Journal of Scientific Computing (2020) 82:67
https://doi.org/10.1007/s10915-020-01173-5

A Riemannian Optimization Approach for Solving the Generalized Eigenvalue Problem for Nonsquare Matrix Pencils

Jiao-fen Li[1] · Wen Li[2] · Seak-Weng Vong[3] · Qi-Lun Luo[4] · MingQing Xiao[4]

Received: 28 January 2019 / Revised: 4 January 2020 / Accepted: 20 February 2020 /
Published online: 27 February 2020

Abstract
In this paper, based on the Riemannian optimization approach we propose a Riemannian nonlinear conjugate gradient method with nonmonotone line search technique for solving the l parameterized original problem on generalized eigenvalue problems for nonsquare matrix pencils, which was first proposed by Chu and Golub (SIAM J Matrix Anal Appl 28:770–787, 2006). The new innovative approach is to reformulate the original optimization problem as a feasible optimization problem over a certain real product manifold. The global convergence of the proposed method is then established. Some numerical tests are given to demonstrate the efficiency of the proposed method. Comparisons with some latest methods are also given.

Keywords Generalized eigenvalue · Nonsquare pencils · Riemannian optimization · Stiefel manifold

Mathematics Subject Classification 15A24 · 15A18 · 65F10 · 65F15

1 Introduction

Given two matrices $A, B \in \mathbb{C}^{m\times n}$ and $m > n$, the set $\{A-\lambda B; \lambda \in \mathbb{C}\}$ constitutes a nonsquare matrix pencil. The generalized eigenvalues of the nonsquare matrix pencil are those values $\lambda \in \mathbb{C}$ so that for each λ there exists a nonzero vector $\mathbf{v} \in \mathbb{C}^n$, called a generalized eigenvector, such that the pair $(\lambda, \mathbf{v})$ satisfies

$$(A - \lambda B)\mathbf{v} = 0.$$

Research was supported in part by National Natural Science Foundation of China (11761024, 11671158, U1811464, 11561015, 11771159), MYRG2017-00098-FST from University of Macau, Natural Science Foundation of Guangxi Province (2016GXNSFAA380074, 2016GXNSFFA380009, 2017GXNSFBA198082), and in part by NSF-DMS 1419028, 1854638 of the United States.

✉ Wen Li
liwen@scnu.edu.cn

Extended author information available on the last page of the article

During recent years such problems (e.g., see [1–5] and references therein) have received considerable attention across various areas due to many potential applications.

The main challenge of the generalized eigenvalue problem for nonsquare pencils in practice is following: even if n linearly independent eigenpairs $(\lambda_k, \mathbf{v}_k)$, $k = 1, \ldots, n$, are known to exist in the noiseless case, these eigenpairs may fail to exist under some perturbations. To overcome this difficulty, Boutry et al. [1] considered an optimization problem by looking for the minimum perturbation of the given pair of matrices (A, B) such that the perturbed pair $(\hat{A}, \hat{B})$ still remains to be n linearly independent eigenvectors:

$$\left\{\begin{array}{ll} \inf & \|A-\hat{A}\|^2 + \|B-\hat{B}\|^2 \\ \text{subject to} & \hat{A}, \hat{B} \in \mathbb{C}^{m\times n},\ \{(\lambda_k, \mathbf{v}_k)\}_{k=1}^{n} \subset \mathbb{C}\times\mathbb{C}^n, \\ & \hat{A}\mathbf{v}_k = \lambda_k \hat{B}\mathbf{v}_k,\ k = 1, 2, \ldots, n, \\ & \lambda_k \neq \lambda_j,\ \forall 1 \le k \neq j \le n, \\ & \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\} : \text{linearly independent}, \end{array}\right. \tag{1.1}$$

where $\|M\|$ denotes the Frobenius norm of a complex matrix M which will be defined in the end of this section. Two special cases of (1.1) that allow for a simpler formulation of this optimization problem were studied in [1]. The first case is $n = 1$, i.e., $A, B \in \mathbb{C}^m$, that was discussed. For this case Boutry et al. have showed that (1.1) is equivalent to a total least squares problem, and have derived a closed form solution. The second case with $n > 1$ was further studied, under the assumption that a single finite eigenvalue is known to be existed. The simplified formulation of the minimal perturbation approach for this special case can then be characterized as

$$\left\{\begin{array}{ll} \inf & \|A-\hat{A}\|^2 + \|B-\hat{B}\|^2 \\ \text{subject to} & \hat{A}, \hat{B} \in \mathbb{C}^{m\times n},\ (\lambda, \mathbf{v}) \subset \mathbb{C}\times\mathbb{C}^n, \\ & \hat{A}\mathbf{v} = \lambda\hat{B}\mathbf{v},\ \mathbf{v} \neq \mathbf{0}. \end{array}\right. \tag{1.2}$$

For this case an efficient numerical algorithm is proposed to seek the possible solutions. However, the authors in [1] pointed out that the general problem (1.1) is complicated and is still open in literature.

Subsequently, Chu and Golub [2] considered the more general case of the problem (1.1) with a positive integer l, $1 \le l \le n$, i.e.,

$$\left\{\begin{array}{ll} \inf & \|A-\hat{A}\|^2 + \|B-\hat{B}\|^2 \\ \text{subject to} & \hat{A}, \hat{B} \in \mathbb{C}^{m\times n},\ \{(\lambda_k, \mathbf{v}_k)\}_{k=1}^{l} \subset \mathbb{C}\times\mathbb{C}^n, \\ & \hat{A}\mathbf{v}_k = \lambda_k \hat{B}\mathbf{v}_k,\ k = 1, 2, \ldots, l, \\ & \lambda_k \neq \lambda_j,\ \forall 1 \le k \neq j \le l, \\ & \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_l\} : \text{linearly independent}. \end{array}\right. \tag{1.3}$$

It is noted that the problem (1.3) is formulated more general than (1.1) and (1.2), since the case of $l = n$ is (1.1) and that of $l = 1$ in (1.2). In [2], Chu and Golub have given the algebraic characterization for the infimum of the cost function in the problem (1.3), and they have also shown that this infimum can be obtained by solving the following optimization problem over a compact set of matrices,

$$\text{minimize} \left\{ \sqrt{\sum_{i=l+1}^{\min\{m,2l\}} \sigma_i^2([AV\ \ BV]) :\quad V \in \mathbb{C}^{n\times l},\quad V^H V = I_l} \right\}, \tag{1.4}$$

where $\sigma_i(M)$ is the ith singular value of $M \in \mathbb{C}^{m\times n}$ with decreasing order, namely, $\sigma_1(M) \ge \sigma_2(M) \ge \cdots \ge \sigma_{\min\{m,n\}}(M)$. Moreover, as pointed out in [2], the infimum of the objection function in (1.1) has a very concise algebraic characterization by (1.4) with $l = n$, and some equivalent characterizations for the problem (1.2) given in [1] can also be rederived by (1.4) with $l = 1$. But no algorithm for solving the corresponding optimization problem (1.4) was given in [2]. Very recently, Ito and Murota [5] proposed a robust algorithm for the problem (1.1), which is based on the total least squares problem introduced by Golub and Van Loan [6]. They have shown by numerical examples that their algorithm is more robust against data noise than the algorithm given by Boutry et al in [1]. Further, they also point out in the conclusion part of [5]: “A challenging future work is to extend our results to the problem (1.3) containing a parameter l, which seems to be much more complicated than (1.1)”. To the best of our knowledge, the numerical algorithms for solving the problem (1.3) are still remaining open at this point.

Motivated by the aforementioned studies, in this paper, we are going to establish numerical approaches for solving the problem (1.3), by directly focusing on the optimization problem (1.4) from the numerical point of view. The main contributions of this paper are summarized in the following:

1. We reformulate the problem (1.4) into a Riemannian optimization problem on the product of two complex Stiefel manifolds under the optimization framework. For the purpose of approach feasibility, we then consider the reformulated problem as an equivalent one on the product of two real manifolds, each of which is an intersection of the real Stiefel manifold and the “quasi-symplectic set”, which will be defined in Sect. 2. Our formulation leads to the numerical approach for solving (1.3) being not only feasible and approachable but also efficient in numerical implementation.
2. We develop a Riemannian nonlinear conjugate gradient method for the optimization problem on the real product manifold, for which we generalize Dai’s nonmonotone conjugate gradient method [7] from the Euclidean space to Riemannian product manifolds. To make the algorithm be more transparent, we first translate the algorithm into a complex-value form, and then develope a complete algorithm for solving the original problem (1.3) efficiently.

Numerical results show that the proposed algorithm is quite efficient for solving the problem (1.3) with any $1 \leq l \leq n$. Detailed comparisons with Boutry et al.’s algorithm [1] for the problem (1.2) in the case $l = 1$, Ito and Murota’s algorithm [5] for the problem (1.1) in the case $l = n$ are also given. For the case $l = n$, the algorithm is less affected by the data noise and guaranteed to yield n eigenpairs, and the algorithm generated exactly the same optimal solution as Ito and Murota’s algorithm [5], which is essentially a direct method to solve the problem (1.1), and was proposed by Boutry et al. in [1]. Moreover, the algorithm runs faster than Boutry et al.’s algorithm [1] for the case $l = 1$, where the perturbed pair with minimal perturbation admits at least one engenpair.

The rest of this paper is organized as follows. In Sect. 2 we reformulate (1.4) as a Riemannian optimization problem on the product of two complex Stiefel manifolds, and then reformulate into a real optimization problem over a certain real product manifold for the purpose of feasibility. Sect. 3 studies the geometric properties of the resultant real product manifold. A Riemannian nonlinear conjugate gradient numerical method is developed in Sect. 4, and its global convergence is also established in this section. Finally, some numerical tests are reported in Sect. 5, and some concluding remarks are given in Sect. 6.

Notation Let M^T, M^H and $\mathrm{tr}(M)$ represent the transpose, conjugate transpose and trace of a matrix M, respectively. I_n is the identity matrix of order n. $\mathrm{Re}(M)$ and $\mathrm{Im}(M)$ denote the real and imaginary parts of M. $\mathrm{sym}(M) := (M + M^T)/2$ denotes the symmetric part of a real matrix M. Let $\mathbb{N}^+$ be the set of positive integers. Let $\mathbb{R}^{p\times q}$ and $\mathbb{C}^{p\times q}$ be the set of all

$p \times q$ real matrices and the set of all $p \times q$ complex matrices, respectively. Let $\mathbb{S}^{l\times l}$ and $\mathbb{AS}^{l\times l}$ be the sets of all $l \times l$ real symmetric and skew-symmetric matrices, respectively. For two complex matrices $M, N \in \mathbb{C}^{p\times q}$, their inner product is defined by

$$\mathrm{Tr}(M^H N) = \mathrm{Re}(\mathrm{tr}(M^H N)) = \mathrm{tr}\left(\mathrm{Re}(M)^T \mathrm{Re}(N)\right) + \mathrm{tr}\left(\mathrm{Im}(M)^T \mathrm{Im}(N)\right). \quad (1.5)$$

Then $\mathbb{C}^{p\times q}$ is a Hilbert inner product space over $\mathbb{R}$ and the matrix norm induced by this inner product is

$$\|M\| = \sqrt{\mathrm{Tr}(M^H M)} = (\mathrm{Re}(\mathrm{tr}(M^H M)))^{\frac{1}{2}} = (\mathrm{tr}(M^H M))^{\frac{1}{2}}. \quad (1.6)$$

2 The Transformed Problem

In this section, we reformulate the problem (1.4) into a Riemannian optimization problem on the product of two complex Stiefel manifolds. For the purpose of feasibility, we further rewrite the reformulated problem as an equivalent problem on the product of two real manifolds, with each of them being an intersection of the real Stiefel manifold and the "quasi-symplectic set" to be defined later on.

First of all, we start with the following elementary lemma.

Lemma 2.1 (Theorem 3.3 of [8]) *Let $M \in \mathbb{C}^{m\times m}$ be a Hermitian matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_m$. Let $U_l \in \mathbb{C}^{m\times l}$, $U_l^H U_l = I_l$. Then $\lambda_{m-l+1} + \cdots + \lambda_m = \min_{U_l} tr(U_l^H M U_l)$.*

Following from Lemma 2.1, the problem (1.4) can be rewritten as follows:

$$\begin{cases} \text{minimize} & \|[AV\ BV]P\|^2 \\ \text{subject to} & V \in \mathbb{C}^{n\times l},\ V^H V = I_l, \\ & P \in \mathbb{C}^{2l\times l},\ P^H P = I_l. \end{cases} \quad (2.1)$$

Note that a real Stiefel manifold is defined to be $\mathrm{St}(n, l, \mathbb{R}) = \{Y \in \mathbb{R}^{n\times l} | Y^T Y = I_l\}$. Because of the constraints, the set of all feasible points of the problem (2.1) naturally becomes the product manifold $\mathrm{St}(n, l, \mathbb{C}) \times \mathrm{St}(2l, l, \mathbb{C})$, where $\mathrm{St}(n, l, \mathbb{C})$ is the complex Stiefel manifold defined by $\mathrm{St}(n, l, \mathbb{C}) = \{Y \in \mathbb{C}^{n\times l} | Y^H Y = I_l\}$. Then (2.1) can be equivalent to the following Riemannian optimization problem

$$\begin{cases} \text{minimize} & f(V, P) := \|[AV\ BV]P\|^2 \\ \text{subject to} & (V, P) \in \mathrm{St}(n, l, \mathbb{C}) \times \mathrm{St}(2l, l, \mathbb{C}). \end{cases} \quad (2.2)$$

When dealing with the complex-value optimization problem, we generally convert them to a real-value form. Motivated by Sato et al. [9,10] for tackling the complex singular decomposition problem, we next give a real form of the problem (2.2). To identify a complex matrix with a real matrix, we consider the following correspondence. A complex matrix $M = M_1 + iM_2 \in \mathbb{C}^{p\times q}$, where $M_1, M_2 \in \mathbb{R}^{p\times q}$, has a one-to-one correspondence with a real matrix

$$\widetilde{M} = \begin{bmatrix} M_1 & M_2 \\ -M_2 & M_1 \end{bmatrix} \in \mathbb{R}^{2p\times 2q}. \quad (2.3)$$

Let $\phi^{(p,q)}$ denote the isomorphism of this correspondence: $\phi^{(p,q)}(M) = \widetilde{M}$. Define the following quasi-symplectic set $\mathcal{SP}(p, q)$ for integers p, q by

$$\mathcal{SP}(p, q) = \left\{ \widetilde{M} \in \mathbb{R}^{2p\times 2q} \middle| \widetilde{M} J_q = J_p \widetilde{M} \quad \text{with } J_p := \begin{bmatrix} 0 & I_p \\ -I_p & 0 \end{bmatrix} \right\}.$$

It is easy to verify that a $2p \times 2q$ real matrix $\widetilde{M}$ has the block form of (2.3) if and only if $\widetilde{M} \in \mathcal{SP}(p,q)$. On the other hand, the set $\mathcal{SP}(p,q)$ is a linear subspace of $\mathbb{R}^{2p\times 2q}$. Therefore, the complex vector space $\mathbb{C}^{p\times q}$ is isomorphic to the linear subspace $\mathcal{SP}(p,q)$, with the isomorphism $\phi^{(p,q)}$. Furthermore, for a complex $M \in \mathbb{C}^{p\times q}$, one can easily prove that $M \in \mathrm{St}(p,q,\mathbb{C}) \Leftrightarrow \phi^{(p,q)}(M) \in \mathrm{St}(2p,2q,\mathbb{R})$.

Define the map $\phi^{(p,q)}|_{\mathrm{St}(p,q,\mathbb{C})}$ as the restriction of $\phi^{(p,q)}$ to the complex Stiefel manifold $\mathrm{St}(p,q,\mathbb{C})$. Then, this map yields a real expression for $\mathrm{St}(p,q,\mathbb{C})$, which is denoted throughout this paper by

$$\mathrm{Stp}(p,q) := \mathrm{St}(2p,2q,\mathbb{R}) \cap \mathcal{SP}(p,q). \tag{2.4}$$

For simplicity, in what follows, we define the symbol ϕ as the collection of $\phi^{(p,q)}$ for any integers p, q as:

$$\phi(M) = \phi^{(p,q)}(M), \quad M \in \mathbb{C}^{p\times q}.$$

That is, for a complex matrix M of any size, we have

$$\phi(M) = \begin{bmatrix} \mathrm{Re}(M) & \mathrm{Im}(M) \\ -\mathrm{Im}(M) & \mathrm{Re}(M) \end{bmatrix}. \tag{2.5}$$

We also use the notation $\widetilde{M}$ for the matrix $\phi(M)$, i.e., $\widetilde{M} = \phi(M)$, for simplicity. Then, matrix operations on matrices without and with tilde are related as follows:

$$\phi(M+N) = \widetilde{M} + \widetilde{N}, \quad \phi(ME) = \widetilde{M}\widetilde{E}, \quad \phi(M^H) = \widetilde{M}^T, \tag{2.6}$$

where M, N, E are complex matrices of appropriate sizes for the addition and multiplication operations. If $\widetilde{M} \in \mathcal{SP}(n,n)$ is nonsingular, then $M = \phi^{-1}(\widetilde{M})$ is also nonsingular since the isomorphism of ϕ together with $\phi(M^{-1})\phi(M) = \phi(M^{-1}M) = \phi(I_n) = I_{2n}$ implies that $\widetilde{M}^{-1} = \phi(M^{-1}) \in \mathcal{SP}(n,n)$. Additionally, traces of square complex matrices M and $\widetilde{M}$ are related by

$$2\mathrm{Re}(\mathrm{tr}(M)) = \mathrm{tr}(\widetilde{M}), \tag{2.7}$$

and the Frobenius norm of M and $\widetilde{M}$ are connected by

$$2\|M\|^2 = 2\mathrm{Re}(\mathrm{tr}(M^H M)) = \|\widetilde{M}\|^2. \tag{2.8}$$

Note that for any $\widetilde{M} \in \mathcal{SP}(p,q)$ and $\widetilde{E} \in \mathcal{SP}(q,s)$, one can easily verify that $\widetilde{M}\widetilde{E} \in \mathcal{SP}(p,s)$, that is to say, the set $\mathcal{SP}(\cdot,\cdot)$ is closed under the operations in the right-hand sides of (2.6).

