta U_k \Sigma V_k^\top + U_k \Sigma \Delta V_k^\top - \Delta W_k.$$

Let $DF(X_k)^* : T_{F(X_k)}(\mathbb{R}^{n \times n}) \to T_{X_k}(\mathscr{O}(n) \times \mathscr{O}(n) \times \mathscr{W}(n))$ be the adjoint of the mapping $DF(X_k)$. The adjoint $DF(X_k)^*$ is determined by the following:

$$\langle \Delta Z_k, DF(X_k)[\Delta X_k] \rangle = \langle DF(X_k)^*[\Delta Z_k], \Delta X_k \rangle$$

and can be expressed as follows:

$$DF(X_k)^*[\Delta Z_k] = (\Delta U_k, \Delta V_k, \Delta W_k),$$

where

$$\Delta U_k = \frac{1}{2}(\Delta Z_k V_k \Sigma^\top - U_k \Sigma V_k^\top \Delta Z_k^\top U_k),$$

$$\Delta V_k = \frac{1}{2}(\Delta Z_k^\top U_k \Sigma - V_k \Sigma^\top U_k^\top \Delta Z_k V_k),$$

$$\Delta W_k = -H \odot \Delta Z_k,$$

where the notation $\odot$ represents the Hadamard product and H is a matrix satisfying

$$H_{i,j} = \begin{cases} 0 & \text{if } \Lambda_{i,j} \neq 0 \text{ or } i \geq j \\ 1 & \text{otherwise} \end{cases}$$

(see [12,50] for a similar discussion).

There is definitely no guarantee that the application of the inexact Newton method can achieve a sufficient decrease in the size of the nonlinear residual $\|F(X_k)\|$. This provides motivation for deriving an iterate for which the size of the nonlinear residual is decreased. One way to do this is to update the Newton step ΔX_k obtained from Eq. (2.5) by choosing $\theta \in [\theta_{\min}, \theta_{\max}]$, with $0 < \theta_{\min} < \theta_{\max} < 1$, and setting

$$\widehat{\Delta X_k} = \Delta X_k, \quad \hat{\eta}_k = \frac{\|F(X_k) + DF(X_k)[\Delta X_k]\|}{\|F(X_k)\|}, \tag{2.9}$$

and $\eta_k = \hat{\eta}_k$. Note that the definition of $\hat{\eta}_k$ is not well-defined if $F(X_k) = 0$. However, we can prevent this happening by assuming without loss of generality that $F(X_0) \neq 0$. Once $F(X_k) = 0$ is satisfied, we terminate our iterations and output the solution $X = X_k$. Otherwise, we update

$$\eta_k \leftarrow 1 - \theta(1 - \eta_k) \text{ and } \Delta X_k \leftarrow \frac{1 - \eta_k}{1 - \hat{\eta}_k} \widehat{\Delta X_k}, \tag{2.10}$$

while

$$\|F(X_k)\| - \|F(R_{X_k}(\Delta X_k))\| > t(1 - \eta_k)\|F(X_k)\|,$$

or, equivalently,

$$\|F(R_{X_k}(\Delta X_k))\| < [1 - t(1 - \eta_k)]\|F(X_k)\|, \tag{2.11}$$

for some $t \in [0, 1)$ [23]. Let $Q_f(\cdot)$ denote the mapping that sends a matrix to the Q factor of its QR decomposition with its R factor having strictly positive diagonal elements [1, Example 4.1.3]. For all $(\xi_U, \xi_V, \xi_W) \in T_{(U,V,W)}(\mathscr{O}(n) \times \mathscr{O}(n) \times \mathscr{W}(n))$, we can compute the retraction R using the following formula:

$$R_{(U,V,W)}(\xi_U, \xi_V, \xi_W) = (R_U(\xi_U), R_V(\xi_V), R_W(\xi_W)),$$

where

$$R_U(\xi_U) = Q_f(U + \xi_U), \quad R_V(\xi_V) = Q_f(V + \xi_V), \quad R_W(\xi_W) = W + \xi_W.$$

We call this the Riemannian inexact Newton backtracking method (RINB) and formalize this method in Algorithm 1. To choose the parameter $\theta \in [\theta_{min}, \theta_{max}]$, we apply a two-point parabolic model [31,50] to achieve a sufficient decrease among steps 6–9. That is, we use the iteration history to model an approximate minimizer of the following scalar function:

$$f(\lambda) := \|F(R_{X_k}(\lambda \Delta X_k))\|^2$$

by defining a parabolic model, as follows:

$$p(\lambda) = f(0) + f'(0)\lambda + (f(1) - f(0) - f'(0))\lambda^2,$$

where

$$f(0) = \|F(X_k)\|^2,$$
$$f'(0) = 2\langle DF(X_k)[\Delta X_k], F(X_k)\rangle, \text{ and } f(1) = \|F(R_{X_k}(\Delta X_k))\|^2.$$

It should be noted that the computational costs for constructing the parabolic $p(\lambda)$ are only $O(n^3)$ operations per iteration.

From (2.7), it can be shown that the function evaluation $f'(0)$ should be negative. Since $f'(0) < 0$, if $p''(\lambda) = 2(f(1) - f(0) - f'(0)) > 0$, then $p(\lambda)$ has its minimum at:

$$\theta = \frac{-f'(0)}{2(f(1) - f(0) - f'(0))} > 0;$$

otherwise, if $p''(\lambda) < 0$, we choose $\theta = \theta_{max}$. By incorporating two types of selection, we can choose the following:

$$\theta = \min\left\{\max\left\{\theta_{min}, \frac{-f'(0)}{2(f(1) - f(0) - f'(0))}\right\}, \theta_{max}\right\}.$$

as the parameter θ in Algorithm 1 [31,50]. In the next section, we mathematically investigate the convergence analysis of Algorithm 1.