With above preliminaries, we can rewrite the problem (2.2) on the product of two complex Stiefel manifolds into its real form by using $\mathrm{Stp}(n,l)$ given in (2.4). Define

$$J = \begin{bmatrix} I_l & 0 & 0 & 0 \\ 0 & 0 & I_l & 0 \\ 0 & I_l & 0 & 0 \\ 0 & 0 & 0 & I_l \end{bmatrix} \in \mathbb{R}^{4l\times 4l},$$

and partition $P \in \mathbb{C}^{2l\times l}$ in (2.2) as $P = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix}$ with $P_1, P_2 \in \mathbb{C}^{l\times l}$, then by using the following relation we have

$$J\widetilde{P} = J\widetilde{\begin{bmatrix} P_1 \\ P_2 \end{bmatrix}} = \begin{bmatrix} \widetilde{P}_1 \\ \widetilde{P}_2 \end{bmatrix}. \tag{2.9}$$

Due to (2.7) and (2.9), the objective function $f(V,P)=\|[AV\ BV]P\|^2$ in (2.2) now can be rewritten as

$$\|[AV\ BV]P\|^2=\frac{1}{2}\|[\widetilde{A}\widetilde{V}\ \widetilde{B}\widetilde{V}]J\widetilde{P}\|^2. \tag{2.10}$$

Throughout this paper we will use the same symbol f to denote the function of $(\widetilde{V},\widetilde{P})\in\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l)$ on the right-hand side of (2.10), which leads to the following optimization problem on $\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l)$:

$$\begin{cases}\text{minimize} & f(\widetilde{V},\widetilde{P}):=\frac{1}{2}\|[\widetilde{A}\widetilde{V}\ \widetilde{B}\widetilde{V}]J\widetilde{P}\|^2\\ \text{subject to} & (\widetilde{V},\widetilde{P})\in\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l).\end{cases} \tag{2.11}$$

For convenience, we define the mapping $F:\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l)\to\mathcal{SP}(m,l)$ by

$$F(\widetilde{V},\widetilde{P})=[\widetilde{A}\widetilde{V}\ \widetilde{B}\widetilde{V}]J\widetilde{P},\quad (\widetilde{V},\widetilde{P})\in\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l). \tag{2.12}$$

3 Riemannian Geometry of Stp(n, l) × Stp($2l, l$)

Riemannian optimization refers to minimizing a function $f(x)$ on a Riemannian manifold $\mathcal{M}$. A general iteration of Riemannian optimization has the scheme:

$$x_{k+1}=\mathcal{R}_{x_k}(\alpha_k\eta_k),$$

where α_k is the step size at the kth iterate x_k, $\eta_k\in\mathbf{T}_{x_k}\mathcal{M}$ is a tangent vector, and where $\mathbf{T}_x\mathcal{M}$ denotes the tangent space at x on a manifold $\mathcal{M}$, and $\mathcal{R}$ is a retraction [11], which is a smooth map from the tangent bundle of $\mathcal{M}$ into $\mathcal{M}$ that approximates the exponential map to the first order. The construction of a conjugate gradient direction in the Euclidean space is associated with the numerical scheme:

$$\eta_{k+1}=-\nabla f(x_{k+1})+\beta_k\eta_k. \tag{3.1}$$

This formula is meaningless for Riemannian manifolds since vectors in different tangent spaces do not permit the operation of additions. So we have to find a way to transport a vector from one tangent space to another, that is, the so-called vector transport [11]. The notion of vector transport $\mathcal{T}$ on a manifold $\mathcal{M}$, roughly speaking, specifies how to transport a tangent vector ξ from a point $x\in\mathcal{M}$ to a point $\mathcal{R}_x(\eta)\in\mathcal{M}$. Given a vector transport, we can define a Riemannian conjugate gradient direction as follows:

$$\eta_{k+1}=-\nabla f(x_{k+1})+\beta_{k+1}\mathcal{T}_{\alpha_k\eta_k}(\eta_k). \tag{3.2}$$

According to previous discussions, we known that, in order to apply Riemannian conjugate gradient method to the problem (2.11), we need the Riemannian gradient of the objective function f, together with a retraction and a vector transport on the product manifold $\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l)$. For this purpose, we now study some basic geometric properties of $\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l)$, including tangent spaces, Riemannian metrics, orthogonal projections, retractions, as well as vector transports, and derive an explicit expression of the Riemannian gradient of the cost function. This is motivated by [9,10,12–14] for applying Riemannian optimization approaches to solve various kinds of eigenproblems during recent years and the latest development on Riemannian conjugate gradient methods [15,16]. For the Riemannian geometry of the real Stiefel manifold $\mathrm{St}(n,l,\mathbb{R})$, the reader is referred to [11,17] for more details.

3.1 Tangent Spaces and the Orthogonal Projection

Since the tangent space of the complex Stiefel manifold $\mathrm{St}(n,l,\mathbb{C})$ at $V \in \mathrm{St}(n,l,\mathbb{C})$ can be expressed as $\mathbf{T}_V\mathrm{St}(n,l,\mathbb{C}) = \{\xi \in \mathbb{C}^{n\times p} | \xi^H V + V^H \xi = 0\}$ [11], the tangent space of $\mathrm{Stp}(n,l)$ at $\widetilde{V} \in \mathrm{Stp}(n,l)$ is naturally given by

$$\mathbf{T}_{\widetilde{V}}\mathrm{Stp}(n,l) = \left\{\widetilde{\xi} \in \mathcal{SP}(n,l) \mid \widetilde{\xi}^T\widetilde{V} + \widetilde{V}^T\widetilde{\xi} = 0\right\}.$$

And the tangent space $\mathbf{T}_{(\widetilde{V},\widetilde{P})}(\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l))$ at $(\widetilde{V},\widetilde{P}) \in \mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l)$ can be expressed as

$$\begin{aligned}\mathbf{T}_{(\widetilde{V},\widetilde{P})}&\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l) \simeq \mathbf{T}_{\widetilde{V}}\mathrm{Stp}(n,l)\times\mathbf{T}_{\widetilde{P}}\mathrm{Stp}(2l,l)\\ &= \left\{(\widetilde{\xi},\widetilde{\eta}) \in \mathcal{SP}(n,l)\times\mathcal{SP}(2l,l) \,\middle|\, \widetilde{\xi}^T\widetilde{V} + \widetilde{V}^T\widetilde{\xi} = \widetilde{\eta}^T\widetilde{P} + \widetilde{P}^T\widetilde{\eta} = 0\right\}, \qquad (3.3)\end{aligned}$$

where "$\simeq$" means the equivalent identification of two sets.

We now proceed to develop $\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l)$ with a Riemannian metric. The Euclidean space $\mathbb{R}^{2n\times 2l}$ is endowed with the standard inner product

$$\langle E_1, E_2\rangle = \mathrm{tr}(E_1^T E_2), \quad E_1, E_2 \in \mathbb{R}^{2n\times 2l}.$$

When restricted to the linear subspace $\mathcal{SP}(n,l)$ of $\mathbb{R}^{2n\times 2l}$, to get rid of the factor 2 when applying the standard inner product directly, we define the inner product on $\mathcal{SP}(n,l)$ to be

$$\langle\widetilde{M},\widetilde{N}\rangle = \frac{1}{2}\mathrm{tr}(\widetilde{M}^T\widetilde{N}) = \mathrm{tr}(M_x^T N_x + M_y^T N_y)$$

for any $\widetilde{M} = \begin{bmatrix} M_x & M_y \\ -M_y & M_x\end{bmatrix}$, $\widetilde{N} = \begin{bmatrix} N_x & N_y \\ -N_y & N_x\end{bmatrix} \in \mathcal{SP}(n,l)$. Then, the manifold $\mathrm{Stp}(n,l)$ as a submanifold of $\mathcal{SP}(n,l)$ is endowed with the induced metric and its induced norm $\|\widetilde{M}\|^2 = \langle\widetilde{M},\widetilde{M}\rangle = \frac{1}{2}\mathrm{tr}(\widetilde{M}^T\widetilde{M})$ for any $\widetilde{M} \in \mathcal{SP}(n,l)$. Further, the product manifold $\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l)$ can also be viewed as an embedded Riemannian submanifold of $\mathcal{SP}(n,l)\times\mathcal{SP}(2l,l)$. Thus, it can be endowed with a induced Riemannian metric

$$\langle(\widetilde{\xi}_1,\widetilde{\eta}_1),(\widetilde{\xi}_2,\widetilde{\eta}_2)\rangle_{(\widetilde{V},\widetilde{P})} := \langle\widetilde{\xi}_1,\widetilde{\xi}_2\rangle_{\widetilde{V}} + \langle\widetilde{\eta}_1,\widetilde{\eta}_2\rangle_{\widetilde{P}} = \frac{1}{2}\left(\mathrm{tr}(\widetilde{\xi}_1^T\widetilde{\xi}_2) + \mathrm{tr}(\widetilde{\eta}_1^T\widetilde{\eta}_2)\right)$$

for all $(\widetilde{V},\widetilde{P}) \in \mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l)$ and $(\widetilde{\xi}_1,\widetilde{\eta}_1),(\widetilde{\xi}_2,\widetilde{\eta}_2) \in \mathbf{T}_{(\widetilde{V},\widetilde{P})}\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l)$. Without causing any confusion, in what follows we use $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$ to denote the Riemannian metric on $\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l)$ and its induced norm.

The following proposition provides us an alternative characterization of $\mathbf{T}_{(\widetilde{V},\widetilde{P})}\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l)$ which will be used in the sequel of this paper.

Proposition 3.1 *For any $(\widetilde{V},\widetilde{P}) \in Stp(n,l)\times Stp(2l,l)$, define the set $\Omega_{(\widetilde{V},\widetilde{P})}$ by*

$$\Omega_{(\widetilde{V},\widetilde{P})} = \left\{(\widetilde{\zeta},\widetilde{\chi}) \in \mathcal{SP}(n,l)\times\mathcal{SP}(2l,l) : \begin{array}{l}\widetilde{\zeta} = \widetilde{W}_{\widetilde{\zeta}}\widetilde{V} \text{ for some skew-symmetric } \widetilde{W}_{\widetilde{\zeta}} \in \mathcal{SP}(n,n),\\ \widetilde{\chi} = \widetilde{W}_{\widetilde{\chi}}\widetilde{P} \text{ for some skew-symmetric } \widetilde{W}_{\widetilde{\chi}} \in \mathcal{SP}(2l,2l)\end{array}\right\}.$$

Then $\mathbf{T}_{(\widetilde{V},\widetilde{P})}Stp(n,l)\times Stp(2l,l) = \Omega_{(\widetilde{V},\widetilde{P})}$.

Proof For any $(\widetilde{\zeta},\widetilde{\chi}) \in \mathbf{T}_{(\widetilde{V},\widetilde{P})}\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l)$, let us define $\widetilde{W}_{\widetilde{\zeta}} = \widetilde{\Pi}_{\widetilde{V}}\widetilde{\zeta}\widetilde{V}^T - \widetilde{V}\widetilde{\zeta}^T\widetilde{\Pi}_{\widetilde{V}}$ and $\widetilde{W}_{\widetilde{\chi}} = \widetilde{\Pi}_{\widetilde{P}}\widetilde{\chi}\widetilde{P}^T - \widetilde{P}\widetilde{\chi}^T\widetilde{\Pi}_{\widetilde{P}}$, where $\widetilde{\Pi}_{\widetilde{V}} = \widetilde{I}_n - \frac{1}{2}\widetilde{V}\widetilde{V}^T$ and $\widetilde{\Pi}_{\widetilde{P}} = \widetilde{I}_{2l} - \frac{1}{2}\widetilde{P}\widetilde{P}^T$, and $\widetilde{I}_n = \begin{bmatrix} I_n & 0 \\ 0 & I_n\end{bmatrix} \in \mathcal{SP}(n,n)$. Clearly, $\widetilde{W}_{\widetilde{\zeta}}$ and $\widetilde{W}_{\widetilde{\chi}}$ are skew-symmetric matrices and

$(\widetilde{W}_{\widetilde{\zeta}}, \widetilde{W}_{\widetilde{\chi}}) \in \mathcal{SP}(n,n) \times \mathcal{SP}(2l,2l)$. By using the fact that $\widetilde{\zeta}^T\widetilde{V} = -\widetilde{V}^T\widetilde{\zeta}$ and $\widetilde{\chi}^T\widetilde{P} = -\widetilde{P}^T\widetilde{\chi}$, and $(\widetilde{V}, \widetilde{P}) \in \mathrm{Stp}(n,l) \times \mathrm{Stp}(2l,l)$, we have

$$\begin{aligned}\left(\widetilde{W}_{\widetilde{\zeta}}\widetilde{V},\ \widetilde{W}_{\widetilde{\chi}}\widetilde{P}\right) &= \left(\widetilde{\Pi}_{\widetilde{V}}\widetilde{\zeta} - \widetilde{V}\widetilde{\zeta}^T\widetilde{\Pi}_{\widetilde{V}}\widetilde{V},\ \widetilde{\Pi}_{\widetilde{P}}\widetilde{\chi} - \widetilde{P}\widetilde{\chi}^T\widetilde{\Pi}_{\widetilde{P}}\widetilde{P}\right)\\ &= \left(\widetilde{\zeta} - \frac{1}{2}\widetilde{V}\widetilde{V}^T\widetilde{\zeta} - \frac{1}{2}\widetilde{V}\widetilde{\zeta}^T\widetilde{V},\ \widetilde{\chi} - \frac{1}{2}\widetilde{P}\widetilde{P}^T\widetilde{\chi} - \frac{1}{2}\widetilde{P}\widetilde{\chi}^T\widetilde{P}\right)\\ &= (\widetilde{\zeta},\ \widetilde{\chi}).\end{aligned}$$

Therefore, we conclude that $\mathbf{T}_{(\widetilde{V},\widetilde{P})}\mathrm{Stp}(n,l) \times \mathrm{Stp}(2l,l) \subset \Omega_{(\widetilde{V},\widetilde{P})}$.

Conversely, consider $(\widetilde{\zeta},\ \widetilde{\chi}) \in \Omega_{(\widetilde{V},\widetilde{P})}$, then there exist skew-symmetric matrices $\widetilde{W}_{\widetilde{\zeta}} \in \mathcal{SP}(n,n)$ and $\widetilde{W}_{\widetilde{\chi}} \in \mathcal{SP}(2l,2l)$ such that $(\widetilde{\zeta},\ \widetilde{\chi}) = (\widetilde{W}_{\widetilde{\zeta}}\widetilde{V}, \widetilde{W}_{\widetilde{\chi}}\widetilde{P})$. Note that

$$\begin{aligned}\widetilde{\zeta}^T\widetilde{V} + \widetilde{V}^T\widetilde{\zeta} &= (\widetilde{W}_{\widetilde{\zeta}}\widetilde{V})^T\widetilde{V} + \widetilde{V}^T\widetilde{W}_{\widetilde{\zeta}}\widetilde{V} = -\widetilde{V}^T\widetilde{W}_{\widetilde{\zeta}}\widetilde{V} + \widetilde{V}^T\widetilde{W}_{\widetilde{\zeta}}\widetilde{V} = 0,\\ \widetilde{\chi}^T\widetilde{P} + \widetilde{P}^T\widetilde{\chi} &= (\widetilde{W}_{\widetilde{\chi}}\widetilde{P})^T\widetilde{P} + \widetilde{P}^T\widetilde{W}_{\widetilde{\chi}}\widetilde{P} = -\widetilde{P}^T\widetilde{W}_{\widetilde{\chi}}\widetilde{P} + \widetilde{P}^T\widetilde{W}_{\widetilde{\chi}}\widetilde{P} = 0.\end{aligned}$$

Thus, we have that $\Omega_{(\widetilde{V},\widetilde{P})} \subset \mathbf{T}_{(\widetilde{V},\widetilde{P})}\mathrm{Stp}(n,l) \times \mathrm{Stp}(2l,l)$, which completes the proof. □

We now derive the orthogonal projection of any $(\widetilde{M}, \widetilde{N}) \in \mathcal{SP}(n,l) \times \mathcal{SP}(2l,l)$ onto $\mathbf{T}_{(\widetilde{V},\widetilde{P})}\mathrm{Stp}(n,l) \times \mathrm{Stp}(2l,l)$.

Proposition 3.2 *For any $(\widetilde{M}, \widetilde{N}) \in \mathcal{SP}(n,l) \times \mathcal{SP}(2l,l)$ and $(\widetilde{V}, \widetilde{P}) \in Stp(n,l) \times Stp(2l,l)$, the orthogonal projection operator $\mathbf{P}_{(\widetilde{V},\widetilde{P})}$ onto the tangent space $\mathbf{T}_{(\widetilde{V},\widetilde{P})}Stp(n,l) \times Stp(2l,l)$ is given by*

$$\mathbf{P}_{(\widetilde{V},\widetilde{P})}(\widetilde{M}, \widetilde{N}) = \left(\mathbf{P}_{\widetilde{V}}(\widetilde{M}),\ \mathbf{P}_{\widetilde{P}}(\widetilde{N})\right), \tag{3.4}$$

where

$$\begin{aligned}\mathbf{P}_{\widetilde{V}}(\widetilde{M}) &= \widetilde{M} - \widetilde{V}\, sym(\widetilde{V}^T\widetilde{M}),\\ \mathbf{P}_{\widetilde{P}}(\widetilde{N}) &= \widetilde{N} - \widetilde{P}\, sym(\widetilde{P}^T\widetilde{N}).\end{aligned} \tag{3.5}$$

Proof Notice that the linear subspaces $\mathcal{SP}(n,l)$ and $\mathcal{SP}(2l,l)$ is closed under the operations in the right-hand sides of (2.6), due to the right-hand sides of (3.5), it is easy to verify that $\mathbf{P}_{(\widetilde{V},\widetilde{P})}(\widetilde{M}, \widetilde{N}) \in \mathcal{SP}(n,l) \times \mathcal{SP}(2l,l)$.