Algorithm 1: The Riemannian inexact Newton backtracking method
$[X] = \text{RINB}(\sigma, X_0)$

Input: An initial value X_0 such that $F(X_0) \neq 0$.
Output: A numerical solution X satisfying $F(X) = 0$

1 **begin**

2   Let $\eta_{\max} \in [0, 1)$, $\eta_0 = \min\{\eta_{\max}, \|F(X_0)\|\}$, and $t \in [0, 1)$, and
$0 < \theta_{\min} < \theta_{\max} < 1$ be given.

3   **repeat**

4   Determine ΔZ_k by using the CG method to (2.6) until (2.7) holds.

5   Set $\Delta X_k = (DF(X_k))^*[\Delta Z_k]$, $\hat{\eta}_k = \frac{\|F(X_k) + DF(X_k)[\Delta X_k]\|}{\|F(X_k)\|}$, $\widehat{\Delta X_k} = \Delta X_k$,
and $\eta_k = \hat{\eta}_k$.

6   **repeat**

7   Choose $\theta \in [\theta_{\min}, \theta_{\max}]$.

8   Update $\eta_k \leftarrow 1 - \theta(1 - \eta_k)$ and $\Delta X_k \leftarrow \frac{1 - \eta_k}{1 - \hat{\eta}_k}\widehat{\Delta X_k}$.

9   **until** (2.11) holds;
10   Set $X_{k+1} = R_{X_k}(\Delta X_k)$ and $\eta_{k+1} = \min\{\eta_k, \eta_{\max}, \|F(X_{k+1})\|\}$.
11   Replace k by $k + 1$.
12   **until** $\|F(X_k)\| < \epsilon$;
13   $X = X_k$.
14 **end**

3 Convergence analysis

By combining the classical inexact Newton method [23] with optimization techniques
on matrix manifolds, Algorithm 1 provides a way to solve the IESP. However, we have
yet to theoretically discuss the convergence analysis of Algorithm 1. In this section,
we provide a theoretical foundation for the RINB method, and show that this RINB
method converges globally and finally converges quadratically when Algorithm 1 does
not terminate prematurely. We address this phenomenon in the following:

Lemma 3.1 *Algorithm 1 does not break down at some X_k if and only if $F(X_k) \neq 0$
and the inverse of $DF(X_k) \circ DF(X_k)^*$ exists.*

Next, we provide an upper bound for the approximate solution $\widehat{\Delta X_k}$ in Algorithm 1.

Theorem 3.1 *Let $\Delta Z_k \in T_{F(X_k)}(\mathbb{R}^{n \times n})$ be a solution that satisfies condition (2.7) and*

$$\widehat{\Delta X_k} = DF(X_k)^*[\Delta Z_k].$$

Then,

$$\text{(a)} \ \|\widehat{\Delta X_k}\| \leq (1 + \hat{\eta}_k)\|DF(X_k)^\dagger\|\|F(X_k)\|, \tag{3.1a}$$

$$\text{(b)} \ \|\sigma_k(\eta)\| \leq \frac{1 + \eta_{\max}}{1 - \eta_{\max}}(1 - \eta)\|DF(X_k)^\dagger\|\|F(X_k)\|, \tag{3.1b}$$

where $DF(X_k)^\dagger$ is the pseudoinverse of $DF(X_k)$, $\hat{\eta}_k$ is defined in Eq. (2.9), and σ_k is the backtracking curve used in Algorithm 1, which is defined by the following:

$$\sigma_k(\eta) = \frac{1 - \eta}{1 - \hat{\eta}_k}\widehat{\Delta X_k}$$

with $\hat{\eta}_k \leq \eta \leq 1$, and

$$\|DF(X_k)^\dagger\| := \max_{\|\Delta Z\| = 1} \|DF(X_k)^\dagger[\Delta Z]\|$$

represents the norm of the pseudoinverse of $DF(X_k)$.

Proof Let $r_k = (DF(X_k) \circ DF(X_k)^*)[\Delta Z_k] + F(X_k)$. We see that

$$\begin{aligned}
\|\widehat{\Delta X_k}\| &\leq \|DF(X_k)^* \circ [DF(X_k) \circ DF(X_k)^*]^{-1}\|\|r_k - F(X_k)\| \\
&\leq (1 + \hat{\eta}_k)\|DF(X_k)^\dagger\|\|F(X_k)\|
\end{aligned}$$

and

$$\begin{aligned}
\|\sigma_k(\eta)\| &= \frac{1 - \eta}{1 - \hat{\eta}_k}\|DF(X_k)^\dagger(r_k - F(X_k))\| \leq \frac{1 + \hat{\eta}_k}{1 - \hat{\eta}_k}(1 - \eta)\|DF(X_k)^\dagger\|\|F(X_k)\| \\
&\leq \frac{1 + \eta_{\max}}{1 - \eta_{\max}}(1 - \eta)\|DF(X_k)^\dagger\|\|F(X_k)\|.
\end{aligned}$$

$\square$

In our subsequent discussion, we assume that Algorithm 1 does not break down and the sequence $\{X_k\}$ is bounded. Let X_* be a limit point of $\{X_k\}$. Since F is continuously differentiable, we have the following:

$$\|DF(X)^\dagger\| \leq 2\|DF(X_*)^\dagger\| \tag{3.2}$$

whenever $X \in B_\delta(X_*)$ for a sufficiently small constant $\delta > 0$. Here, the notation $B_\delta(X_*)$ represents a neighborhood of X_* consisting of all points X such that $\|X -$

$X_*\| < \delta$. By condition (3.1), we can show without any difficulty that whenever X_k is sufficiently close to X_*,

$$\|\widehat{\Delta X_k}\| \leq (1 + \eta_{\max})\|DF(X_*)^\dagger\|\|F(X_k)\|,$$
$$\|\sigma_k(\eta)\| \leq \Gamma(1 - \eta)\|F(X_k)\|, \quad \hat{\eta}_k \leq \eta \leq 1, \tag{3.3}$$

where Γ is a constant independent of k defined by

$$\Gamma = 2\frac{1 + \eta_{\max}}{1 - \eta_{\max}}\|DF(X_*)^\dagger\|.$$

Now, we show that the sequence of $\{F(X_k)\}$ eventually converges to zero.

Theorem 3.2 *Assume that Algorithm 1 does not break down. If $\{X_k\}$ is the sequence generated in Algorithm 1, then*

$$\lim_{k \to \infty} F(X_k) = \mathbf{0}.$$

Proof Observe that

$$\|F(X_k)\| = \|F(R_{X_{k-1}}(\Delta X_{k-1}))\| \leq (1 - t(1 - \eta_{k-1}))\|F(X_{k-1})\|$$
$$\leq \|F(X_0)\| \prod_{j=0}^{k-1}(1 - t(1 - \eta_j)) \leq \|F(X_0)\|e^{-t\sum_{j=0}^{k-1}(1-\eta_j)}.$$

Since $t > 0$ and $\lim_{k \to \infty} \sum_{j=0}^{k-1}(1 - \eta_j) = \infty$, we have $\lim_{k \to \infty} F(X_k) = \mathbf{0}$. □

In our iteration, we implement the repeat loop among steps 6–9 by selecting a sequence $\{\theta_j\}$, with $\theta_j \in [\theta_{\min}, \theta_{\max}]$. For each loop, correspondingly, we let $\eta_k^{(1)} = \hat{\eta}_k$ and $\Delta X^{(1)} = \widehat{\Delta X_k}$, and for $j = 2, \ldots,$ we let

$$\eta_k^{(j)} = 1 - \theta_{j-1}(1 - \eta_k^{(j-1)}),$$
$$\Delta X_k^{(j)} = \frac{1 - \eta_k^{(j)}}{1 - \hat{\eta}_k}\widehat{\Delta X_k}. \tag{3.4}$$

By induction, then, we can easily show that:

$$\Delta X_k^{(j)} = \Theta_{j-1}\widehat{\Delta X_k}, \quad 1 - \eta_k^{(j)} = \Theta_{j-1}(1 - \hat{\eta}_k),$$

where

$$\Theta_{j-1} = \prod_{\ell=1}^{j-1}\theta_\ell, \quad j \geq 2. \tag{3.5}$$

In other words, the sequence $\{\|\Delta X_k^{(j)}\|\}_j$ is a strictly decreasing sequence satisfying $\lim_{j\to\infty} \Delta X_k^{(j)} = \mathbf{0}$, and $\{\eta_k^{(j)}\}_j$ is a sequence satisfying $\eta_k^{(j)} \geq \hat{\eta}_k$ for $j \geq 1$, and $\lim_{j\to\infty} \eta_k^{(j)} = 1$. Based on these observations, next, we show that the repeat loop terminates after a finite number of steps.

Theorem 3.3 *Let $\{\widehat{\Delta X_k}\}$ be the sequence generated from Algorithm 1, i.e.,*

$$\|(DF(X_k)[\widehat{\Delta X_k}] + F(X_k)\| \leq \eta_k \|F(X_k)\|.$$

Then, once j is large enough, the sequence $\{\eta_k^{(j)}\}_j$ satisfies the following:

$$\|F(X_k) + DF(X_k)[\Delta X_k^{(j)}]\| \leq \eta_k^{(j)}\|F(X_k)\|,$$
$$\|F(R_{X_k}(\Delta X_k^{(j)}))\| \leq (1 - t(1 - \eta_k^{(j)}))\|F(X_k)\|. \tag{3.6}$$

Proof Let $\hat{\eta}_k$ be defined in Eq. (2.9) with $\Delta X_k = \widehat{\Delta X_k}$, and $\epsilon_k = \frac{(1-t)(1-\hat{\eta}_k)\|F(X_k)\|}{\|\widehat{\Delta X_k}\|}$. Since F is continuously differentiable, for $\epsilon_k > 0$, there exists a sufficiently small $\delta > 0$ such that $\|\Delta X\| < \delta$ implies that:

$$\|F(R_{X_k}(\Delta X)) - F(R_{X_k}(\mathbf{0}_{X_k})) - DF(R_{X_k}(\mathbf{0}_{X_k}))[\Delta X]\| \leq \epsilon_k\|\Delta X\|,$$

where $\mathbf{0}_{X_k}$ is the origin of $T_{X_k}(\mathscr{O}(n) \times \mathscr{O}(n) \times \mathscr{W}(n))$.

For $\delta > 0$, we let

$$\eta_{\min} = \max\left\{\hat{\eta}_k, 1 - \frac{(1 - \hat{\eta}_k)\delta}{\|\widehat{\Delta X_k}\|}\right\}.$$

Note that once j is sufficiently large,

$$\eta_k^{(j)} - \eta_{\min} \geq \left(\frac{\delta}{\|\widehat{\Delta X_k}\|} - \Theta_{j-1}\right)(1 - \hat{\eta}_k) \geq 0. \tag{3.7}$$

For sufficiently large j, we consider the sequence $\{\Delta X_k^{(j)}\}_j$ in Eq. (3.4) with $\eta_k^{(j)} \in [\eta_{\min}, 1)$. We can see that:

$$\|\Delta X_k^{(j)}\| = \|\frac{1 - \eta_k^{(j)}}{1 - \hat{\eta}_k}\widehat{\Delta X_k}\| \leq \frac{1 - \eta_{\min}}{1 - \hat{\eta}_k}\|\widehat{\Delta X_k}\| \leq \delta.$$

Note that

$$F(X_k) + DF(X_k)[\Delta X_k^{(j)}] = \frac{\eta_k^{(j)} - \hat{\eta}_k}{1 - \hat{\eta}_k}F(X_k) + \frac{1 - \eta_k^{(j)}}{1 - \hat{\eta}_k}\left(DF(X_k)[\widehat{\Delta X_k}] + F(X_k)\right).$$