From the fact that $\widetilde{V}^T\widetilde{V} = \widetilde{I}_l$ and $\widetilde{P}^T\widetilde{P} = \widetilde{I}_l$, we have

$$\begin{aligned}\mathbf{P}_{\widetilde{V}}(\widetilde{M})^T\widetilde{V} + \widetilde{V}^T\mathbf{P}_{\widetilde{V}}(\widetilde{M}) &= \left(\widetilde{M} - \widetilde{V}\mathrm{sym}(\widetilde{V}^T\widetilde{M})\right)^T\widetilde{V} + \widetilde{V}^T\left(\widetilde{M} - \widetilde{V}\mathrm{sym}(\widetilde{V}^T\widetilde{M})\right) = 0,\\ \mathbf{P}_{\widetilde{P}}(\widetilde{N})^T\widetilde{P} + \widetilde{P}^T\mathbf{P}_{\widetilde{P}}(\widetilde{N}) &= \left(\widetilde{N} - \widetilde{P}\mathrm{sym}(\widetilde{P}^T\widetilde{N})\right)^T\widetilde{P} + \widetilde{P}^T\left(\widetilde{N} - \widetilde{P}\mathrm{sym}(\widetilde{P}^T\widetilde{N})\right) = 0,\end{aligned}$$

which implies $\mathbf{P}_{(\widetilde{V},\widetilde{P})}(\widetilde{M}, \widetilde{N}) \in \mathbf{T}_{(\widetilde{V},\widetilde{P})}\mathrm{Stp}(n,l) \times \mathrm{Stp}(2l,l)$. On the other hand, for any $(\widetilde{\xi}, \widetilde{\eta}) \in \mathbf{T}_{(\widetilde{V},\widetilde{P})}\mathrm{Stp}(n,l) \times \mathrm{Stp}(2l,l)$,

$$\begin{aligned}\left\langle(\widetilde{M}, \widetilde{N}) - \mathbf{P}_{(\widetilde{V},\widetilde{P})}(\widetilde{M}, \widetilde{N}), (\widetilde{\xi}, \widetilde{\eta})\right\rangle &= \frac{1}{2}\mathrm{tr}\left(\left(\widetilde{M} - \mathbf{P}_{\widetilde{V}}(\widetilde{M})\right)^T\widetilde{\xi}\right) + \frac{1}{2}\mathrm{tr}\left(\left(\widetilde{N} - \mathbf{P}_{\widetilde{P}}(\widetilde{N})\right)^T\widetilde{\eta}\right)\\ &= \frac{1}{2}\mathrm{tr}\left(\mathrm{sym}(\widetilde{V}^T\widetilde{M})\widetilde{V}^T\widetilde{\xi}\right) + \frac{1}{2}\mathrm{tr}\left(\mathrm{sym}(\widetilde{P}^T\widetilde{N})\widetilde{P}^T\widetilde{\eta}\right)\\ &= 0,\end{aligned}$$

where in the last equality we have used the fact that the trace of the product of symmetric and skew-symmetric matrices is zero. The proof is thus complete. □

3.2 A Retraction and a Vector Transport as the Differentiated Retraction

In this subsection we discuss a retraction and a vector transport on $\mathrm{Stp}(n, l)\times\mathrm{Stp}(2l, l)$, which is quite critical to Riemannian optimization for the problem (2.11). For a given search direction $(\widetilde{\xi}, \widetilde{\eta}) \in \mathbf{T}_{(\widetilde{V},\widetilde{P})}\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$ at a current point $(\widetilde{V}, \widetilde{P}) \in \mathrm{Stp}(n, l)\times\mathrm{Stp}(2l, l)$, a search should be performed on a curve emanating from $(\widetilde{V}, \widetilde{P})$ in the direction of $(\widetilde{\xi}, \widetilde{\eta})$. For this purpose, it is necessary to find a retraction on the product manifold $\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$ in question, which is a map from $\mathbf{T}_{(\widetilde{V},\widetilde{P})}\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$ to $\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$.

In this paper, we adapt the Cayley transform for our retraction which is studied by [15]. The Cayley transform is a mapping between skew-symmetric matrices and special orthogonal matrices. Let W be any $n \times n$ real skew-symmetric matrix. Then $I_n - W$ is invertible. For any $X \in \mathbb{R}^{n\times l}$ and any $\alpha \in \mathbb{R}$, the matrix Y given in the following closed form:

$$Y = QX \quad \text{and} \quad Q := \left(I_n - \frac{\alpha}{2}W\right)^{-1}\left(I_n + \frac{\alpha}{2}W\right)$$

is known as the Cayley transform [15]. Using the skew-symmetry of W and $(I-M)(I+M) = (I + M)(I - M)$ for any square matrix M, one can easily prove that $Y^T Y = X^T X$, which implies that Y preserves the same constrains as X.

From Proposition 3.1, for any $\widetilde{\xi} \in \mathbf{T}_{\widetilde{V}}\mathrm{Stp}(n, l)$, it holds $\widetilde{\xi} = \widetilde{W}_{\widetilde{\xi}}\widetilde{V}$, where $\widetilde{W}_{\widetilde{\xi}} = \widetilde{\Pi}_{\widetilde{V}}\widetilde{\xi}\widetilde{V}^T - \widetilde{V}\widetilde{\xi}^T\widetilde{\Pi}_{\widetilde{V}}$ is a skew-symmetric matrix, and $\widetilde{\Pi}_{\widetilde{V}} = \widetilde{I}_n - \frac{1}{2}\widetilde{V}\widetilde{V}^T$. This leads to a retraction based on the Cayley transform on the manifold $\mathrm{Stp}(n, l)$ of the form

$$\mathcal{R}_{\widetilde{V}}(\alpha\widetilde{\xi}) = \left(\widetilde{I}_n - \frac{\alpha}{2}\widetilde{W}_{\widetilde{\xi}}\right)^{-1}\left(\widetilde{I}_n + \frac{\alpha}{2}\widetilde{W}_{\widetilde{\xi}}\right)\widetilde{V}, \tag{3.6}$$

for any $\alpha \in \mathbb{R}$. It can be shown from [15] that the curve $\Gamma(\alpha) = \mathcal{R}_{\widetilde{V}}(\alpha\widetilde{\xi})$ is contained in $\mathrm{Stp}(n, l)$, and satisfies $\Gamma(0) = \widetilde{V}$ and $\Gamma'(0) = \widetilde{W}_{\widetilde{\xi}}\widetilde{V} = \widetilde{\xi}$. In a similar manner, for any $\widetilde{\eta} \in \mathbf{T}_{\widetilde{P}}\mathrm{Stp}(2l, l)$, we can derive a retraction based on the Cayley transform on the manifold $\mathrm{Stp}(2l, l)$ of the form

$$\mathcal{R}_{\widetilde{P}}(\alpha\widetilde{\eta}) = \left(\widetilde{I}_{2l} - \frac{\alpha}{2}\widetilde{W}_{\widetilde{\eta}}\right)^{-1}\left(\widetilde{I}_{2l} + \frac{\alpha}{2}\widetilde{W}_{\widetilde{\eta}}\right)\widetilde{P}, \tag{3.7}$$

where $\widetilde{\eta} = \widetilde{W}_{\widetilde{\eta}}\widetilde{P}$, and $\widetilde{W}_{\widetilde{\eta}} = \widetilde{\Pi}_{\widetilde{P}}\widetilde{\eta}\widetilde{P}^T - \widetilde{P}\widetilde{\eta}^T\widetilde{\Pi}_{\widetilde{P}}$ with $\widetilde{\Pi}_{\widetilde{P}} = \widetilde{I}_{2l} - \frac{1}{2}\widetilde{P}\widetilde{P}^T$. Therefore, a retraction $\mathcal{R}$ on the product manifold $\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$ is immediately defined as

$$\mathcal{R}_{(\widetilde{V},\widetilde{P})}(\alpha(\widetilde{\xi}, \widetilde{\eta})) = \left(\mathcal{R}_{\widetilde{V}}(\alpha\widetilde{\xi}),\ \ \mathcal{R}_{\widetilde{P}}(\alpha\widetilde{\eta})\right) \tag{3.8}$$

where $(\widetilde{V}, \widetilde{P}) \in \mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$ and $(\widetilde{\xi}, \widetilde{\eta}) \in \mathbf{T}_{(\widetilde{V},\widetilde{P})}(\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l))$.

Noting that the matrix inverse $(\widetilde{I}_n - \frac{\alpha}{2}\widetilde{W}_{\widetilde{\xi}})^{-1}$ dominates the computation of $\mathcal{R}_{\widetilde{V}}(\alpha\widetilde{\xi})$ in (3.6). To overcome this difficulty, according to technique of [15], we can obtain a refined scheme by the following procedure. Rewriting $\widetilde{W}_{\widetilde{\xi}} \in \mathcal{SP}(n, n) \subset \mathbb{R}^{2n\times 2n}$ as $\widetilde{W}_{\widetilde{\xi}} = U_{\widetilde{\xi}}V_{\widetilde{\xi}}^T$, where $U_{\widetilde{\xi}} = [\widetilde{\Pi}_{\widetilde{V}}\widetilde{\xi}\ \ \widetilde{V}] \in \mathbb{R}^{2n\times 4l}$ and $V_{\widetilde{\xi}} = [\widetilde{V}\ \ -\widetilde{\Pi}_{\widetilde{V}}\widetilde{\xi}] \in \mathbb{R}^{2n\times 4l}$, then apply the Sherman–Morrison–Woodbury(SMW) formula

$$\left(\widetilde{I}_n - \frac{\alpha}{2}\widetilde{W}_{\widetilde{\xi}}\right)^{-1} = \left(\widetilde{I}_n - \frac{\alpha}{2}U_{\widetilde{\xi}}V_{\widetilde{\xi}}^T\right)^{-1} = \widetilde{I}_n + \frac{\alpha}{2}U_{\widetilde{\xi}}\left(I_{4l} - \frac{\alpha}{2}V_{\widetilde{\xi}}^T U_{\widetilde{\xi}}\right)^{-1}V_{\widetilde{\xi}}{}^T, \tag{3.9}$$

we can derive the following refined scheme

$$\mathcal{R}_{\widetilde{V}}(\alpha\widetilde{\xi}) = \widetilde{V} + \alpha U_{\widetilde{\xi}}\left(I_{4l} - \frac{\alpha}{2}V_{\widetilde{\xi}}^T U_{\widetilde{\xi}}\right)^{-1}V_{\widetilde{\xi}}^T\widetilde{V}. \tag{3.10}$$

Apparently, if $l \ll n/2$, inverting $I_{4l} - \frac{1}{2} V_{\widetilde{\xi}}^T U_{\widetilde{\xi}} \in \mathbb{R}^{4l \times 4l}$ is much easier than inverting $\widetilde{I}_n - \frac{\alpha}{2} \widetilde{W}_{\widetilde{\xi}} \in \mathbb{R}^{2n \times 2n}$, hence (3.10) should be used to compute $\mathcal{R}_{\widetilde{V}}(\alpha\widetilde{\xi})$. However, if $l \geq n/2$, then (3.10) has no advantage over (3.6), if this is the case, we still use (3.6) to compute $\mathcal{R}_{\widetilde{V}}(\alpha\widetilde{\xi})$. We also note that there is no need to refine $\mathcal{R}_{\widetilde{P}}(\alpha\widetilde{\eta})$ in (3.7) because $\widetilde{I}_{2l} - \frac{\alpha}{2} \widetilde{W}_{\widetilde{\eta}} \in \mathbb{R}^{4l \times 4l}$ but $\widetilde{\eta} \in \mathbb{R}^{4l \times 2l}$.

Remark 3.1 According to the refined formula (3.10), we can form the partition matrix

$$V_{\widetilde{\xi}}^T U_{\widetilde{\xi}} = \begin{bmatrix} \frac{1}{2}\widetilde{V}^T\widetilde{\xi} & \widetilde{I}_l \\ -\widetilde{\xi}^T\widetilde{\xi} + \frac{3}{4}(\widetilde{\xi}^T\widetilde{V})(\widetilde{V}^T\widetilde{\xi}) & -\frac{1}{2}\widetilde{V}^T\widetilde{\xi} \end{bmatrix},$$

and then calculate

$$I_{4l} - \frac{\alpha}{2} V_{\widetilde{\xi}}^T U_{\widetilde{\xi}} = \begin{bmatrix} \widetilde{I}_l - \frac{\alpha}{4}\widetilde{V}^T\widetilde{\xi} & -\frac{\alpha}{2}\widetilde{I}_l \\ \frac{\alpha}{2}\widetilde{\xi}^T\widetilde{\xi} - \frac{3\alpha}{8}(\widetilde{\xi}^T\widetilde{V})(\widetilde{V}^T\widetilde{\xi}) & \widetilde{I}_l + \frac{\alpha}{4}\widetilde{V}^T\widetilde{\xi} \end{bmatrix}.$$

By using the following inverse formula of 2-by-2 block matrix

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} -(C - DB^{-1}A)^{-1}DB^{-1} & (C - DB^{-1}A)^{-1} \\ B^{-1} + B^{-1}A(C - DB^{-1}A)^{-1}DB^{-1} & -B^{-1}A(C - DB^{-1}A)^{-1} \end{bmatrix},$$

where $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{m \times m}$ is invertible, $C \in \mathbb{R}^{n \times n}$, $D \in \mathbb{R}^{n \times m}$ and $C - DB^{-1}A$ is invertible, we have that, if

$$N = \frac{\alpha}{2}\widetilde{\xi}^T\widetilde{\xi} - \frac{3\alpha}{8}(\widetilde{\xi}^T\widetilde{V})(\widetilde{V}^T\widetilde{\xi}) + \frac{2}{\alpha}(\widetilde{I}_l + \frac{\alpha}{4}\widetilde{V}^T\widetilde{\xi})(\widetilde{I}_l - \frac{\alpha}{4}\widetilde{V}^T\widetilde{\xi}) \in \mathbb{R}^{2l \times 2l}$$

is invertible, then $(I_{4l} - \frac{\alpha}{2} V_{\widetilde{\xi}}^T U_{\widetilde{\xi}})^{-1}$ can be expressed as

$$(I_{4l} - \frac{\alpha}{2} V_{\widetilde{\xi}}^T U_{\widetilde{\xi}})^{-1} = \begin{bmatrix} \frac{2}{\alpha}N^{-1}(\widetilde{I}_l + \frac{\alpha}{4}\widetilde{V}^T\widetilde{\xi}) & N^{-1} \\ -\frac{\alpha}{2}\widetilde{I}_l + \frac{4}{\alpha^2}(\widetilde{I}_l - \frac{\alpha}{4}\widetilde{V}^T\widetilde{\xi})N^{-1}(\widetilde{I}_l + \frac{\alpha}{4}\widetilde{V}^T\widetilde{\xi}) & -\frac{2}{\alpha}(\widetilde{I}_l - \frac{\alpha}{4}\widetilde{V}^T\widetilde{\xi})N^{-1} \end{bmatrix}.$$

Then we can further refine the computation of $\mathcal{R}_{\widetilde{V}}(\alpha\widetilde{\xi})$ for the case $l \ll n/2$ as follows

$$\mathcal{R}_{\widetilde{V}}(\alpha\widetilde{\xi}) = \widetilde{V} + \alpha U_{\widetilde{\xi}} \begin{bmatrix} \frac{2}{\alpha}N^{-1}N_{\widetilde{\xi}}^2 & N^{-1} \\ -\frac{\alpha}{2}\widetilde{I}_l + \frac{4}{\alpha^2}N_{\widetilde{\xi}}^3 N^{-1} N_{\widetilde{\xi}}^2 & -\frac{2}{\alpha}N_{\widetilde{\xi}}^3 N^{-1} \end{bmatrix} \begin{bmatrix} \widetilde{I}_l \\ -\frac{1}{2}{N_{\widetilde{\xi}}^1}^T \end{bmatrix}, \tag{3.11}$$

where $N_{\widetilde{\xi}}^1 = \widetilde{V}^T\widetilde{\xi}$, $N_{\widetilde{\xi}}^2 = \widetilde{I}_l + \frac{\alpha}{4}N_{\widetilde{\xi}}^1$, $N_{\widetilde{\xi}}^3 = \widetilde{I}_l - \frac{\alpha}{4}N_{\widetilde{\xi}}^1$ and $N = \frac{\alpha}{2}\widetilde{\xi}^T\widetilde{\xi} - \frac{3\alpha}{8}{N_{\widetilde{\xi}}^1}^T N_{\widetilde{\xi}}^1 + \frac{2}{\alpha}N_{\widetilde{\xi}}^2 N_{\widetilde{\xi}}^3$. We observe from (3.11) that we only need to inverse $N \in \mathbb{R}^{2l \times 2l}$, and there will be no high computational costs for computing (3.11) if we store the matrices $N_{\widetilde{\xi}}^1$, $N_{\widetilde{\xi}}^2$ and $N_{\widetilde{\xi}}^3$ after $\widetilde{\xi}$ is obtained.