From triangle inequality, we have

$$\|F(X_k) + DF(X_k)[\Delta X_k^{(j)}]\| \le \frac{\eta_k^{(j)} - \hat{\eta}_k}{1 - \hat{\eta}_k}\|F(X_k)\| + \frac{1 - \eta_k^{(j)}}{1 - \hat{\eta}_k}\hat{\eta}_k\|F(X_k)\|$$
$$= \eta_k^{(j)}\|F(X_k)\|,$$

and

$$\|F(R_{X_k}(\Delta X_k^{(j)}))\| \le \|F(R_{X_k}(\Delta X_k^{(j)})) - F(R_{X_k}(\mathbf{0}_{X_k})) - DF(R_{X_k}(\mathbf{0}_{X_k}))[\Delta X_k^{(j)}]\|$$
$$+ \|F(X_k) + DF(X_k)[\Delta X_k^{(j)}]\|$$
$$\le \epsilon_k\|\Delta X_k^{(j)}\| + \eta_k^{(j)}\|F(X_k)\|$$
$$= \frac{(1-t)(1-\hat{\eta}_k)\|F(X_k)\|}{\|\widehat{\Delta X_k}\|}\left\|\frac{1-\eta_k^{(j)}}{1-\hat{\eta}_k}\widehat{\Delta X_k}\right\| + \eta_k^{(j)}\|F(X_k)\|$$
$$= (1 - t(1 - \eta_k^{(j)}))\|F(X_k)\|.$$

$\square$

From the proof of Theorem 3.3, we can see that for each k, the repeat loop for the backtracking line search will terminate in a finite number of steps once condition (3.7) is satisfied. Moreover, Theorem 3.2 and condition (3.3) imply the following:

$$\lim_{k\to\infty}\|\widehat{\Delta X_k}\| = \mathbf{0}.$$

That is, if k is sufficient large, i.e., $\|\widehat{\Delta X_k}\|$ is small enough, then from the proof of Theorem 3.3 we see that condition (2.11) is always satisfied, i.e., $\eta_k = \hat{\eta}_k$ for all sufficient large k.

To show that Algorithm 1 is a globally convergent algorithm, we have one additional requirement for the retraction R_X, i.e., there exist $\nu > 0$ and $\delta_\nu > 0$ such that:

$$\nu\|\Delta X\| \ge \text{dist}(R_X(\Delta X), X), \tag{3.8}$$

for all $X \in \mathscr{O}(n) \times \mathscr{O}(n) \times \mathscr{W}(n)$ and for all $\Delta X \in T_X(\mathscr{O}(n) \times \mathscr{O}(n) \times \mathscr{W}(n))$ with $\|\Delta X\| \le \delta_\nu$ [1]. Here the notation "dist$(\cdot, \cdot)$" represents the Riemannian distance on $\mathscr{O}(n) \times \mathscr{O}(n) \times \mathscr{W}(n)$. Under this assumption, our next theorem shows the global convergence property of Algorithm 1. We have borrowed the strategy for this proof from that used in [23, Theorem 3.5] to prove the nonlinear matrix equation.

Theorem 3.4 *Assume that Algorithm 1 does not break down. Let X_* be a limit point of $\{X_k\}$. Then X_k converges to X_* and $F(X_*) = \mathbf{0}$. Moreover, X_k converges to X_* quadratically.*

Proof Suppose X_k does not converge to X_*. This implies that there exist two sequences of numbers $\{k_j\}$ and $\{\ell_j\}$ for which:

$$
\begin{aligned}
&X_{k_j} \in N_{\delta/j}(X_*), \\
&X_{k_j+\ell_j} \notin N_\delta(X_*), \\
&X_{k_j+i} \in N_\delta(X_*), \text{ if } i = 1, \ldots, \ell_{j-1} \\
&k_j + \ell_j \le k_{j+1}.
\end{aligned}
$$

From Theorem 3.3, we see that the repeat loop among steps 6–9 of Algorithm 1 terminates in finite steps. For each k, let m_k be the smallest number such that condition (3.6) is satisfied, i.e., $\Delta X_k = \Theta_{m_k} \widehat{\Delta X_k}$ and $\eta_k = 1 - \Theta_{m_k}(1 - \hat{\eta}_k)$ with Θ_{m_k} being defined in Eq. (3.5). It follows from condition (3.1b) that:

$$
\|\Delta X_k\| \le 2\Theta_{m_k} \left(\frac{1 + \eta_{\max}}{1 - \eta_{\max}}\right)(1 - \eta_k)\|DF(X_*)^\dagger\|\|F(X_k)\|, \tag{3.9}
$$

for a sufficiently small δ and $X_k \in B_\delta(X_*)$, so that condition (3.2) is satisfied. Let

$$
\Gamma_{m_k} = 2\Theta_{m_k} \left(\frac{1 + \eta_{\max}}{1 - \eta_{\max}}\right)\|DF(X_*)^\dagger\|.
$$

According to condition (3.8), there exist $\nu > 0$ and $\delta_\nu > 0$ such that:

$$
\nu\|\Delta X\| \ge \text{dist}\left(R_X(\Delta X), X\right),
$$

when $\|\Delta X\| \le \delta_\nu$. Since $F(X_k)$ approaches zero as k approaches infinity, for δ_ν, condition (3.9) implies that there exists a sufficiently large k such that:

$$
\nu\|\Delta X_k\| \ge \text{dist}\left(R_{X_k}(\Delta X_k), X_k\right) \tag{3.10}
$$

is satisfied whenever $\|\Delta X_k\| \le \delta_\nu$.

Then for a sufficiently large j, we can see from conditions (3.9) and (3.10) that:

$$
\begin{aligned}
\frac{\delta}{2} &\le \text{dist}(X_{k_j+\ell_j}, X_{k_j}) \le \sum_{k=k_j}^{k_j+\ell_j-1} \text{dist}(X_{k+1}, X_k) \\
&= \sum_{k=k_j}^{k_j+\ell_j-1} \text{dist}(R_{X_k}(\Delta X_k), X_k) \le \sum_{k=k_j}^{k_j+\ell_j-1} \nu\|\Delta X_k\| \\
&\le \sum_{k=k_j}^{k_j+\ell_j-1} \nu\Gamma_{m_k}(1 - \eta_k)\|F(X_k)\| \le \sum_{k=k_j}^{k_j+\ell_j-1} \frac{\nu\Gamma_{m_k}}{t}\left(\|F(X_k)\| - \|F(X_{k+1})\|\right) \\
&\le \frac{\nu\Gamma_{m_k}}{t}\left(\|F(X_{k_j})\| - \|F(X_{k_{j+1}})\|\right).
\end{aligned}
$$

This is a contraction, since Theorem 3.2 implies that $F(X_{k_j})$ converges to zero as j approaches infinity and Γ_{m_k} is bounded. Thus, X_k converges to X_*, and immediately, we have $F(X_*) = \mathbf{0}$. This completes the proof of the first part.

To show that X_k converges to X_* quadratically once X_k is sufficiently close to X_*, we let C_1 and C_2 be two numbers satisfying the following:

$$\|F(X_{k+1}) - F(X_k) - DF(X_k)[\Delta X_k]\| \leq C_1\|\Delta X_k\|^2,$$
$$\|F(X_k)\| \leq C_2 \text{dist}(X_k, X_*),$$

for a sufficiently large k. The above assumptions are true since F is second differentiable and $F(X_*) = \mathbf{0}$. We can also observe that:

$$
\begin{aligned}
\|F(X_{k+1})\| &\leq \|F(X_{k+1}) - F(X_k) - DF(X_k)[\Delta X_k]\| + \|F(X_k) + DF(X_k)[\Delta X_k]\| \\
&\leq C_1\|\Delta X_k\|^2 + \hat{\eta}_k\|F(X_k)\| \leq C_1(\Gamma_{m_k}\|F(X_k)\|)^2 + \|F(X_k)\|^2 \\
&\leq \left(C_1\Gamma^2 C_2^2 + C_2^2\right)\text{dist}(X_k, X_*)^2,
\end{aligned}
\tag{3.11}
$$

where $\Gamma = 2\left(\dfrac{1 + \eta_{\max}}{1 - \eta_{\max}}\right)\|DF(X_*)^\dagger\|$.

Since X_k converges to X_* as k converges to infinity, for a sufficiently large k, it follows from conditions (3.9), (3.10), (3.6), and (3.11) that:

$$
\begin{aligned}
\text{dist}(X_{k+1}, X_*) = \lim_{p \to \infty} \text{dist}(X_{k+1}, X_p) &\leq \sum_{s=k}^{\infty} \text{dist}\left(X_{s+1}, R_{X_{s+1}}(\Delta X_{s+1})\right) \\
&\leq \sum_{s=k}^{\infty} \nu\|\Delta X_{s+1}\| \leq \sum_{s=k}^{\infty} \nu\Gamma_{m_{s+1}}(1 + \eta_{\max})\|F(X_{s+1})\| \\
&\leq \nu\Gamma(1 + \eta_{\max})\sum_{j=0}^{\infty}(1 - t(1 - \eta_{\max}))^j\|F(X_{k+1})\| \\
&\leq C\text{dist}(X_k, X_*)^2,
\end{aligned}
$$

for some constant $C = \dfrac{\nu\Gamma(1 + \eta_{\max})\left(C_1\Gamma^2 C_2^2 + C_2^2\right)}{t(1 - \eta_{\max})}$.   □

It is true that we might assume without loss of generality that the inverse of $DF(X_k) \circ DF(X_k)^*$ always exists numerically. However, once $DF(X_k) \circ DF(X_k)^*$ is ill-conditioned or (nearly) singular, we choose an operator $E_k = \sigma_k \text{id}_{T_{F(X_k)}}$, where σ_k is a constant and $\text{id}_{T_{F(X_k)}}$ is an identity operator on $T_{F(X_k)}(\mathbb{R}^{n \times n})$ to make $DF(X_k) \circ DF(X_k)^* + \sigma_k\text{id}_{T_{F(X_k)}}$ well-conditioned or nonsingular. In the calculation, this replaces the calculation in Eq. (2.6) with the following equation:

$$(DF(X_k) \circ DF(X_k)^* + \sigma_k\text{id}_{T_{F(X_k)}})[Z_k] = -F(X_k).$$

That is, Algorithm 1 can be modified to fit in this case by replacing the satisfaction of condition (2.7) with the following two conditions:

$$\|(DF(X_k) \circ DF(X_k)^* + \sigma_k \mathrm{id}_{\mathrm{T}_{\mathrm{F}(X_k)}})[Z_k]\| \leq {}_k\|\mathrm{F}(X_k)\|, \tag{3.12a}$$

$$\|(DF(X_k) \circ DF(X_k)^*)[\Delta Z_k] + F(X_k)\| \leq \eta_{\max}\|F(X_k)\|, \tag{3.12b}$$

where $\sigma_k := \min\{\sigma_{\max}, \|F(X_k)\|\}$ is a selected perturbation determined by the parameter $\sigma_{\max}$ and $\|F(X_k)\|$. Of course, we can provide the proof of the quadratic convergence under condition (3.12) without any difficulty (see [50] for a similar discussion). Thus, we ignore the proof here. However, we note that even if a selected perturbation is applied to an ill-conditioned problem, the linear operator $DF(X_k) \circ DF(X_k)^* + \sigma_k \mathrm{id}_{\mathrm{T}_{\mathrm{F}(X_k)}}$ in condition (3.12a) might become nearly singular or ill-conditioned once σ_k is small enough. This will prevent the iteration in the CG method from converging in fewer than n^2 steps, and cause the value of $f'(0)$ to not be negative. This possibility suggests that we apply Algorithm 1 without performing any perturbation in our numerical experiments. If the CG method cannot terminate within n^2 iterations, it may be necessary to compute a new approximated solution ΔZ_k by selecting a new initial value for X_0.