We use the differentiated retraction as a vector transport $\mathcal{T}_{(\widetilde{\xi},\widetilde{\eta})}(\widetilde{\zeta}, \widetilde{\chi})$ on $\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$, defined as

$$\begin{aligned} \mathcal{T}_{(\widetilde{\xi},\widetilde{\eta})}(\widetilde{\zeta}, \widetilde{\chi}) &= (\mathcal{T}_{\widetilde{\xi}}(\widetilde{\zeta}),\ \mathcal{T}_{\widetilde{\eta}}(\widetilde{\chi})) = (D\mathcal{R}_{\widetilde{V}}(\widetilde{\xi})[\widetilde{\zeta}],\ D\mathcal{R}_{\widetilde{P}}(\widetilde{\eta})[\widetilde{\chi}]) \\ &= \left(\frac{d}{dt}\mathcal{R}_{\widetilde{V}}(\widetilde{\xi} + t\widetilde{\zeta})\Big|_{t=0},\ \frac{d}{dt}\mathcal{R}_{\widetilde{P}}(\widetilde{\eta} + t\widetilde{\chi})\Big|_{t=0} \right), \end{aligned}$$

for $(\widetilde{\xi}, \widetilde{\eta}), (\widetilde{\zeta}, \widetilde{\chi}) \in \mathbf{T}_{(\widetilde{V},\widetilde{P})}(\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l))$. Consequently, we have

$$\begin{aligned}\mathcal{T}_{\alpha(\widetilde{\xi},\widetilde{\eta})}(\widetilde{\xi},\widetilde{\eta}) &= (\mathcal{T}_{\alpha\widetilde{\xi}}(\widetilde{\xi}), \mathcal{T}_{\alpha\widetilde{\eta}}(\widetilde{\eta})) = (D\mathcal{R}_{\widetilde{V}}(\alpha\widetilde{\xi})[\widetilde{\xi}],\ \ D\mathcal{R}_{\widetilde{P}}(\alpha\widetilde{\eta})[\widetilde{\eta}]) \\ &= \left(\frac{d}{dt}\mathcal{R}_{\widetilde{V}}(t\widetilde{\xi})\Big|_{t=\alpha},\ \ \frac{d}{dt}\mathcal{R}_{\widetilde{P}}(t\widetilde{\eta})\Big|_{t=\alpha}\right). \end{aligned} \tag{3.12}$$

As in [16], we can derive computational formulas for the vector transport as the differentiated retraction of (3.6) and (3.7) by

$$\mathcal{T}_{\widetilde{\xi}}(\widetilde{\zeta}) = \frac{d}{dt}\mathcal{R}_{\widetilde{V}}(\widetilde{\xi}+t\widetilde{\zeta})\Big|_{t=0} = \left(\widetilde{I}_n - \frac{1}{2}\widetilde{W}_{\widetilde{\xi}}\right)^{-1}\widetilde{W}_{\widetilde{\zeta}}\left(\widetilde{I}_n - \frac{1}{2}\widetilde{W}_{\widetilde{\xi}}\right)^{-1}\widetilde{V}$$

and

$$\mathcal{T}_{\widetilde{\eta}}(\widetilde{\chi}) = \frac{d}{dt}\mathcal{R}_{\widetilde{P}}(\widetilde{\eta}+t\widetilde{\chi})\Big|_{t=0} = \left(\widetilde{I}_{2l} - \frac{1}{2}\widetilde{W}_{\widetilde{\eta}}\right)^{-1}\widetilde{W}_{\widetilde{\chi}}\left(\widetilde{I}_{2l} - \frac{1}{2}\widetilde{W}_{\widetilde{\eta}}\right)^{-1}\widetilde{P}.$$

Therefore, we have

$$\mathcal{T}_{\alpha\widetilde{\xi}}(\widetilde{\xi}) = \left(\widetilde{I}_n - \frac{\alpha}{2}\widetilde{W}_{\widetilde{\xi}}\right)^{-2}\widetilde{W}_{\widetilde{\xi}}\widetilde{V} = \left(\widetilde{I}_n - \frac{\alpha}{2}\widetilde{W}_{\widetilde{\xi}}\right)^{-2}\widetilde{\xi}, \tag{3.13}$$

and

$$\mathcal{T}_{\alpha\widetilde{\eta}}(\widetilde{\eta}) = \left(\widetilde{I}_{2l} - \frac{\alpha}{2}\widetilde{W}_{\widetilde{\eta}}\right)^{-2}\widetilde{W}_{\widetilde{\eta}}\widetilde{P} = \left(\widetilde{I}_{2l} - \frac{\alpha}{2}\widetilde{W}_{\widetilde{\eta}}\right)^{-2}\widetilde{\eta}. \tag{3.14}$$

Following from Lemma 2 in [16], we have that the vector transport formulas given by (3.12), (3.13) and (3.14) satisfy the following Ring-Wirth nonexpansive condition

$$\begin{aligned}&\langle \mathcal{T}_{\alpha(\widetilde{\xi},\widetilde{\eta})}(\widetilde{\xi},\widetilde{\eta}),\ \mathcal{T}_{\alpha(\widetilde{\xi},\widetilde{\eta})}(\widetilde{\xi},\widetilde{\eta})\rangle \\ &= \frac{1}{2}\mathrm{tr}\left(\widetilde{\xi}^T\left(\widetilde{I}_n + \frac{\alpha}{2}\widetilde{W}_{\widetilde{\xi}}\right)^{-2}\left(\widetilde{I}_n - \frac{\alpha}{2}\widetilde{W}_{\widetilde{\xi}}\right)^{-2}\widetilde{\xi}\right) \\ &\quad + \frac{1}{2}\mathrm{tr}\left(\widetilde{\eta}^T\left(\widetilde{I}_{2l} + \frac{\alpha}{2}\widetilde{W}_{\widetilde{\eta}}\right)^{-2}\left(\widetilde{I}_{2l} - \frac{\alpha}{2}\widetilde{W}_{\widetilde{\eta}}\right)^{-2}\widetilde{\eta}\right) \\ &= \frac{1}{2}\mathrm{tr}\left(\widetilde{\xi}^T\left(\widetilde{I}_n - \frac{\alpha^2}{4}\widetilde{W}_{\widetilde{\xi}}^2\right)^{-2}\widetilde{\xi}\right) + \frac{1}{2}\mathrm{tr}\left(\widetilde{\eta}^T\left(\widetilde{I}_{2l} - \frac{\alpha^2}{4}\widetilde{W}_{\widetilde{\eta}}^2\right)^{-2}\widetilde{\eta}\right) \\ &\le \langle(\widetilde{\xi},\widetilde{\eta}),(\widetilde{\xi},\widetilde{\eta})\rangle, \end{aligned} \tag{3.15}$$

where the last inequality makes use of the fact that both $\widetilde{W}_{\widetilde{\xi}}$ and $\widetilde{W}_{\widetilde{\eta}}$ are skew-symmetric, implying that their eigenvalues are zeros or pure imaginary numbers, and therefore $\widetilde{I}_n - \frac{\alpha^2}{4}\widetilde{W}_{\widetilde{\xi}}^2$ and $\widetilde{I}_{2l} - \frac{\alpha^2}{4}\widetilde{W}_{\widetilde{\eta}}^2$ are both symmetric matrices with all eigenvalues not less than 1.

For refining the computation of (3.13) for the case $l \ll n/2$, as in [16], applying again the SMW formula (3.9), we can obtain the following refined scheme

$$\mathcal{T}_{\alpha\widetilde{\xi}}(\widetilde{\xi}) = U_{\widetilde{\xi}}\left(M_{\widetilde{\xi}}^1 + \frac{\alpha}{2}M_{\widetilde{\xi}}^2M_{\widetilde{\xi}}^3 + \frac{\alpha}{2}\left(I_{4l} - \frac{\alpha}{2}M_{\widetilde{\xi}}^2\right)^{-1}M_{\widetilde{\xi}}^2M_{\widetilde{\xi}}^3\right), \tag{3.16}$$

where $M_{\widetilde{\xi}}^1 = V_{\widetilde{\xi}}^T\widetilde{V}$, $M_{\widetilde{\xi}}^2 = V_{\widetilde{\xi}}^TU_{\widetilde{\xi}}$ and $M_{\widetilde{\xi}}^3 = (I_{4l} - \frac{\alpha}{2}V_{\widetilde{\xi}}^TU_{\widetilde{\xi}})^{-1}V_{\widetilde{\xi}}^T\widetilde{V}$. One may refer to [16] for the details of derivations. We should observe that there will be no high computational costs for computing (3.12) if we store the matrices $M_{\widetilde{\xi}}^1$, $M_{\widetilde{\xi}}^2$ and $M_{\widetilde{\xi}}^3$ after computing (3.10). For the case $l \ge n/2$, we still use (3.13) to compute the vector transport $\mathcal{T}_{\alpha\widetilde{\xi}}(\widetilde{\xi})$. Similar to $\mathcal{R}_{\widetilde{P}}(\alpha\widetilde{\eta})$, there is no reason to refine the computation of $\mathcal{T}_{\alpha\widetilde{\eta}}(\widetilde{\eta})$ and we still use (3.14) to compute it.

3.3 Riemannian Gradient

We now derive explicit expression of the Riemannian gradient of the cost function $f(\widetilde{V}, \widetilde{P})$ defined in (2.11). To do so, we define the mapping $\overline{F} : \mathcal{SP}(n, l) \times \mathcal{SP}(2l, l) \to \mathcal{SP}(m, l)$ and the cost function $\overline{f} : \mathcal{SP}(n, l) \times \mathcal{SP}(2l, l) \to \mathbb{R}$ by

$$\overline{F}(\widetilde{V}, \widetilde{P}) := [\widetilde{A}\widetilde{V}\ \widetilde{B}\widetilde{V}]J\widetilde{P} \quad \text{and} \quad \overline{f}(\widetilde{V}, \widetilde{P}) := \frac{1}{2}\|\overline{F}(\widetilde{V}, \widetilde{P})\|^2, \quad (\widetilde{V}, \widetilde{P}) \in \mathcal{SP}(n, l) \times \mathcal{SP}(2l, l). \tag{3.17}$$

Thus, the mapping F defined in (2.12) and the cost function $f(\widetilde{V}, \widetilde{P})$ are the restrictions of $\overline{F}$ and $\overline{f}$ onto $\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$. That is, $F = \overline{F}|_{\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l)}$ and $f = \overline{f}|_{\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l)}$. The Euclidean gradient of $\overline{f}$ at a point $(\widetilde{V}, \widetilde{P}) \in \mathcal{SP}(n, l) \times \mathcal{SP}(2l, l)$ is given by

$$\mathrm{grad}\overline{f}(\widetilde{V}, \widetilde{P}) = \left(\frac{\partial}{\partial \widetilde{V}}\overline{f}(\widetilde{V}, \widetilde{P}),\ \frac{\partial}{\partial \widetilde{P}}\overline{f}(\widetilde{V}, \widetilde{P})\right).$$

By a simple computation, we have

$$\frac{\partial}{\partial \widetilde{V}}\overline{f}(\widetilde{V}, \widetilde{P}) = \left[\widetilde{A}^T\overline{F}(\widetilde{V}, \widetilde{P})\ \ \widetilde{B}^T\overline{F}(\widetilde{V}, \widetilde{P})\right]\begin{bmatrix}\widetilde{P_1}^T\\ \widetilde{P_2}^T\end{bmatrix},\ \ \frac{\partial}{\partial \widetilde{P}}\overline{f}(\widetilde{V}, \widetilde{P}) = J^T[\widetilde{A}\widetilde{V}\ \widetilde{B}\widetilde{V}]^T\overline{F}(\widetilde{V}, \widetilde{P}). \tag{3.18}$$

Lemma 3.1 (p. 48 of [11]) *Let $\overline{h}$ be a smooth function defined on a Riemannian manifold $\overline{\mathcal{M}}$ and let h denote the restriction of $\overline{h}$ to a Riemannian submanifold $\mathcal{M}$. The gradient of h is equal to the projection of the gradient of $\overline{h}$ onto $\mathcal{T}_x\mathcal{M}$, that is, grad $h(x) = \mathbf{P}_x\left(\text{grad }\overline{h}(x)\right)$.*

Since $\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$ is a Riemannian submanifold of $\mathcal{SP}(n, l) \times \mathcal{SP}(2l, l)$ endowed with the induced metric. Applying Lemma 3.1, the Riemannian gradient of $f(\widetilde{V}, \widetilde{P})$ at a point $(\widetilde{V}, \widetilde{P}) \in \mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$ is given by

$$\mathrm{grad} f(\widetilde{V}, \widetilde{P}) = \mathbf{P}_{(\widetilde{V},\widetilde{P})}\left(\mathrm{grad}\overline{f}(\widetilde{V}, \widetilde{P})\right) = \left(\mathbf{P}_{\widetilde{V}}\left(\frac{\partial}{\partial \widetilde{V}}\overline{f}(\widetilde{V}, \widetilde{P})\right),\ \mathbf{P}_{\widetilde{P}}\left(\frac{\partial}{\partial \widetilde{P}}\overline{f}(\widetilde{V}, \widetilde{P})\right)\right). \tag{3.19}$$

Following from Proposition 3.2, we have

$$\begin{cases}\mathbf{P}_{\widetilde{V}}\left(\dfrac{\partial}{\partial \widetilde{V}}\overline{f}(\widetilde{V}, \widetilde{P})\right) = \left[\widetilde{A}^T F(\widetilde{V}, \widetilde{P})\ \ \widetilde{B}^T F(\widetilde{V}, \widetilde{P})\right]\begin{bmatrix}\widetilde{P}_1^T\\ \widetilde{P}_2^T\end{bmatrix}\\ \qquad\qquad\qquad - \widetilde{V}\,\mathrm{sym}\left(\widetilde{V}^T\left[\widetilde{A}^T F(\widetilde{V}, \widetilde{P})\ \ \widetilde{B}^T F(\widetilde{V}, \widetilde{P})\right]\begin{bmatrix}\widetilde{P}_1^T\\ \widetilde{P}_2^T\end{bmatrix}\right),\\ \mathbf{P}_{\widetilde{P}}\left(\dfrac{\partial}{\partial \widetilde{P}}\overline{f}(\widetilde{V}, \widetilde{P})\right) = J^T[\widetilde{A}\widetilde{V}\ \ \widetilde{B}\widetilde{V}]^T F(\widetilde{V}, \widetilde{P}) - \widetilde{P}\,\mathrm{sym}\left(\widetilde{P}^T J^T[\widetilde{A}\widetilde{V}\ \ \widetilde{B}\widetilde{V}]^T F(\widetilde{V}, \widetilde{P})\right).\end{cases} \tag{3.20}$$

Together with (3.18) and the map ϕ^{-1}, we can derive the first-order optimality conditions of the problem (2.2) as follows.

Corollary 3.1 *Suppose that (V^*, P^*) is a local minimizer of the problem (2.2). Then (V^*, P^*) satisfies the following first-order optimality conditions:*

$$\left[A^H[AV^*\ \ BV^*]P^*\ \ \ B^H[AV^*\ \ BV^*]P^*\right]\begin{bmatrix}P_1^{*H}\\ P_2^{*H}\end{bmatrix}$$
$$- V^*\left(\left[A^H[AV^*\ \ BV^*]P^*\ \ \ B^H[AV^*\ \ BV^*]P^*\right]\begin{bmatrix}P_1^{*H}\\ P_2^{*H}\end{bmatrix}\right)^H V^* = 0.$$

$$[AV^* \;\; BV^*]^H[AV^* \;\; BV^*]P^* - P^*\Big([AV^* \;\; BV^*]^H[AV^* \;\; BV^*]P^*\Big)^H P^* = 0.$$
$$V^{*H}V^* = I_l, \;\; P^{*H}P^* = I_l, \tag{3.21}$$

where $P^* = \begin{bmatrix} P_1^* \\ P_2^* \end{bmatrix}$ *with* $P_1^*, P_2^* \in \mathbb{C}^{l\times l}$.

Proof The Lagrangian function of the problem (2.11) is given by

$$\mathcal{L} := \mathcal{L}(\widetilde{V}, \widetilde{P}, \widetilde{\Lambda}, \widetilde{\Omega}) = \overline{f}(\widetilde{V}, \widetilde{P}) - \frac{1}{2}\mathrm{tr}(\widetilde{\Lambda}(\widetilde{V}^T\widetilde{V} - \widetilde{I}_l)) - \frac{1}{2}\mathrm{tr}(\widetilde{\Omega}(\widetilde{P}^T\widetilde{P} - \widetilde{I}_l))$$

where $\overline{f} := \overline{f}(\widetilde{V}, \widetilde{P})$ is defined in (3.17) and $\widetilde{\Lambda}, \widetilde{\Omega} \in \mathcal{SP}(l, l)$ are symmetric matrices representing the Lagrange multipliers. The Lagrangian function leads to the first order optimality conditions for the problem (2.11):

$$\frac{\partial}{\partial \widetilde{V}}\mathcal{L} = \frac{\partial}{\partial \widetilde{V}}\overline{f} - \widetilde{V}\widetilde{\Lambda} = 0, \quad \frac{\partial}{\partial \widetilde{P}}\mathcal{L} = \frac{\partial}{\partial \widetilde{P}}\overline{f} - \widetilde{P}\widetilde{\Omega} = 0,$$
$$\frac{\partial}{\partial \widetilde{\Lambda}}\mathcal{L} = \widetilde{V}^T\widetilde{V} - \widetilde{I}_l = 0, \quad \frac{\partial}{\partial \widetilde{\Omega}}\mathcal{L} = \widetilde{P}^T\widetilde{P} - \widetilde{I}_l = 0, \tag{3.22}$$

where $\frac{\partial}{\partial \widetilde{V}}\overline{f}$ and $\frac{\partial}{\partial \widetilde{P}}\overline{f}$ are given in (3.18). From the first two equations in (3.22), we have $\widetilde{\Lambda} = \widetilde{V}^T\frac{\partial}{\partial \widetilde{V}}\overline{f}$ and $\widetilde{\Omega} = \widetilde{P}^T\frac{\partial}{\partial \widetilde{P}}\overline{f}$. Since $\widetilde{\Lambda}$ and $\widetilde{\Omega}$ must be symmetric matrices, we obtain $\widetilde{\Lambda} = \frac{\partial}{\partial \widetilde{V}}\overline{f}^T\widetilde{V}$ and $\widetilde{\Omega} = \frac{\partial}{\partial \widetilde{P}}\overline{f}^T\widetilde{P}$. Then the first-order optimality conditions of the problem (2.11) in the Euclidean sense satisfy

$$\frac{\partial}{\partial \widetilde{V}}\overline{f} - \widetilde{V}\frac{\partial}{\partial \widetilde{V}}\overline{f}^T\widetilde{V} = 0, \quad \frac{\partial}{\partial \widetilde{P}}\overline{f} - \widetilde{P}\frac{\partial}{\partial \widetilde{P}}\overline{f}^T\widetilde{P} = 0, \quad \widetilde{V}^T\widetilde{V} = \widetilde{I}_l, \quad \widetilde{P}^T\widetilde{P} = \widetilde{I}_l. \tag{3.23}$$

By using the map ϕ^{-1}, the first-order optimality conditions of the problem (2.2) can be expressed as (3.21). □

Next, we establish explicit formulas for the differential of $F(\widetilde{V}, \widetilde{P})$ defined in (2.12) and the definition of pullback mappings which will be used in the sequel. For any given $(\widetilde{V}, \widetilde{P}) \in \mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$ and $(\widetilde{\xi}, \widetilde{\eta}) \in \mathbf{T}_{(\widetilde{V}, \widetilde{P})}\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$, we obtain

$$\begin{aligned}\overline{F}(\widetilde{V} + t\widetilde{\xi}, \widetilde{P} + t\widetilde{\eta}) - \overline{F}(\widetilde{V}, \widetilde{P}) &= [\widetilde{A}(\widetilde{V} + t\widetilde{\xi}) \;\; \widetilde{B}(\widetilde{V} + t\widetilde{\xi})]J(\widetilde{P} + t\widetilde{\eta}) - [\widetilde{A}\widetilde{V} \;\; \widetilde{B}\widetilde{V}]J\widetilde{P} \\ &= \big([\widetilde{A}\widetilde{V} \;\; \widetilde{B}\widetilde{V}] + t[\widetilde{A}\widetilde{\xi} \;\; \widetilde{B}\widetilde{\xi}]\big)\, J(\widetilde{P} + t\widetilde{\eta}) - [\widetilde{A}\widetilde{V} \;\; \widetilde{B}\widetilde{V}]J\widetilde{P} \\ &= t[\widetilde{A}\widetilde{\xi} \;\; \widetilde{B}\widetilde{\xi}]J\widetilde{P} + t[\widetilde{A}\widetilde{V} \;\; \widetilde{B}\widetilde{V}]J\widetilde{\eta} + t^2[\widetilde{A}\widetilde{\xi} \;\; \widetilde{B}\widetilde{\xi}]J\widetilde{\eta}.\end{aligned}$$

Since $\overline{F}$ is a smooth mapping between two linear manifolds $\mathcal{SP}(n, l) \times \mathcal{SP}(2l, l)$ and $\mathcal{SP}(m, l)$, and F is the restriction of $\overline{F}$ to an embedded Riemannian submanifold $\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$ of $\mathcal{SP}(n, l) \times \mathcal{SP}(2l, l)$. Hence, the differential $DF(\widetilde{V}, \widetilde{P}) : \mathbf{T}_{(\widetilde{V}, \widetilde{P})}\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l) \to \mathbf{T}_{F(\widetilde{V}, \widetilde{P})}\mathcal{SP}(m, l) \simeq \mathcal{SP}(m, l)$ (see p. 38 of [11]) of $F(\widetilde{V}, \widetilde{P})$ at a point $(\widetilde{V}, \widetilde{P}) \in \mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$ is determined by

$$DF(\widetilde{V}, \widetilde{P})[(\widetilde{\xi}, \widetilde{\eta})] = D\overline{F}(\widetilde{V}, \widetilde{P})[(\widetilde{\xi}, \widetilde{\eta})] = [\widetilde{A}\widetilde{\xi} \;\; \widetilde{B}\widetilde{\xi}]J\widetilde{P} + [\widetilde{A}\widetilde{V} \;\; \widetilde{B}\widetilde{V}]J\widetilde{\eta}. \tag{3.24}$$

for any $(\widetilde{\xi}, \widetilde{\eta}) \in \mathbf{T}_{(\widetilde{V}, \widetilde{P})}\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$. Moreover, for the Riemannian gradient of $f(\widetilde{V}, \widetilde{P})$ and the differential of $F(\widetilde{V}, \widetilde{P})$, we have the following relationship (see p. 185 of [11])

$$\mathrm{grad} f(\widetilde{V}, \widetilde{P}) = (DF(\widetilde{V}, \widetilde{P}))^*[F(\widetilde{V}, \widetilde{P})], \tag{3.25}$$

where $(DF(\widetilde{V}, \widetilde{P}))^*$ denotes the adjoint of the operator $DF(\widetilde{V}, \widetilde{P})$.

Let $\mathbf{T}\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$ denote the tangent bundle of $\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$. As in p.55 of [11], the pullback mappings $\widehat{F} : \mathbf{T}\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l) \to \mathcal{SP}(m, l)$ and $\widehat{f} : \mathbf{T}\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l) \to \mathbb{R}$ of F and f through the retraction $\mathcal{R}$ on $\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$ can be defined by

$$\widehat{F}(\widetilde{\xi}, \widetilde{\eta}) = F(\mathcal{R}(\widetilde{\xi}, \widetilde{\eta})) \ \text{ and } \ \widehat{f}(\widetilde{\xi}, \widetilde{\eta}) = f(\mathcal{R}(\widetilde{\xi}, \widetilde{\eta})), \tag{3.26}$$

for all $(\widetilde{\xi}, \widetilde{\eta}) \in \mathbf{T}\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$, where $\widehat{F}_{(\widetilde{V},\widetilde{P})}$ and $\widehat{f}_{(\widetilde{V},\widetilde{P})}$ are the restrictions of $\widehat{F}$ and $\widehat{f}$ to $\mathbf{T}_{(\widetilde{V},\widetilde{P})}\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$, respectively. Moreover, we have (see p. 55 of [11])

$$DF(\widetilde{V}, \widetilde{P}) = D\widehat{F}_{(\widetilde{V},\widetilde{P})}(0_{(\widetilde{V},\widetilde{P})}), \quad (\widetilde{V}, \widetilde{P}) \in \mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l), \tag{3.27}$$

where $0_{(\widetilde{V},\widetilde{P})}$ is the origin of $\mathbf{T}_{(\widetilde{V},\widetilde{P})}\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$. For the Riemannian gradient of f and the Euclidean gradient of its pullback function $\widehat{f}$, we have (see p. 56 of [11])

$$\mathrm{grad} f(\widetilde{V}, \widetilde{P}) = \mathrm{grad} \widehat{f}_{(\widetilde{V},\widetilde{P})}(0_{(\widetilde{V},\widetilde{P})}). \tag{3.28}$$

4 A Riemannian Conjugate Gradient Algorithm

Recently, a few Riemannian optimization methods for solving eigenproblems have been proposed. For instance, in [18], a truncated conjugate gradient method was presented for the symmetric generalized eigenvalue problem. In [19], a Riemannian trust-region method was proposed for the symmetric generalized eigenproblem. In [20], a Riemannian Newton method was proposed for a kind of a nonlinear eigenvalue problem. In particular, in [12], Zhao, Jin and Bai presented a geometric Polak–Ribière–Polyak-based nonlinear conjugate gradient method for the inverse eigenvalue problem of reconstructing a stochastic matrix from the given spectrum data. In [13], Yao, Bai, Zhao and Ching proposed a Riemannian Fletcher–Reeves conjugate gradient method for the construction of a doubly stochastic matrix with the prescribed spectrum. In [14], Sato and Iwai presented a Riemannian conjugate gradient method for the problem of the singular value decomposition of a matrix. The complex singular value decomposition problem were also considered by Sato and Iwai [9] and Sato [10] based on a Riemannian Newton method and a Riemannian conjugate gradient method, respectively.

Very recently, in [16], Zhu developed a new Riemannian conjugate gradient method for optimization on the Stiefel manifold, which generalized Dai's nonmonotone conjugate gradient method [7] from the Euclidean space to Riemannian manifolds. The author in [16] introduced two novel vector transports associated with the retraction constructed by the Cayley transform [15]. The first one is the differentiated retraction of the Cayley transform and the second one approximates the differentiated matrix exponential by the Cayley transform.

In this section, inspired by the work of [7,16], we present a Riemannian nonmonotone conjugate gradient method with nonmonotone line search technique for solving the problem (2.11). We show the global convergence for the proposed algorithm, whose analytic framework can be regarded as an extension of the proof proposed in [7,16]. In addition, to make the algorithm easy-to-perform, we translate the algorithm into the complex-value form. Finally, we provide a complete algorithm for the original problem (1.3) in the end of this section, which is to find a minimal perturbation for the pencil to have l eigenpairs with $1 \leq l \leq n$.

4.1 A Riemannian Conjugate Gradient Method for the Problem (2.11)

In Dai's nonmonotone conjugate gradient method [7], the parameter β_{k+1} in (3.1) is given by

$$\beta_{k+1} = \frac{\|\nabla f(x_{k+1})\|^2}{\max\{y_k^T\eta_k, -\nabla f(x_k)^T\eta_k\}},$$

where $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$. Aiming at the problem (2.11), the Riemannian generalization of this formula is given by

$$\beta_{k+1} = \frac{\|\mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1})\|^2}{\max\left\{Y_{k+1},\ -\langle \mathrm{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle\right\}},$$

where $Y_{k+1} = \langle \mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ \mathcal{T}_{\alpha_k(\widetilde{\xi}_k,\widetilde{\eta}_k)}(\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle - \langle \mathrm{grad} f(\widetilde{V}^k, \widetilde{P}^k), (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle$. Dai's algorithm is globally convergent with a nonmonotone line search condition. The Riemannian generalization of this nonmonotone condition is

$$\begin{aligned} f(\mathcal{R}_{(\widetilde{V}^k,\widetilde{P}^k)}(\alpha_k(\widetilde{\xi}_k, \widetilde{\eta}_k))) \le \max\{&f(\widetilde{V}^k, \widetilde{P}^k), \cdots, f(\widetilde{V}^{k-h(k)}, \widetilde{P}^{k-h(k)})\} \\ &+ \delta\alpha_k\langle \mathrm{grad} f(\widetilde{V}^k, \widetilde{P}^k), (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle. \end{aligned}$$

On the basis of these arrangements, the corresponding Riemannian nonmonotone conjugate gradient method for the problem (2.11) is described as follows.

Algorithm 4.1 A Riemannian nonmonotone conjugate gradient method for the problem (2.11)

```
Require: $(\widetilde{V}^0, \widetilde{P}^0) \in \mathrm{Stp}(n,l) \times \mathrm{Stp}(2l,l)$, $(\widetilde{\xi}_0, \widetilde{\eta}_0) = -\mathrm{grad} f(\widetilde{V}^0, \widetilde{P}^0)$. $\epsilon, \delta, \rho \in (0,1)$,
    $h \in \mathbb{N}^+$. $\alpha_{\max} > \overline{\alpha}_0 > \alpha_{\min} > 0$. $k := 0$.
while $\|\mathrm{grad} f(\widetilde{V}^k, \widetilde{P}^k)\| > \epsilon$ do
    Compute the least non-negative integer $s$ such that the steplength $\alpha_k = \overline{\alpha}_k\rho^s$ satisfies
        $f(\mathcal{R}_{(\widetilde{V}^k,\widetilde{P}^k)}(\alpha_k(\widetilde{\xi}_k, \widetilde{\eta}_k)))$
        $\le \max\{f(\widetilde{V}^k, \widetilde{P}^k), \ldots, f(\widetilde{V}^{k-h(k)}, \widetilde{P}^{k-h(k)})\} + \delta\alpha_k\langle \mathrm{grad} f(\widetilde{V}^k, \widetilde{P}^k), (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle,$   (4.1)
    where $h(k) = \min(h-1, k)$ and $\overline{\alpha}_k$ denotes the initial steplength at iteration $k$.
    Set $(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}) = \mathcal{R}_{(\widetilde{V}^k,\widetilde{P}^k)}(\alpha_k(\widetilde{\xi}_k, \widetilde{\eta}_k))$.
    Compute
        $(\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1}) = -\mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}) + \beta_{k+1}\mathcal{T}_{\alpha_k(\widetilde{\xi}_k,\widetilde{\eta}_k)}(\widetilde{\xi}_k, \widetilde{\eta}_k)$   (4.2)
    and
        $Y_{k+1} = \langle \mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ \mathcal{T}_{\alpha_k(\widetilde{\xi}_k,\widetilde{\eta}_k)}(\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle - \langle \mathrm{grad} f(\widetilde{V}^k, \widetilde{P}^k), (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle$   (4.3)
    where $\beta_{k+1} \in (0, \beta_{k+1}^D)$ with
        $\beta_{k+1}^D = \frac{\|\mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1})\|^2}{\max\left\{Y_{k+1},\ -\langle \mathrm{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle\right\}}.$   (4.4)
    Update $\overline{\alpha}_{k+1} \in (\alpha_{\min}, \alpha_{\max})$ and set $k \leftarrow k+1$.
end while
```

We have several remarks for the above algorithm as follows.

Remark 4.1 In Algorithm 4.1, the initial steplength at the kth iteration is set to be $\overline{\alpha}_k$. We now derive a reasonable choice of this initial steplength. For the nonlinear mapping $\widehat{F}_{(\widetilde{V}^k,\widetilde{P}^k)}(\widetilde{\xi},\widetilde{\eta}) : \mathbf{T}_{(\widetilde{V}^k,\widetilde{P}^k)}\mathrm{Stp}(n,l) \times \mathrm{Stp}(2l,l) \to \mathcal{SP}(m,l)$ defined in (3.26), we consider the first-order approximation of $\widehat{F}_{(\widetilde{V}^k,\widetilde{P}^k)}$ at the origin $0_{(\widetilde{V}^k,\widetilde{P}^k)}$ along the search direction $(\widetilde{\xi}_k,\widetilde{\eta}_k)$:

$$\widehat{F}_{(\widetilde{V}^k,\widetilde{P}^k)}(\alpha(\widetilde{\xi}_k,\widetilde{\eta}_k)) \approx \widehat{F}_{(\widetilde{V}^k,\widetilde{P}^k)}(0_{(\widetilde{V}^k,\widetilde{P}^k)}) + \alpha D\widehat{F}_{(\widetilde{V}^k,\widetilde{P}^k)}(0_{(\widetilde{V}^k,\widetilde{P}^k)})[(\widetilde{\xi}_k,\widetilde{\eta}_k)], \tag{4.5}$$

for all α close to 0. From (3.26) and (3.27), this is reduced to

$$F(R_{(\widetilde{V}^k,\widetilde{P}^k)}(\alpha(\widetilde{\xi}_k,\widetilde{\eta}_k))) \approx F(\widetilde{V}^k,\widetilde{P}^k) + \alpha DF(\widetilde{V}^k,\widetilde{P}^k)[(\widetilde{\xi}_k,\widetilde{\eta}_k)],$$

for all α close to 0. From (3.26) and (4.5), we have

$$\begin{aligned}\widehat{f}_{(\widetilde{V}^k,\widetilde{P}^k)}(\alpha(\widetilde{\xi}_k,\widetilde{\eta}_k)) &= \frac{1}{2}\|\widehat{F}_{(\widetilde{V}^k,\widetilde{P}^k)}(\alpha(\widetilde{\xi}_k,\widetilde{\eta}_k))\|^2 \\ &\approx \frac{1}{2}\|\widehat{F}_{(\widetilde{V}^k,\widetilde{P}^k)}(0_{(\widetilde{V}^k,\widetilde{P}^k)})\|^2 + \frac{1}{2}\alpha^2\|D\widehat{F}_{(\widetilde{V}^k,\widetilde{P}^k)}(0_{(\widetilde{V}^k,\widetilde{P}^k)})[(\widetilde{\xi}_k,\widetilde{\eta}_k)]\|^2 \\ &\quad + \alpha\left\langle (D\widehat{F}_{(\widetilde{V}^k,\widetilde{P}^k)}(0_{(\widetilde{V}^k,\widetilde{P}^k)}))^*\left[\widehat{F}_{(\widetilde{V}^k,\widetilde{P}^k)}(0_{(\widetilde{V}^k,\widetilde{P}^k)})\right],\ (\widetilde{\xi}_k,\widetilde{\eta}_k)\right\rangle.\end{aligned}$$

where $(D\widehat{F}_{(\widetilde{V}^k,\widetilde{P}^k)}(0_{(\widetilde{V}^k,\widetilde{P}^k)}))^*$ denotes the adjoint of the operator $D\widehat{F}_{(\widetilde{V}^k,\widetilde{P}^k)}(0_{(\widetilde{V}^k,\widetilde{P}^k)})$. Based on (3.26) and (3.27), this yields

$$\begin{aligned} & f\left(R_{(\widetilde{V}^k,\widetilde{P}^k)}(\alpha(\widetilde{\xi}_k,\widetilde{\eta}_k))\right) \\ &\approx f(\widetilde{V}^k,\widetilde{P}^k) + \alpha\langle (DF(\widetilde{V}^k,\widetilde{P}^k))^*\left[F(\widetilde{V}^k,\widetilde{P}^k)\right],\ (\widetilde{\xi}_k,\widetilde{\eta}_k)\rangle \\ &\quad + \frac{1}{2}\alpha^2\|DF(\widetilde{V}^k,\widetilde{P}^k)[(\widetilde{\xi}_k,\widetilde{\eta}_k)]\|^2. \end{aligned} \tag{4.6}$$

From (3.25), we obtain the local approximation of $f(\widetilde{V},\widetilde{P})$ at $(\widetilde{V}^k,\widetilde{P}^k)$ along the direction $(\widetilde{\xi}_k,\widetilde{\eta}_k)$ as follows:

$$\varphi_k(\alpha) := f(\widetilde{V}^k,\widetilde{P}^k) + \alpha\langle \mathrm{grad} f(\widetilde{V}^k,\widetilde{P}^k),\ (\widetilde{\xi}_k,\widetilde{\eta}_k)\rangle + \frac{1}{2}\alpha^2\|DF(\widetilde{V}^k,\widetilde{P}^k)[(\widetilde{\xi}_k,\widetilde{\eta}_k)]\|^2.$$

Let $\varphi_k'(\alpha) = 0$, we arrive at

$$\langle \mathrm{grad} f(\widetilde{V}^k,\widetilde{P}^k),\ (\widetilde{\xi}_k,\widetilde{\eta}_k)\rangle + \alpha\|DF(\widetilde{V}^k,\widetilde{P}^k)[(\widetilde{\xi}_k,\widetilde{\eta}_k)]\|^2 = 0.$$