4 Numerical experiments

To test the capacity and efficiency of our method, we generate sets of eigenvalues and singular values from a series of randomly generated matrices. We performed all of the computations in this work in MATLAB version 2016a on a desktop with a 4.2 GHz Intel Core i7 processor and 32 GB of main memory. For our experiments, we set $\eta_{\max} = 0.9$, $\theta_{\min} = 0.1$, $\theta_{\max} = 0.9$, $t = 10^{-4}$, and $\epsilon = 10^{-10}$. Also, in our computation, we emphasize two things. First, once the CG method computed in Algorithm 1 cannot be terminated within n^2 iterations, restart Algorithm 1 with a different initial value X_0. Second, due to the rounding errors in numerical computation, care must be taken in the selection of η_k so that the upper bound $\eta_k\|F(X_k)\|$ in condition (2.7) is not too small to cause the CG method abnormal. To this end, in our experiments, we use the condition

$$\max\{\eta_k\|F(X_k)\|, 10^{-12}\},$$

instead of $\eta_k\|F(X_k)\|$. The implementations of the Algorithm 1 are available online: http://myweb.ncku.edu.tw/~mhlin/Bitcodes.zip.

Example 4.1 To demonstrate the capacity of our approach for solving problems that are relatively large, we randomly generate a set of eigenvalues and a set of singular values of different size, say, $n = 20, 60, 100, 150, 200, 500,$ and 700 from matrices given by the MATLAB command:

$$A = \mathtt{randn(n)}.$$

Table 1 Comparison of the required CGIt♯, INMIt♯, Residual, Error values, and Time for solving the IESP by Algorithm 1

n	CGIt♯	INMIt♯	Residual	Error	Time
20	208	9.4	5.54×10^{-12}	9.65×10^{-13}	2.47×10^{-2}
60	740	10	8.13×10^{-12}	7.23×10^{-13}	4.11×10^{-1}
100	1231	10.4	1.06×10^{-12}	9.74×10^{-14}	2.22
150	1773	10.1	1.01×10^{-12}	1.06×10^{-13}	6.82
200	1939	10.5	1.20×10^{-12}	1.49×10^{-13}	19.3
500	6070	10.6	1.47×10^{-12}	4.12×10^{-13}	665
700	8905	10.6	5.42×10^{-12}	7.24×10^{-13}	2465

For each size, we perform 10 experiments. To illustrate the elasticity of our approach, we randomly generate the initial value $X_0 = (U_0, V_0, W_0)$ in the following way:

$$W_0 = \texttt{triu(randn(n))}, \ W_0(\texttt{find()}) = \mathbf{0}, \ \text{and} \ [U_0, tmp, V_0] = \texttt{svd}(\Lambda + W_0).$$

In Table 1, we show the average residual value (Residual), the average final error (Error), as defined by:

$$\text{final error} = \|\boldsymbol{\lambda}(A_{\text{new}}) - \boldsymbol{\lambda}\|_2 + \|\boldsymbol{\sigma}(A_{\text{new}}) - \boldsymbol{\sigma}\|_2,$$

the average number of iterations within the CG method (CGIt)♯, the average number of iterations within the inexact Newton method (INMIt)♯, and the average elapsed time (Time), as performed by our algorithm. In Table 1, we can see that the elapsed time and the average number of iterations within the CG method increase dramatically as the size of the matrices increases. This can be explained by the fact that the number of degrees of freedom of the problem increases significantly. Thus, the number of the iterations required by the CG method and the required computed time increase correspondingly. However, it is interesting to see that the required number of iterations within the inexact Newton method remains almost the same for matrices of different sizes. One way to speed up the entire process of iterations is to transform the problem (2.6) into a form that is more suitable for the CG method, for example, apply the CG method with a preselected preconditioner. Still, this selection of the preconditioner requires further investigation.

Example 4.2 In this example, we use Algorithm 1 to construct a nonnegative matrix with prescribed eigenvalues and singular values and a specific structure. We specify this IESP and call it the IESP with desired entries (DIESP). The DIESP can be defined as follows.

(**DIESP**) Given a subset $\mathscr{I} = \{(i_t, j_t)\}_{t=1}^{\ell}$ with double subscripts, a set of real numbers $\mathscr{K} = \{k_t\}_{t=1}^{\ell}$, a set of n complex numbers $\{\lambda_i\}_{i=1}^{n}$, satisfying $\{\lambda_i\}_{i=1}^{n} = \{\bar{\lambda}_i\}_{i=1}^{n}$, and a set of n nonnegative numbers $\{\sigma_i\}_{i=1}^{n}$, find a nonnegative $n \times n$

matrix A that has eigenvalues $\lambda_1, \ldots, \lambda_n$, singular values $\sigma_1, \ldots, \sigma_n$ and $A_{i_t, j_t} = k_t$ for $t = 1, \ldots, \ell$.

Note that once $i_t = j_t = t$ for $t = 1, \ldots, n$, we investigate a numerical approach for solving the IESP with prescribed diagonal entries. As far as we know, the research result close to this problem is only available in [46]. However, for a general structure, no research has been conducted to implement this investigation. To solve the DIESP, our first step is to obtain a real matrix A with prescribed eigenvalues and singular values. Our second step is to derive entries of $Q^\top A Q$, where $Q \in \mathcal{O}(n)$, that satisfy the nonnegative property and desired values determined by the sets $\mathcal{I}$ and $\mathcal{K}$. We solve the first step in the same manner as in Example 4.1, but for the second step, we consider the following two sets $\mathcal{L}_1$ and $\mathcal{L}_2$, which are defined by:

$$\mathcal{L}_1 = \{A \in \mathbb{R}^{m \times n} \mid A_{i_t, j_t} = k_t, \text{ for } 1 \le t \le \ell; \text{ otherwise } A_{i,j} = 0\},$$
$$\mathcal{L}_2 = \{A \in \mathbb{R}^{m \times n} \mid A_{i,j} = 0, \text{ for } 1 \le i, j \le n \text{ and } (i,j) \in \mathcal{I}\},$$

and then solve the following problem:

find $P \in \mathcal{L}_2$ and $Q \in \mathcal{O}(n)$ such that $H(P, Q) = \hat{A} + P \odot P - QAQ^\top = \mathbf{0}$,