Then we have

$$\alpha = -\frac{\langle \mathrm{grad} f(\widetilde{V}^k,\widetilde{P}^k),\ (\widetilde{\xi}_k,\widetilde{\eta}_k)\rangle}{\|DF(\widetilde{V}^k,\widetilde{P}^k)[(\widetilde{\xi}_k,\widetilde{\eta}_k)]\|^2}.$$

Hence, from (3.24), we derive a suitable choice of the initial steplength as

$$\overline{\alpha}_k := \frac{|\langle \mathrm{grad} f(\widetilde{V}^k,\widetilde{P}^k),\ (\widetilde{\xi}_k,\widetilde{\eta}_k)\rangle|}{\|[\widetilde{A}\widetilde{V}^k\ \ \widetilde{B}\widetilde{V}^k]J\widetilde{\eta}_k + [\widetilde{A}\widetilde{\xi}_k\ \ \widetilde{B}\widetilde{\xi}_k]J\widetilde{P}^k\|^2}. \tag{4.7}$$

Remark 4.2 Similar to unconstrained optimization in the Euclidean space, $\|\mathrm{grad} f(\widetilde{V}^k,\widetilde{P}^k)\| < \epsilon$ for some constant $\epsilon > 0$ is a reasonable stopping criterion for Riemannian optimization approaches to the problem (2.11). In fact, under the conditions $\widetilde{V}^T\widetilde{V} = \widetilde{I}_l$ and $\widetilde{P}^T\widetilde{P} = \widetilde{I}_l$,

$\frac{\partial}{\partial \widetilde{V}}\mathcal{L} = 0$ and $\frac{\partial}{\partial \widetilde{P}}\mathcal{L} = 0$ in the proof of Corollary 3.1 are equivalent to $\mathrm{grad} f(\widetilde{V}, \widetilde{P}) = 0$, since it follows from (3.19), (3.20), (3.22) and (3.23) that

$$\mathbf{P}_{\widetilde{V}}\Big(\frac{\partial}{\partial \widetilde{V}}\overline{f}\Big) = (\widetilde{I}_l - \frac{1}{2}\widetilde{V}\widetilde{V}^T)\Big(\frac{\partial}{\partial \widetilde{V}}\overline{f} - \widetilde{V}\frac{\partial}{\partial \widetilde{V}}\overline{f}^T\widetilde{V}\Big) = (\widetilde{I}_l - \frac{1}{2}\widetilde{V}\widetilde{V}^T)\frac{\partial}{\partial \widetilde{V}}\mathcal{L},$$
$$\mathbf{P}_{\widetilde{P}}\Big(\frac{\partial}{\partial \widetilde{P}}\overline{f}\Big) = (\widetilde{I}_l - \frac{1}{2}\widetilde{P}\widetilde{P}^T)\Big(\frac{\partial}{\partial \widetilde{P}}\overline{f} - \widetilde{P}\frac{\partial}{\partial \widetilde{P}}\overline{f}^T\widetilde{P}\Big) = (\widetilde{I}_l - \frac{1}{2}\widetilde{P}\widetilde{P}^T)\frac{\partial}{\partial \widetilde{P}}\mathcal{L}.$$

Thus, first-order critical points in the Euclidean sense can be interpreted as stationary points in the Riemannian sense.

4.2 Global Convergence

In this subsection we study the convergence of Algorithm 4.1. Note that the convergence of the nonlinear conjugate gradient method with nonmonotone line search technique for unconstraint optimization has been well studied in the literature [7]. Considering that the problem (2.11) is a model with matrix variables while most of the nonmonotone CG-type literature is in the context of vectors, and to make the paper self-contained, we here provide more details about the convergence analysis of Algorithm 4.1 in solving the problem (2.2) for the purpose of completeness. The following analytic framework can be regarded as an extension of the proof proposed in [7,16]. In particular, the author in [16] generalized Dai's nonmonotone conjugate gradient method [7] from the Euclidean space to Riemannian manifold, while here we further extend to the Riemannian product manifolds, which appears to be more general.

Lemma 4.1 *For any point $(\widetilde{V}^0, \widetilde{P}^0) \in Stp(n,l) \times Stp(2l,l)$, the level set*

$$\Phi := \{(\widetilde{V}, \widetilde{P}) \in Stp(n,l) \times Stp(2l,l) | f(\widetilde{V}, \widetilde{P}) \leq f(\widetilde{V}^0, \widetilde{P}^0)\} \tag{4.8}$$

is compact.

The proof to Lemma 4.1 can be found in “Appendix 7.1”. The following remark shows some basic properties of the involved product manifold $\mathrm{Stp}(n,l) \times \mathrm{Stp}(l,l)$, and will provide the transparency in our coming convergence analysis.

Remark 4.3 One can see that $\mathrm{Stp}(n,l) \times \mathrm{Stp}(l,l)$ is an embedded Riemannian submanifold of $\mathcal{SP}(n,l) \times \mathcal{SP}(2l,l)$. We may rely on the natural inclusion $\mathbf{T}_{(\widetilde{V},\widetilde{P})}\mathrm{Stp}(n,l) \times \mathrm{Stp}(2l,l)$. Thus, the Riemannian gradient $\mathrm{grad} f(\widetilde{V}, \widetilde{P})$ can be viewed as a continuous nonlinear mapping between $\mathrm{Stp}(n,l) \times \mathrm{Stp}(l,l)$ and $\mathcal{SP}(n,l) \times \mathcal{SP}(2l,l)$. Since Φ is a compact set, then there exists a constant $\gamma > 0$ such that

$$\|\mathrm{grad} f(\widetilde{V}, \widetilde{P})\| \leq \widetilde{\gamma}, \quad \forall (\widetilde{V}, \widetilde{P}) \in \Phi. \tag{4.9}$$

Moreover, for any point $(\widetilde{V}_1, \widetilde{P}_1), (\widetilde{V}_2, \widetilde{P}_2) \in \mathrm{Stp}(n,l) \times \mathrm{Stp}(2l,l)$, the operation grad $f(\widetilde{V}_1, \widetilde{P}_1) - \mathrm{grad} f(\widetilde{V}_2, \widetilde{P}_2)$ is meaningful since both $\mathrm{grad} f(\widetilde{V}_1, \widetilde{P}_1)$ and $\mathrm{grad} f(\widetilde{V}_2, \widetilde{P}_2)$ can be treated as vectors in $\mathcal{SP}(n,l) \times \mathcal{SP}(2l,l)$. Since Φ is compact, then $\mathrm{grad} f(\widetilde{V}, \widetilde{P})$ is Lipschitz continuous on Φ, i.e., there exists a constant $\widetilde{L} > 0$ such that

$$\|\mathrm{grad} f(\widetilde{V}_1, \widetilde{P}_1) - \mathrm{grad} f(\widetilde{V}_2, \widetilde{P}_2)\| \leq \widetilde{L}\mathrm{dist}((\widetilde{V}_1, \widetilde{P}_1) - (\widetilde{V}_2, \widetilde{P}_2)), \quad \forall (\widetilde{V}, \widetilde{P}) \in \Phi \tag{4.10}$$

for all $(\widetilde{V}_1, \widetilde{P}_1), (\widetilde{V}_2, \widetilde{P}_2) \in \Phi \subset \mathrm{Stp}(n,l) \times \mathrm{Stp}(2l,l)$, where “dist” defines the Riemannian distance on $\mathrm{Stp}(n,l) \times \mathrm{Stp}(2l,l)$.

The following lemmas show some properties of the nonmonotone line search. Their proofs can be found in "Appendix 7.2 and 7.3" respectively. Lemma 4.2 shows that Algorithm 4.1 produces a descent direction at each iteration. Lemma 4.3 shows that for Algorithm 4.1 using the technique the nonmonotone line search, the function need not decrease at every iteration, but there must be a reduction in the function every $h - 1$ iterations.

Lemma 4.2 *Suppose Algorithm 4.1 does not terminate in finitely many iterations. Then we have, for all k,*

$$\langle grad f(\widetilde{V}^k, \widetilde{P}^k), (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle < 0. \tag{4.11}$$

Therefore $\beta_{k+1}^D > 0$ and $\beta_{k+1} \in [0, \beta_{k+1}^D]$ is well defined.

Lemma 4.3 *There exists a positive constant $\mu > 0$ for Algorithm 4.1 such that*

$$\alpha_k \geq \min\left\{\overline{\alpha}_k, -\mu\frac{\langle grad f(\widetilde{V}^k, \widetilde{P}^k), \ \ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle}{\|(\widetilde{\xi}_k, \widetilde{\eta}_k)\|^2}\right\}, \ \textit{for all } k, \tag{4.12}$$

where $\overline{\alpha}_k$ denotes the initial step length at iteration k. Further, we have

$$\sum_{j\geq 1}\min_{i=1,\ldots,h}\left\{-\alpha_{hj+i-2}\left\langle grad\ f(\widetilde{V}^{hj+i-2}, \widetilde{P}^{hj+i-2}), \ \ (\widetilde{\xi}_{hj+i-2}, \widetilde{\eta}_{hj+i-2})\right\rangle\right\} < +\infty. \tag{4.13}$$

We now establish the global convergence of Algorithm 4.1 by the idea of the proof of Theorem 3.4 of [7] and Theorem 1 of [16]. The proof can be found in "Appendix 7.4".

Theorem 4.1 *Suppose Algorithm 4.1 does not terminate in finitely many iterations. Then the sequence $\{(\widetilde{V}^k, \widetilde{P}^k)\}$ generated by Algorithm 4.1 converges in the sense that*

$$\mathop{lim\ inf}_{k\to\infty}\|grad f(\widetilde{V}^k, \widetilde{P}^k)\| = 0. \tag{4.14}$$

Hence there exists at least one accumulation point which is a first-order critical point.

4.3 A Riemannian Conjugate Gradient Method for the Problem (2.2)

In the implementation of Algorithm 4.1, as we can observe from (2.5), treating matrices on $\mathrm{Stp}(n, l)$ needs twice as much computer memory as those on $\mathrm{St}(n, l, \mathbb{C})$. Also, addition and multiplication of matrices on $\mathrm{Stp}(n, l)$ need about twice as much computation time as those on $\mathrm{St}(n, l, \mathbb{C})$. To avoid these difficulties, through the map ϕ^{-1}, we now translate Algorithm 4.1 for the problem (2.11) into Riemannian conjugate gradient method for the problem (2.2) on $\mathrm{St}(n, l, \mathbb{C}) \times \mathrm{St}(2l, l, \mathbb{C})$.

In the process of translation, the relations (2.6) are used, together with the relation for $E \in \mathbb{C}^{l\times l}$ and $\widetilde{E} = \phi(E) \in \mathcal{SP}(l, l)$,

$$\mathrm{sym}(\widetilde{E}) = \frac{\widetilde{E} + \widetilde{E}^T}{2} \longleftrightarrow \frac{E + E^H}{2} = \mathrm{her}(E),$$

where $\mathrm{her}(\cdot)$ denotes the Hermitian part of the matrix in the parentheses. Further, by using the relationships between matrix operations on $\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$ and $\mathrm{St}(n, l, \mathbb{C}) \times \mathrm{St}(2l, l, \mathbb{C})$, the retraction $\mathcal{R}$ given in (3.8) on $\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$ correspond to the retraction $\mathcal{R}^{\mathbb{C}}$ on $\mathrm{St}(n, l, \mathbb{C}) \times \mathrm{St}(2l, l, \mathbb{C})$ defined by

$$\mathcal{R}_{(V,P)}^{\mathbb{C}}(\alpha(\xi, \eta)) = \left(\mathcal{R}_V^{\mathbb{C}}(\alpha\xi), \ \ \mathcal{R}_P^{\mathbb{C}}(\alpha\eta)\right) \tag{4.15}$$

where

$$(V, P) \in \mathrm{St}(n, l, \mathbb{C}) \times \mathrm{St}(2l, l, \mathbb{C}), \quad (\xi, \eta) \in \mathbf{T}_{(V,P)}\mathrm{St}(n, l, \mathbb{C}) \times \mathrm{St}(2l, l, \mathbb{C})$$

and

$$\begin{aligned} \mathcal{R}_V^{\mathbb{C}}(\alpha\xi) &= \left(I_n - \frac{\alpha}{2}W_\xi\right)^{-1}\left(I_n + \frac{\alpha}{2}W_\xi\right)V, \\ \mathcal{R}_P^{\mathbb{C}}(\alpha\eta) &= \left(I_{2l} - \frac{\alpha}{2}W_\eta\right)^{-1}\left(I_{2l} + \frac{\alpha}{2}W_\eta\right)P, \end{aligned} \tag{4.16}$$

with

$$W_\xi = \Pi_V \xi V^H - V\xi^H \Pi_V, \quad \Pi_V = I_n - \frac{1}{2}VV^H \tag{4.17}$$

and

$$W_\eta = \Pi_P \eta P^H - P\eta^H \Pi_P, \quad \Pi_P = I_{2l} - \frac{1}{2}PP^H. \tag{4.18}$$

Similarly, the vector transport $\mathcal{T}$ given in (3.12) on $\mathrm{Stp}(n, l) \times \mathrm{Stp}(2l, l)$ correspond to the retraction $\mathcal{T}^{\mathbb{C}}$ on $\mathrm{St}(n, l, \mathbb{C}) \times \mathrm{St}(2l, l, \mathbb{C})$ defined by

$$\mathcal{T}^{\mathbb{C}}_{\alpha(\xi,\eta)}(\xi, \eta) = (\mathcal{T}^{\mathbb{C}}_{\alpha\xi}(\xi), \;\; \mathcal{T}^{\mathbb{C}}_{\alpha\eta}(\eta)) \tag{4.19}$$

where

$$\mathcal{T}^{\mathbb{C}}_{\alpha\xi}(\xi) = \left(I_n - \frac{\alpha}{2}W_\xi\right)^{-2}\xi, \quad \mathcal{T}^{\mathbb{C}}_{\alpha\eta}(\eta) = \left(I_{2l} - \frac{\alpha}{2}W_\eta\right)^{-2}\eta. \tag{4.20}$$

Thus, Algorithm 4.1 is translated to Algorithm 4.2 for the problem (2.2), which provides a Riemannian conjugate gradient method for the problem (2.2). Here, we use the notation $(\overline{\xi}_k, \overline{\eta}_k)$ to express $\phi^{-1}(\mathrm{grad} f(\widetilde{V}^k, \widetilde{P}^k))$.

Algorithm 4.2 A Riemannian nonmonotone conjugate gradient method to the problem (2.2)

```
Require: $(V^0, P^0) \in \mathrm{St}(n, l, \mathbb{C}) \times \mathrm{St}(2l, l, \mathbb{C})$. $\epsilon, \delta, \rho \in (0, 1)$, $h \in \mathbb{N}^+$. $\alpha_{\max} > \overline{\alpha}_0 > \alpha_{\min} > 0$. $k := 0$.
Partition $P^0$ by $P^0 = \begin{bmatrix} P_1^0 \\ P_2^0 \end{bmatrix}$ with $P_1^0, P_2^0 \in \mathbb{C}^{l\times l}$. Compute
    $S_{V,0} = \left[A^H[AV^0 \;\; BV^0]P^0 \;\; B^H[AV^0 \;\; BV^0]P^0\right]\begin{bmatrix} P_1^{0H} \\ P_2^{0H} \end{bmatrix}$, $S_{P,0}=[AV^0 \;\; BV^0]^H[AV^0 \;\; BV^0]P^0$.
    Set $(\overline{\xi}_0, \overline{\eta}_0) = \left(S_{V,0} - V^0\mathrm{her}(V^{0H}S_{V,0}), \; S_{P,0} - P^0\mathrm{her}(P^{0H}S_{P,0})\right)$ and $(\xi_0, \eta_0) = -(\overline{\xi}_0, \overline{\eta}_0)$.
while $\|(\overline{\xi}_k, \overline{\eta}_k)\| > \epsilon$ do
    Compute the least non-negative integer $s$ such that the steplength $\alpha_k = \overline{\alpha}_k\rho^s$ satisfies
        $f\left(\mathcal{R}^{\mathbb{C}}_{(V^k,P^k)}(\alpha_k(\xi_k, \eta_k))\right)$
        $\leq \max\left\{f(V^k, P^k), \ldots, f(V^{k-h(k)}, P^{k-h(k)})\right\} + \delta\alpha_k\left(\mathrm{Tr}(\overline{\xi}_k^H\xi_k) + \mathrm{Tr}(\overline{\eta}_k^H\eta_k)\right)$,
    where $h(k) = \min(h-1, k)$.
    Compute the next iterate $(V^{k+1}, P^{k+1}) := \mathcal{R}^{\mathbb{C}}_{(V^k,P^k)}(\alpha_k(\xi_k, \eta_k))$, where $\mathcal{R}^{\mathbb{C}}$ is the retraction on $\mathrm{St}(n, l, \mathbb{C}) \times \mathrm{St}(2l, l, \mathbb{C})$ defined in (4.15)–(4.18).
```

```
5:  Partition $P^{k+1}$ by $P^{k+1} = \begin{bmatrix} P_1^{k+1} \\ P_2^{k+1} \end{bmatrix}$ with $P_1^{k+1}, P_2^{k+1} \in \mathbb{C}^{l \times l}$. Compute
        $S_{V,k+1} = \left[ A^H [AV^{k+1} \;\; BV^{k+1}] P^{k+1} \;\; B^H [AV^{k+1} \;\; BV^{k+1}] P^{k+1} \right] \begin{bmatrix} {P_1^{k+1}}^H \\ {P_2^{k+1}}^H \end{bmatrix},$
        $S_{P,k+1} = [AV^{k+1} \;\; BV^{k+1}]^H [AV^{k+1} \;\; BV^{k+1}] P^{k+1},$
    and compute
        $(\overline{\xi}_{k+1}, \overline{\eta}_{k+1}) = \left( S_{V,k+1} - V^{k+1} \mathrm{her}({V^{k+1}}^H S_{V,k+1}),\; S_{P,k+1} - P^{k+1} \mathrm{her}({P^{k+1}}^H S_{P,k+1}) \right).$
6:  Compute
        $(\xi_{k+1}, \eta_{k+1}) = -(\overline{\xi}_{k+1}, \overline{\eta}_{k+1}) + \beta_{k+1} \mathcal{T}^{\mathbb{C}}_{\alpha_k(\xi_k,\eta_k)}(\xi_k, \eta_k)$
    where $\mathcal{T}^{\mathbb{C}}$ is the vector transport on $\mathrm{St}(n, l, \mathbb{C}) \times \mathrm{St}(2l, l, \mathbb{C})$ defined in (4.19)–(4.20),
    and $\beta_{k+1} \in (0, \beta^D_{k+1})$ with
        $\beta^D_{k+1} = \frac{\mathrm{Tr}(\overline{\xi}^H_{k+1} \overline{\xi}_{k+1}) + \mathrm{Tr}(\overline{\eta}^H_{k+1} \overline{\eta}_{k+1})}{\max \left\{ \mathrm{Tr}(\overline{\xi}^H_{k+1} \mathcal{T}^{\mathbb{C}}_{\alpha_k \xi_k}(\xi_k)) + \mathrm{Tr}(\overline{\eta}^H_{k+1} \mathcal{T}^{\mathbb{C}}_{\alpha_k \eta_k}(\eta_k)) - (\mathrm{Tr}(\overline{\xi}^H_k \xi_k) + \mathrm{Tr}(\overline{\eta}^H_k \eta_k)),\;\; -(\mathrm{Tr}(\overline{\xi}^H_k \xi_k) + \mathrm{Tr}(\overline{\eta}^H_k \eta_k)) \right\}}.$
7:  Update $\overline{\alpha}_{k+1} \in (\alpha_{\min}, \; \alpha_{\max})$ and set $k \leftarrow k + 1$.
8: end while
```