(4.1)

with $\hat{A} \in \mathcal{L}_1$. Let $[A, B] := AB - BA$ denote the Lie bracket notation. It follows from direct computation that the corresponding differential DH and its adjoint DH^* have the following form [50]:

$$DH(P, Q)[(\Delta P, \Delta Q)] = 2P \odot \Delta P + [QAQ^\top, \Delta Q Q^\top],$$
$$DH(P, Q)^*[\Delta Z] = \left(2P \odot \Delta Z, \frac{1}{2}([QAQ^\top, \Delta Z^\top] + [QA^\top Q^\top, \Delta Z])Q \right),$$

and, for all $(\xi_P, \xi_Q) \in T_{(P,Q)}(\mathcal{L}_2 \times \mathcal{O}(n))$, we can compute the retraction R using the following formula:

$$R(P, Q) = (R_P(\xi_P), R_Q(\xi_Q)),$$

where

$$R_P(\xi_P) = P + \xi_P, \quad R_Q(\xi_Q) = Q_f(Q + \xi_Q).$$

For these experiments, we randomly generate nonnegative matrices 20×20 in size by the MATLAB command "$A = rand(20)$" to provide the desired eigenvalues, singular values, and diagonal entries, i.e., to solve the DIESP with the specified diagonal entries. We record the final error, as given by the following formula:

$$\text{final error} = \|\lambda(A_{\text{new}}) - \lambda\|_2 + \|\sigma(A_{\text{new}}) - \sigma\|_2 + \|(A_{\text{new}})_{i_t, j_t} - k_t\|_2.$$

Table 2 Records of final errors and residual values for solving the DIESP by Algorithm 1

	Interval	Average
Final errors	$[7.27 \times 10^{-13}, 1.21 \times 10^{-11}]$	2.91×10^{-12}
Residual values	$[7.77 \times 10^{-13}, 4.93 \times 10^{-12}]$	1.85×10^{-12}

After randomly choosing 10 different matrices, Table 2 shows our results with the intervals (Interval) containing all of the residual values and final errors, and their corresponding average values (Average). These results provide sufficient evidence that Algorithm 1 can be applied to solve the DIESP with high accuracy.

Although Example 4.2 considers examples with a nonnegative structure, we emphasize that Algorithm 1 can work with entries that are not limited to being nonnegative. That is, to solve the IESP without nonnegative constraints but with another specific structure, Algorithm 1 can fit perfectly well by replacing $H(P, Q)$ in problem (4.1) with

$$G(S, Q) := \hat{A} + S - QAQ^\top,$$

where $\hat{A} \in \mathscr{L}_1$, $S \in \mathscr{L}_2$ and $Q \in \mathcal{O}(n)$.

5 Conclusions

In this paper, we apply the Riemannian inexact Newton method to solve an initially complicated and challenging IESP. We provide a thorough analysis of the entire iterative processes and show that this algorithm converges globally and quadratically to the desired solution. We must emphasize that our theoretical discussion and numerical implementations can also be extended to solve an IESP with a particular structure such as desired diagonal entries and a matrix whose entries are nonnegative. This capacity can be observed in our numerical experiments. It should be emphasized that this research is the first to provide a unified and effective means to solve the IESP with or without a particular structure.

However, the numerical stability for extremely ill-conditioned problems is a case that we should pay attention to, though reselecting the initial values could be a strategy to get rid of this difficulty. Another way to tackle this difficulty is to select a good preconditioner. But, the operator encountered in our algorithm is nonlinear and high-dimensional. Thus, the selection of the preconditioner could involve the study of tensor analysis, where further research is needed.

References

1. Absil, P.-A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2008)
2. Baker, C.G., Absil, P.-A., Gallivan, K.A.: An implicit Riemannian trust-region method for the symmetric generalized eigenproblem. In: Computational Science—ICCS 2006, LNCS. Springer, Berlin (2006)
3. Boley, D.L., Golub, G.H.: A survey of matrix inverse eigenvalue problems. Inverse Problems **3**(4), 595–622 (1987)
4. Boyd, S., Xiao, L.: Least-squares covariance matrix adjustment. SIAM J. Matrix Anal. Appl. **27**(2), 532–546 (2005)
5. Chan, N.N., Li, K.H.: Diagonal elements and eigenvalues of a real symmetric matrix. J. Math. Anal. Appl. **91**(2), 562–566 (1983)
6. Chu, E.K., Datta, B.N.: Numerically robust pole assignment for second-order systems. Int. J. Control **64**(6), 1113–1127 (1996)
7. Chu, M.T.: Numerical methods for inverse singular value problems. SIAM J. Numer. Anal. **29**(3), 885–903 (1992)
8. Chu, M.T.: On constructing matrices with prescribed singular values and diagonal elements. Linear Algebra Appl. **288**(1–3), 11–22 (1999)
9. Chu, M.T.: A fast recursive algorithm for constructing matrices with prescribed eigenvalues and singular values. SIAM J. Numer. Anal. **37**(3), 1004–1020 (2000)
10. Chu, M.T.: Linear algebra algorithms as dynamical systems. Acta Numer. **17**, 1–86 (2008)
11. Chu, M.T.: On the first degree Fejér–Riesz factorization and its applications to $X + A^* X^{-1} A = Q$. Linear Algebra Appl. **489**, 123–143 (2016)
12. Chu, M.T., Driessel, K.R.: The projected gradient method for least squares matrix approximations with spectral constraints. SIAM J. Numer. Anal. **27**(4), 1050–1060 (1990)
13. Chu, M.T., Driessel, K.R.: Constructing symmetric nonnegative matrices with prescribed eigenvalues by differential equations. SIAM J. Math. Anal. **22**(5), 1372–1387 (1991)
14. Chu, M.T., Golub, G.H.: Inverse Eigenvalue Problems: Theory, Algorithms, and Applications. Numerical Mathematics and Scientific Computation. Oxford University Press, New York (2005)
15. Chu, M.T., Lin, W.-W., Xu, S.-F.: Updating quadratic models with no spillover effect on unmeasured spectral data. Inverse Problems **23**(1), 243–256 (2007)
16. Chu, M.T., Wright, J.W.: The educational testing problem revisited. IMA J. Numer. Anal. **15**(1), 141–160 (1995)
17. Cotae, P., Aguirre, M.: On the construction of the unit tight frames in code division multiple access systems under total squared correlation criterion. AEU Int. J. Electron. Commun. **60**(10), 724–734 (2006)
18. Datta, B.N.: Finite-element model updating, eigenstructure assignment and eigenvalue embedding techniques for vibrating systems. Mech. Sys. Signal Process. **16**(1), 83–96 (2002)
19. Datta, B.N., Elhay, S., Ram, Y.M., Sarkissian, D.R.: Partial eigenstructure assignment for the quadratic pencil. J. Sound Vib. **230**(1), 101–110 (2000)
20. Datta, B.N., Lin, W.-W., Wang, J.-N.: Robust partial pole assignment for vibrating systems with aerodynamic effects. IEEE Trans. Autom. Control **51**(12), 1979–1984 (2006)
21. Deakin, A.S., Luke, T.M.: On the inverse eigenvalue problem for matrices (atomic corrections). J. Phys. A Math. Gen. **25**(3), 635 (1992)
22. Dhillon, I.S., Heath Jr., R.W., Sustik, M.A., Tropp, J.A.: Generalized finite algorithms for constructing Hermitian matrices with prescribed diagonal and spectrum. SIAM J. Matrix Anal. Appl. **27**(1), 61–71 (2005)
23. Eisenstat, S.C., Walker, H.F.: Globally convergent inexact Newton methods. SIAM J. Optim. **4**(2), 393–422 (1994)
24. Gladwell, G.M.L.: Inverse Problems in Vibration, Volume 119 of Solid Mechanics and Its Applications, 2nd edn. Kluwer Academic Publishers, Dordrecht (2004)
25. Gohberg, I., Lancaster, P., Rodman, L.: Matrix Polynomials. Society for Industrial and Applied Mathematics, Philodelphia (2009)
26. Golub, G.H.: Some modified matrix eigenvalue problems. SIAM Rev. **15**(2), 318–334 (1973)
27. Grubišić, I., Pietersz, R.: Efficient rank reduction of correlation matrices. Linear Algebra Appl. **422**(2–3), 629–653 (2007)

28. Higham, N.J.: Accuracy and Stability of Numerical Algorithms. SIAM, Philadelphia (1996)
29. Horn, A.: On the eigenvalues of a matrix with prescribed singular values. Proc. Am. Math. Soc. **5**, 4–7 (1954)
30. Jacek, K.: Inverse problems in quantum chemistry. Int. J. Quantum Chem. **109**(11), 2456–2463 (2009)
31. Kelley, C.T.: Iterative Methods for Linear and Nonlinear Equations, Volume 16 of Frontiers in Applied Mathematics. SIAM, Philadelphia (1995)
32. Kosowski, P., Smoktunowicz, A.: On constructing unit triangular matrices with prescribed singular values. Computing **64**(3), 279–285 (2000)
33. Li, C.-K., Mathias, R.: Construction of matrices with prescribed singular values and eigenvalues. BIT Numer. Math. **41**(1), 115–126 (2001)
34. Li, N.: A matrix inverse eigenvalue problem and its application. Linear Algebra Appl. **266**, 143–152 (1997)
35. Luenberger, D.G.: Optimization by Vector Space Methods. Wiley, New York (1969)
36. Mirsky, L.: Matrices with prescribed characteristic roots and diagonal elements. J. Lond. Math. Soc. **33**, 14–21 (1958)
37. Möller, M., Pivovarchik, V.: Inverse Sturm–Liouville Problems. Springer, Cham (2015)
38. Nichols, N.K., Kautsky, J.: Robust eigenstructure assignment in quadratic matrix polynomials: non-singular case. SIAM J. Matrix Anal. Appl. **23**(1), 77–102 (2001)
39. Rao, R., Dianat, S.: Basics of Code Division Multiple Access (CDMA). SPIE Tutorial Texts. SPIE Press, Washington (2005)
40. Simonis, J.P.: Inexact Newton methods applied to under-determined systems. Ph.D. dissertation, Worcester Polytechnic Institute, Worcester, MA (2006)
41. Sing, F.Y.: Some results on matrices with prescribed diagonal elements and singular values. Can. Math. Bull. **19**(1), 89–92 (1976)
42. Thompson, R.C.: Singular values, diagonal elements, and convexity. SIAM J. Appl. Math. **32**(1), 39–63 (1977)
43. Tropp, J.A., Dhillon, I.S., Heath Jr., R.W.: Finite-step algorithms for constructing optimal CDMA signature sequences. IEEE Trans. Inf. Theory **50**(11), 2916–2921 (2004)
44. Wang, L., Chu, M.T., Bo, Y.: A computational framework of gradient flows for general linear matrix equations. Numer. Algorithms **68**(1), 121–141 (2015)
45. Weyl, H.: Inequalities between the two kinds of eigenvalues of a linear transformation. Proc. Natl. Acad. Sci. USA **35**, 408–411 (1949)
46. Wu, S.-J., Chu, M.T.: Solving an inverse eigenvalue problem with triple constraints on eigenvalues, singular values, and diagonal elements. Inverse Problems **33**(8), 085003, 21 (2017)
47. 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**(1), 215–234 (2016)
48. Zha, H., Zhang, Z.: A note on constructing a symmetric matrix with specified diagonal entries and eigenvalues. BIT **35**(3), 448–451 (1995)
49. Zhao, Z., Bai, Z.-J., Jin, X.-Q.: A Riemannian Newton algorithm for nonlinear eigenvalue problems. SIAM J. Matrix Anal. Appl. **36**(2), 752–774 (2015)
50. 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**(4), 827–855 (2018)
51. Zhao, Z., Jin, X.-Q., Bai, Z.-J.: A geometric nonlinear conjugate gradient method for stochastic inverse eigenvalue problems. SIAM J. Numer. Anal. **54**(4), 2015–2035 (2016)