Remark 4.4 In the implementation of Algorithm 4.2, if $l \ll n/2$, according to (3.10), the computation of V_{α_k} in (4.16) is replaced by its refined scheme

$$\mathcal{R}^{\mathbb{C}}_{V^k}(\alpha_k \xi_k) = V^k + \alpha_k U_{\xi_k} \left(I_{2l} - \frac{\alpha_k}{2} V^H_{\xi_k} U_{\xi_k} \right)^{-1} V^H_{\xi_k} V^k,$$

where $U_{\xi_k} = (\Pi_{V^k} \xi_k, \;\; V^k) \in \mathbb{C}^{n \times 2l}$ and $V_{\xi_k} = (V^k, \;\; -\Pi_{V^k} \xi_k) \in \mathbb{C}^{n \times 2l}$. Similarly, if $l \ll n/2$, according to (3.16), the computation of $\mathcal{T}^{\mathbb{C}}_{\alpha_k \xi_k}(\xi_k)$ in (4.20) is replaced by

$$\mathcal{T}^{\mathbb{C}}_{\alpha_k \xi_k}(\xi_k) = U_{\xi_k} \left(M^1_{\xi_k} + \frac{\alpha_k}{2} M^2_{\xi_k} M^3_{\xi_k} + \frac{\alpha_k}{2} \left(I_{2l} - \frac{\alpha_k}{2} M^2_{\xi_k} \right)^{-1} M^2_{\xi_k} M^3_{\xi_k} \right),$$

where $M^1_{\xi_k} = V^H_{\xi_k} V^k$, $M^2_{\xi_k} = V^H_{\xi_k} U_{\xi_k}$ and $M^3_{\xi_k} = (I_{2l} - \frac{\alpha_k}{2} V^H_{\xi_k} U_{\xi_k})^{-1} V^H_{\xi_k} V^k$.

4.4 An Algorithm for Solving the Problem (1.3)

Since our ultimate goal is to find an optimal pencil $(\hat{A}, \hat{B})$ with minimal perturbation in the problem (1.3) in the sense that $(\hat{A}, \hat{B})$ has l distinct eigenvalues. In this subsection, we first review roughly how to obtain such an optimal pencil once the optimization problem (2.2) is solved, one may refer to pages 781–782 in [2] for more details, and then we present a complete algorithm for the problem (1.3). According to the Proofs of Theorems 1 and 4 and pages 781–782 in [2], such a pencil can be obtained by the following procedure.

Let $V^* \in \mathbb{C}^{n \times l}$ with $V^{*H} V^* = I_l$ and $P^* = \begin{bmatrix} P_1^* \\ P_2^* \end{bmatrix}$ with $P_1^*, P_2^* \in \mathbb{C}^{l \times l}$ and $P_1^{*H} P_1^* + P_2^{*H} P_2^* = I_l$ solve (2.2), i.e.,

$$(V^*, P_1^*, P_2^*) = \arg\min \left\{ \|AVP_1 + BVP_2\|^2 : \begin{array}{l} V \in \mathbb{C}^{n \times l}, V^H V = I_l, \\ P_1, P_2 \in \mathbb{C}^{l \times l}, \\ P_1^H P_1 + P_2^H P_2 = I_l \end{array} \right\}. \tag{4.21}$$

If P_1^* is nonsingular, let

$$\begin{bmatrix} P_1^* & Q_1 \\ P_2^* & Q_2 \end{bmatrix} \quad \text{and} \quad V_0 = [V^* \;\; W]$$

be unitary matrices. If P_1 is singular, then as in page 781 of [2], we can construct $\left[\begin{smallmatrix} \Delta P_1 \\ \Delta P_2 \end{smallmatrix}\right]$ such that $P_1^* + \Delta P_1$ is nonsingular and $\|[A \;\; B]\| \|\left[\begin{smallmatrix} \Delta P_1 \\ \Delta P_2 \end{smallmatrix}\right]\| \le \epsilon$ for ϵ being small enough, then set $P_1^* := P_1^* + \Delta P_1$ and $P_2^* := P_2^* + \Delta P_2$. Moreover, since P_1^* is nonsingular, it follows from the cosine-sine decomposition that Q_2 is also nonsingular. Then define

$$\Delta_0 := Q_2^{-H}, \quad \Lambda_0 = \Delta_0 Q_1^H, \quad B_0^{(1)} = [AV^* \;\; BV^*] \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \Delta_0^{-1}$$

and

$$\hat{A} = [B_0^{(1)} \Lambda_0 \;\; AW] V_0^H, \quad \hat{B} = [B_0^{(1)} \;\; BW] V_0^H.$$

If all eigenvalues of Λ_0 are distinct and $\operatorname{rank}(\overline{B}_0) = n$, then $(\hat{A}, \hat{B})$ is the minimum perturbed pair of (A, B) that has l linearly independent eigenvectors. If all eigenvalues of Λ_0 are not distinct, according to [2], we can also compute $\Delta\Lambda_0$ such that all eigenvalues of $\Lambda_0 + \Delta\Lambda_0$ are distinct and $\|B_0^{(1)}\| \|\Delta\Lambda_0\| \le \epsilon$, and then set $\Lambda_0 := \Lambda_0 + \Delta\Lambda_0$. Besides, let the spectral decomposition of Λ_0 be

$$\Lambda_0 = \mathcal{K} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_l \end{bmatrix} \mathcal{K}^{-1},$$

and

$$[\mathbf{v}_1 \;\cdots\; \mathbf{v}_l] = V_0 \begin{bmatrix} \mathcal{K} \\ 0 \end{bmatrix}.$$

Then $\{(\lambda_k, \mathbf{v}_k), k = 1, \ldots, l\}$ are l linearly independent eigenpairs of pencil $\hat{A} - \lambda \hat{B}$.

From the above arguments, we now derive Algorithm 4.3 for solving the problem (1.3), which is to find a minimal perturbation for the pencil to have l eigenpairs with $1 \le l \le n$.

Algorithm 4.3 Algorithm of solving the problem (1.3) via Algorithm 4.2

```
Require: $A, B \in \mathbb{C}^{m \times n} (m > n)$.
  Implement Algorithm 4.2 and set the generated optimal solution of problem (2.2) by
    $(V^*, P^*)$. Partition $P^*$ by $P^* = \left[\begin{smallmatrix} P_1^* \\ P_2^* \end{smallmatrix}\right]$ with $P_1^*, P_2^* \in \mathbb{C}^{l \times l}$.
  Compute the full complex $QR$ decomposition of $\left[\begin{smallmatrix} P_1^* \\ P_2^* \end{smallmatrix}\right]$ and $V^*$ to obtain $Q_1, Q_2 \in \mathbb{C}^{l \times l}$
    and $W \in \mathbb{C}^{n \times (n-l)}$ such that $\left[\begin{smallmatrix} P_1^* & Q_1 \\ P_2^* & Q_2 \end{smallmatrix}\right]$ and $V_0 = [V^* \;\; W]$ are unitary.
  Compute $\Delta_0 = Q_2^{-H}$, $\Lambda_0 = \Delta_0 Q_1^H$ and $B_0^{(1)} = [AV^* \;\; BV^*]\left[\begin{smallmatrix} Q_1 \\ Q_2 \end{smallmatrix}\right]\Delta_0^{-1}$.
  Compute $\hat{A} = [B_0^{(1)} \Lambda_0 \;\; AW] V_0^H$ and $\hat{B} = [B_0^{(1)} \;\; BW] V_{\underline{0}}^H$.
  Compute the spectral decomposition of $\Lambda_0$ to obtain $\mathcal{K}$, $\overline{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_l)$ such that
    $\Lambda_0 = \mathcal{K}\overline{\Lambda}\mathcal{K}^{-1}$. Compute $V_0\left[\begin{smallmatrix} \mathcal{K} \\ 0_{(n-l) \times l} \end{smallmatrix}\right]$ and partition it as $V_0\left[\begin{smallmatrix} \mathcal{K} \\ 0_{(n-l) \times l} \end{smallmatrix}\right] = [\mathbf{v}_1 \;\cdots\; \mathbf{v}_l]$.
  Output $(\hat{A}, \hat{B}, \{(\lambda_k, \mathbf{v}_k)\}_{k=1}^l)$, which is an optimal solution to the problem (1.3).
```

5 Numerical Experiments

In this section, we report some numerical results to illustrate the efficiency of the proposed algorithms. Because we focus on finding an optimal pencil $(\hat{A}, \hat{B})$ in the problem (1.3) in the sense that $(\hat{A}, \hat{B})$ has l distinct eigenvalues with $1 \le l \le n$, in the first part we shall numerically compare Algorithm 4.3 with Boutry et al.'s algorithm [1] for the case $l = 1$, which is denoted by BEGM algorithm, and Ito and Murota's algorithm [5] for the case $l = n$, which is denoted by TLS-IM algorithm. In the second part we show more numerical results of Algorithm 4.2 in comparison with some other forms of Riemannian conjugate gradient methods proposed in [12–14] which are all applicable to the problem (2.2). All the numerical experiments were completed on a personal computer with a Intel(R) Core(TM)2 Quad of 2.33 GHz CPU and 3.00 GB of RAM.

5.1 Algorithmic Issues

According to Algorithm 4.2, the parameter β_{k+1} is chosen from the range $(0, \beta^D_{k+1})$. In our implementation, we choose β_{k+1} by means of

$$\beta_{k+1} = \min\{\beta^D_{k+1}, \ \beta^{FR}_{k+1}\}$$

where

$$\beta^{FR}_{k+1} = \frac{\|\phi^{-1}(\mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}))\|^2}{\|\phi^{-1}(\mathrm{grad} f(\widetilde{V}^{k}, \widetilde{P}^{k}))\|^2} = \frac{\mathrm{Tr}(\overline{\xi}^H_{k+1}\overline{\xi}_{k+1}) + \mathrm{Tr}(\overline{\eta}^H_{k+1}\overline{\eta}_{k+1})}{\mathrm{Tr}(\overline{\xi}^H_k\overline{\xi}_k) + \mathrm{Tr}(\overline{\eta}^H_k\overline{\eta}_k)}$$

is the Fletcher–Reeves parameter. From Remark 4.1, the initial steplength $\overline{\alpha}_k$ at iteration k is set to be

$$\overline{\alpha}_k = \frac{|\mathrm{Tr}(\overline{\xi}^H_k \xi_k) + \mathrm{Tr}(\overline{\eta}^H_k \eta_k)|}{\|[AV^k \ \ BV^k]\eta_k + [A\xi_k \ \ B\xi_k]P^k\|^2}.$$

Here, we choose $\overline{\alpha}_k$ by means of

$$\overline{\alpha}_k = \max\left\{\min\{\overline{\alpha}_k, \ \ \overline{\alpha}^{BB}_k, \ \ \alpha_{\max}\}, \alpha_{\min}\right\},$$

where $\overline{\alpha}^{BB}_k$ is a Riemannian generalization of the Barzilai–Borwein steplength [21], which is given by

$$\overline{\alpha}^{BB}_k = \frac{\mathrm{Tr}(S^H_{\xi,k-1}S_{\xi,k-1}) + \mathrm{Tr}(S^H_{\eta,k-1}S_{\eta,k-1})}{|\mathrm{Tr}(Z^H_{\xi,k-1}S_{\xi,k-1}) + \mathrm{Tr}(Z^H_{\eta,k-1}S_{\eta,k-1})|}$$

with $S_{\xi,k-1} = \alpha_{k-1}\xi_{k-1}$ and $S_{\eta,k-1} = \alpha_{k-1}\eta_{k-1}$ being the displacement of the variables in the tangent space and $Z_{\xi,k-1} = \overline{\xi}_k - \overline{\xi}_{k-1}$ and $Z_{\eta,k-1} = \overline{\eta}_k - \overline{\eta}_{k-1}$ being the difference of gradients. Here we do not use a vector transport for $Z_{\xi,k-1}$ and $Z_{\eta,k-1}$ but just compute them as subtraction of two gradients in different tangent spaces because using a vector transport might increase CPU time.

To achieve high efficiency in the implementation of Algorithm 4.2, the actual stopping criterion is much more sophisticated than the simple one $\|(\overline{\xi}_k, \overline{\eta}_k)\| \le \epsilon$. In our implementation we use the same stopping criterion as that in [15]: we let Algorithm 4.2 run up to K iterations and stop it at iteration $k < K$ if $\|(\overline{\xi}_k, \overline{\eta}_k)\| \le \epsilon$, or $\mathrm{rel}^k_{(V,P)} \le \epsilon_{vp}$ and $\mathrm{rel}^k_f \le \epsilon_f$,

Table 1 Summary of input parameters for Algorithm 4.2

Parameter	Value	Description
ϵ	10^{-6}	Tolerance for the norm of gradients
$\epsilon_{(V,P)}$	10^{-6}	Tolerance for the difference of variables
ϵ_f	10^{-12}	Tolerance for the difference of function values
ϵ_c	10^{-12}	Tolerance for constraint violation
δ	10^{-4}	Nonmonotone line search Armijo–Wolfe constant
h	3	Nonmonotone line search backward integer for consecutive previous function values
ρ	0.2	Steplength shrinkage factor
$\alpha_{\max}$	1	Upper threshold for steplength
$\alpha_{\min}$	10^{-20}	Lower threshold for steplength
$\overline{\alpha}_0$	10^{-3}	Initial steplength
T	5	Backward integer for consecutive previous differences of variables and function values
K	3000	Maximal number of iterations

or

$$\text{mean}\{\text{rel}_{(V,P)}^{k-\min\{k,T\}+1},\ldots,\text{rel}_{(V,P)}^{k}\}\le 10\epsilon_{vp} \text{ and } \text{mean}\{\text{rel}_{f}^{k-\min\{k,T\}+1},\ldots,\text{rel}_{f}^{k}\}\le 10\epsilon_{f}$$

for some constants $\epsilon, \epsilon_{vp}, \epsilon_f \in (0,1)$, and $T, K \in \mathbb{N}^+$, where

$$\text{rel}_{(V,P)}^{k}=\frac{\|V^{k+1}-V^k\|}{\sqrt{n}}+\frac{\|P^{k+1}-P^k\|}{\sqrt{2l}},\quad \text{rel}_{f}^{k}=\frac{|f(V^{k+1},P^{k+1})-f(V^k,P^k)|}{|f(V^k,P^k)|+1}.$$

For convenience, we summarize the default values and description for the parameters in our algorithms in Table 1. All starting points V^0 and P^0 were feasible and generated randomly by means of $V^0=\text{orth}(randn(n,l))$ and $P^0=\text{orth}(randn(2l,l))$ in Matlab style.

5.2 Comparison with the BEGM Algorithm and the TLS-IM Algorithm for Noisy Pencils

To start with our experiments, let us review roughly the following function defined by [1]

$$h(\lambda)=\sigma_{\min}(A-\lambda B)=\min_{\|v\|_2^2=1}\|(A-\lambda B)v\|_2, \tag{5.1}$$

which can help us visualize the behaviors of the nonsquare pencil $A-\lambda B$. This function is a mapping from the complex plane (values of λ) to the real and nonnegative numbers. If the matrix pair (A,B) has an eigenvalue λ_0, the value of the function at λ_0 is exactly zero. This is because, if the pencil $A-\lambda_0 B$ is rank deficient, it must have zero as its smallest singular value (and may have more than one zero singular value). For such a solvable pair the eigenvalues are the local minimum points of this function. If λ_0 is perturbed, we get a full rank pencil that leads to a strictly positive minimal singular value and, in the case of a nonsolvable pair A and B, the function is strictly positive for all λ. For a nonsquare matrix pair (A,B), if we describe the contour plots of the function $h(\lambda)=\sigma_{\min}(A-\lambda B)$ on the complex plane, the local minimum of this function could help us to estimate how many eigenvalues the nonsquare pencil $A-\lambda B$ have.

In our experiments, we generate the measured matrices $A, B \in \mathbb{C}^{300\times 5}$ $(m = 300, n = 5)$ in the problem (1.3) by the following procedure employed in [1,5]:

- Choose random matrices $\breve{A}, \breve{B} \in \mathbb{C}^{5\times 5}$, and $\breve{Q} \in \mathbb{C}^{300\times 5}$ whose entries' real and imaginary parts are drawn independently from Gaussian distribution with zero mean and standard deviation $\sigma = 1$.
- Compute QR decomposition of $\breve{Q}$, and define Q_0 to be its Q part.
- Define A_0 and B_0 by $A_0 = 50 * Q_0\breve{A}$, $B_0 = 50 * Q_0\breve{B}$. Note that the nonsquare pencil $A_0 - \lambda B_0$ has the same eigenpairs as the square matrix pencil $\breve{A} - \lambda\breve{B}$.
- Define $A = A_0 + N_A$, $B = B_0 + N_B$, where N_A, N_B are matrices of random noise.

Real and imaginary parts of N_A, N_B are drawn independently from Gaussian distribution with zero mean and standard deviation equal to σ. The parameter σ here represents the level of noise in inputs. We also used the relative errors $\mathrm{rel}_A := \frac{\|N_A\|}{\|A\|}$ and $\mathrm{rel}_B := \frac{\|N_B\|}{\|B\|}$ to measure the magnitude of the perturbation.

In our experiments, we used the matrices A_0, B_0 with the pencil $A_0 - \lambda B_0$ having five engenvalues $-0.0186 - 0.3735i$, $-0.2457 - 0.0982i$, $0.1503 + 0.7123i$, $0.9989 + 0.9233i$, $2.8651 + 1.6573i$. We prepared different noise (N_A, N_B) with noise level $\sigma = 0, 0.25, 0.50, 0.75, 1.0, 1.25$. We note that, the exact eigenvalues of noiseless pencil $A_0 - \lambda B_0$ is not necessarily equal to the optimal solution to the problem (1.1) with noisy inputs.

5.2.1 Comparison with the TLS-IM Algorithm

We first compare Algorithm 4.3 with TLS-IM algorithm [5] for the case of $l = n$, which is the case to find a minimal perturbation for the pencil $A - \lambda B$ to have n eigenpairs. We show that our proposed algorithm can achieve exactly the same eigenvalues as TLS-IM algorithm, which is essentially a direct method to find n optimal eigenpairs. We also show that both of two algorithms are robust and less affected by the data noise. To do so, for the perturbed pair (A, B) with $A = A_0 + N_A$ and $B = B_0 + N_B$, in Fig. 1, we first describe the contours of function $h(\lambda) = \sigma_{\min}(A - \lambda B)$ in the λ-complex plane with different noise $\sigma = 0, 0.25, 0.50, 0.75, 1.0, 1.25$. We can see from Fig. 1 that, with $\sigma = 0$, the function $h(\lambda) = \sigma_{\min}(A_0 - \lambda B_0)$ have five minimum points, which match exactly the location of the exact eigenvalues (represented by "∘" signs) of the unperturbed pair (A_0, B_0). As the noise level σ increases, some of the local minima points of $h(\lambda)$ vanished gradually. This implies that, with noisy inputs, the perturbed pair (A, B) may have less than five eigenvalues.

In Fig. 1, the red ×'s mark the optimal eigenvalues computed by Algorithm 4.3, and the blue +'s mark the optimal eigenvalues computed by TLS-IM algorithm. Our computational results show that, in the noiseless case, i.e., when $\sigma = 0$, both of two algorithms capture the "exact solution", i.e., eigenvalues of $A_0 - \lambda B_0$, accurately. The other figures in Fig. 1 show that, as the noise level σ increases, the discrepancy between numerical solutions and exact solution increases. However, in either case, the two algorithms can achieve exactly the same eigenvalues. On the other hand, most importantly, both of the two algorithms can always give five solutions even for the case when the measured pair (A, B) have less than five eigenvalues. This shows that both algorithms are robust and less affected by the noisy data.

We prepare 10 data sets with different noise (N_A, N_B) with $\sigma = 0, 0.25, 0.50, 0.75, 1.0, 1.25$, respectively. Figure 2 shows numerically computed eigenvalues by the two compared algorithms. Since we used 10 data sets for each σ, each figure has many points representing numerical solutions. Figure 2 shows again that, Algorithm 4.3 can achieve exactly the same eigenvalues as TLS-IM algorithm, and can capture five eigenpairs. Our computational

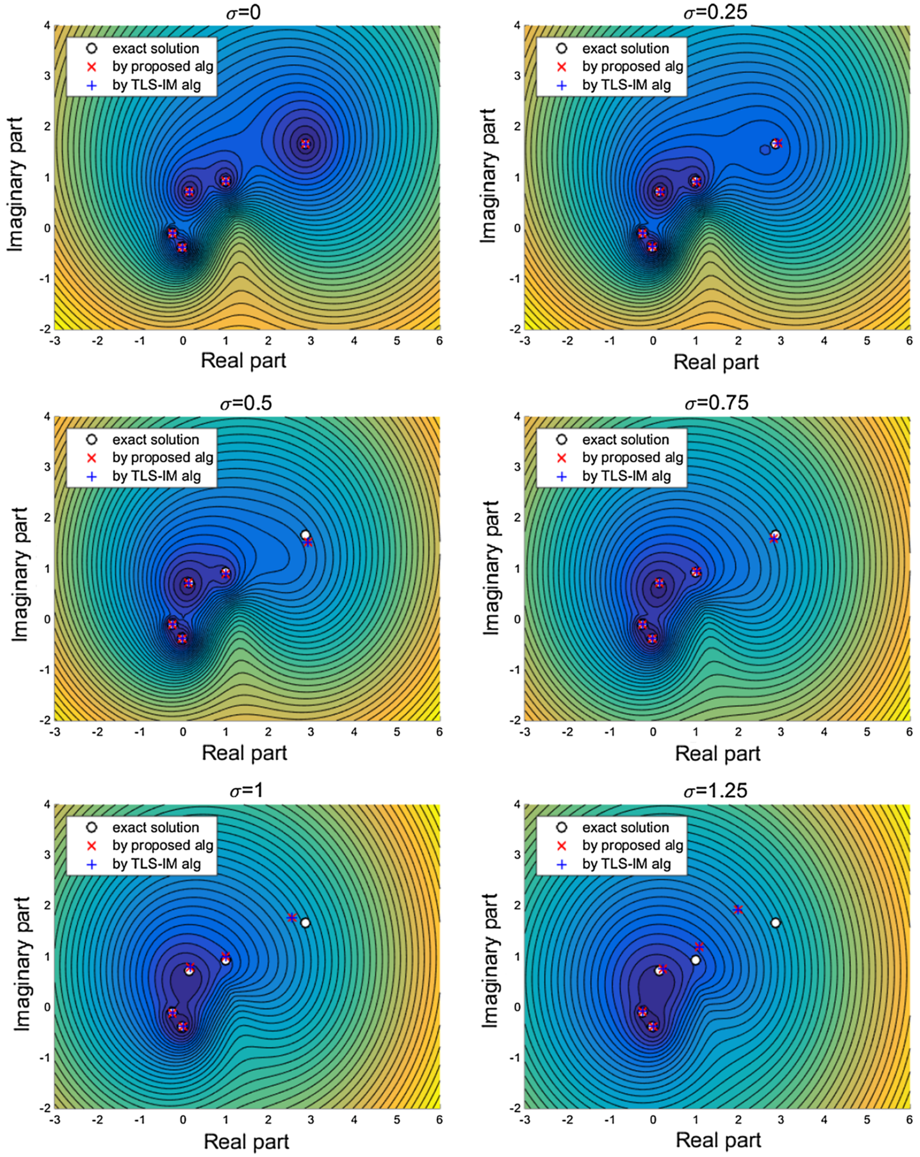

Fig. 1 Contour plots of the function $h(\lambda) = \sigma_{\min}(A - \lambda B)$, built using the perturbed pair (A, B) with $\sigma = 0, 0.25, 0.50, 0.75, 1.0, 1.25$. “Exact solution” means the eigenvalues of the noiseless pencil

results are also reported in Table 2, in which ’σ’ represents the level of noise in inputs, ’rel_A’ and ’rel_B’ denote the mean relative error of perturbation, ’IT.’, ’CT.’, ’Obj.’ and ’Res.’ denote the mean iteration numbers, the mean computational time in seconds, the mean value of $\|A - \hat{A}\|^2 + \|B - \hat{B}\|^2$ and the mean value of the residual $\|\hat{A}X - \hat{B}X\Lambda\|$ for the computed optimal solution $(\hat{A}, \hat{B}, \{(\lambda_k, \mathbf{v}_k)\}_{k=1}^{n})$ by Algorithm 4.3 with $X = [\mathbf{v_1} \ \cdots \ \mathbf{v_n}]$, $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ based on 10 repeated tests, respectively. In particular, the last column

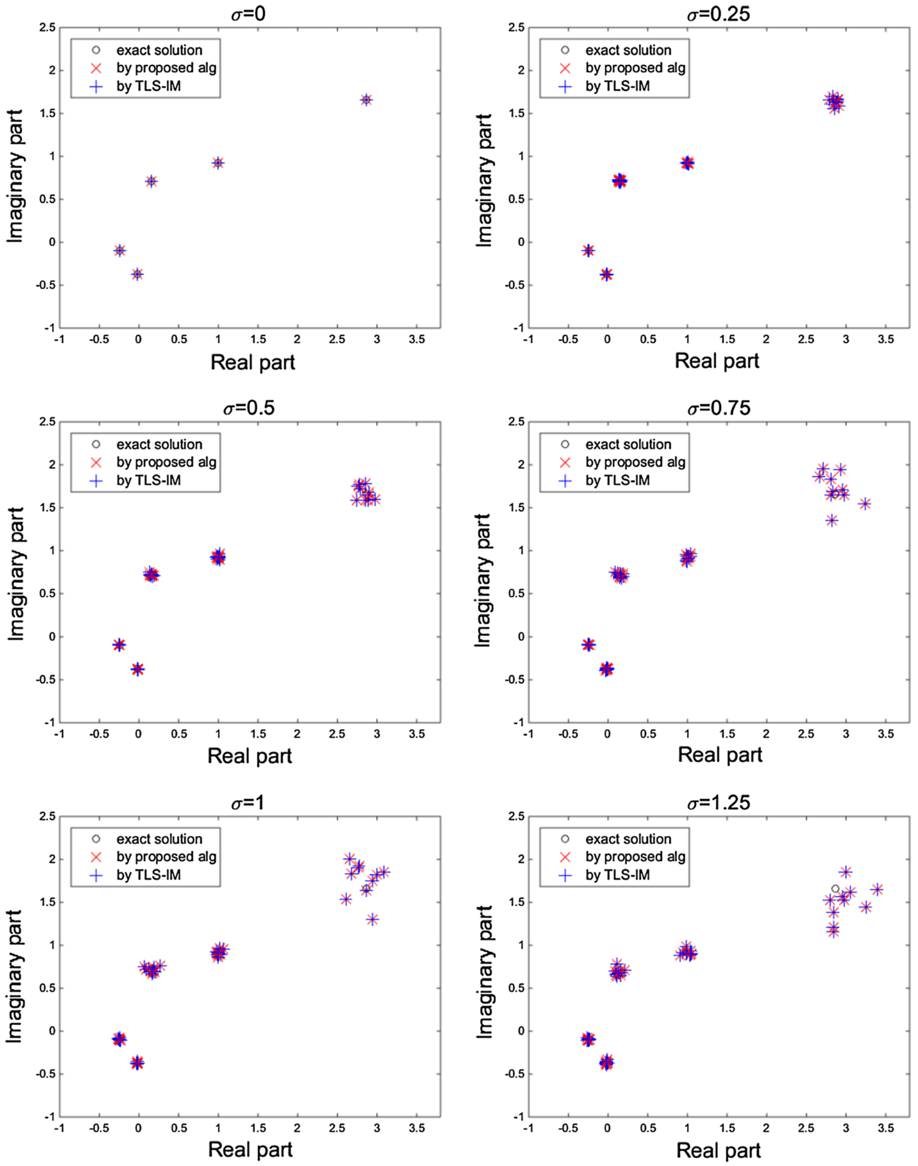

Fig. 2 Eigenvalues computed by Algorithm 4.3 and TLS-IM algorithm. "Exact solution" means the eigenvalues of the noiseless pencil. σ represents the level of noise in inputs

in Table 2 shows the mean value of the error which is defined by

$$\text{Error} := \sqrt{\|A-\hat{A}\|^2+\|B-\hat{B}\|^2}-\Big(\sum_{i=n+1}^{2n}\sigma_i^2([A\ B])\Big)^{\frac{1}{2}}.$$

Here, the singular values of $[A\ B]$ are arranged with decreasing order. Generally, it is meaningless to compare the iteration steps and computational time between two algorithms since

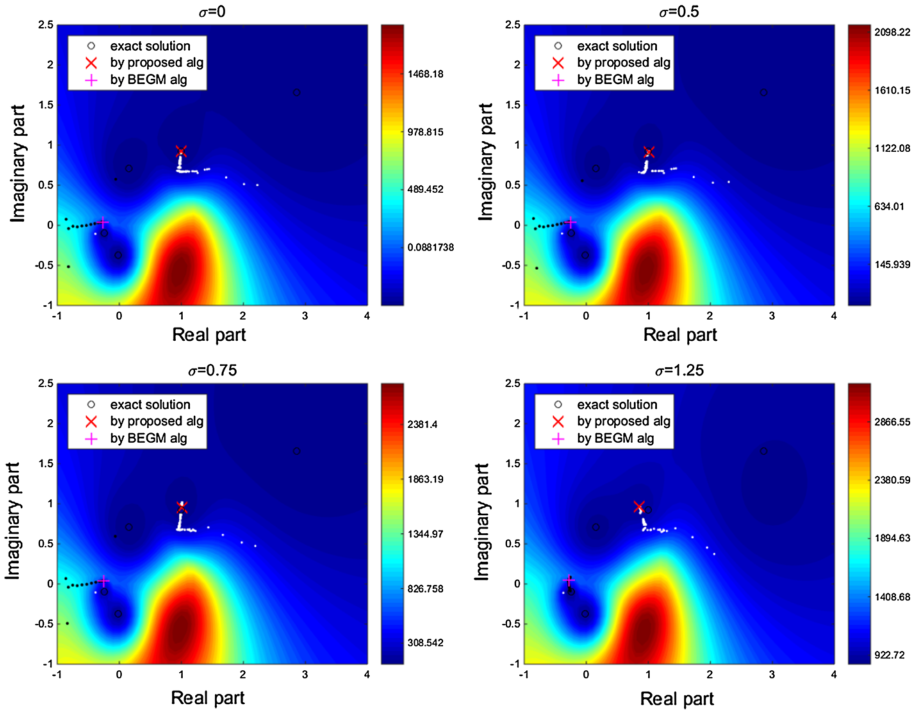

Fig. 3 Contour plots of the function $h(\lambda) = \min_{\mathbf{v}} \frac{\|(A-\lambda B)\mathbf{v}\|^2}{1+|\lambda|^2}$, built using the perturbed pair (A, B) with $\sigma = 0, 0.50, 0.75, 1.25$. "Exact solution" means the eigenvalues of the noiseless pencil

TLS-IM algorithm is essentially a direct method. We here report the iteration steps only to emphasize that our proposed algorithm is very effective for finding a minimal perturbation for the pencil to have n eigenpairs. The results reported in the last column show that the computation results are in accordance with the theory established in the Corollary 2 in [2].

5.2.2 Comparison with the BEGM Algorithm

We then compare Algorithm 4.3 with BEGM algorithm [1] for the case of $l = 1$, i.e., the problem (1.2), in the sense that the perturbed pencil admits at least one eigenpair. As pointed out in [1], the optimization problem posed in (1.2) is equivalent to the optimization problem

$$\min_{\lambda,\mathbf{v}} h(\lambda, \mathbf{v}) = \frac{\|(A - \lambda B)\mathbf{v}\|^2}{1 + |\lambda|^2} \quad \text{subject to} \quad \|\mathbf{v}\|_2^2 = 1.$$

A stable and convergent numerical algorithm is proposed in [1] which is guaranteed to converge to a local minimum of the function $h(\lambda, \mathbf{v})$. Although, with different initial values, the BEGM algorithm may result in the optimization process converging at different local minima, and therefore, estimations of different eigenvalues. However, we here focus only on the case for finding one eigenpair. The initial λ in BEGM algorithm is set as the largest eigenvalue of $B^{\dagger}A$ [1]. In Fig. 3, we describe the convergence trajectory of two algorithms on the contour plots of the function $h(\lambda) = \min_{\mathbf{v}} \frac{\|(A-\lambda B)\mathbf{v}\|^2}{1+|\lambda|^2}$(with given λ, the optimal $\mathbf{v}$ is obtained by the right singular vector corresponding to the smallest singular

Table 2 Comparison results by Algorithms 4.3 and Algorithm TLS-IM based on 10 repeated tests

σ	rel$_A$	rel$_B$	Algorithm 4.3				TLS-IM			
			IT.	Obj.	CT.	Res.	Obj.	CT.	Res.	Error
0	0	0	85.1	0	0.085	1.66×10^{-12}	0	0.007	5.92×10^{-13}	3.42×10^{-6}
0.25	3.89%	3.36%	74.5	185.163	0.080	1.89×10^{-12}	185.163	0.006	8.18×10^{-13}	6.98×10^{-11}
0.50	7.72%	6.74%	75.0	739.658	0.074	1.77×10^{-12}	739.658	0.008	1.02×10^{-12}	7.56×10^{-11}
0.75	11.66%	10.02%	66.6	1678.351	0.062	1.81×10^{-12}	1678.351	0.006	1.22×10^{-12}	2.25×10^{-10}
1.00	15.50%	13.37%	71.4	2985.597	0.063	1.68×10^{-12}	2985.597	0.006	1.07×10^{-12}	2.38×10^{-10}
1.25	19.11%	16.55%	65.3	4589.892	0.061	1.72×10^{-12}	4589.892	0.007	1.05×10^{-12}	2.33×10^{-10}

value of $(A - \lambda B)^H(A - \lambda B)$), which built using the perturbed pair (A, B) with $\sigma = 0, 0.50, 0.75, 1.25$. In Fig. 3, the white ∘'s still mark the exact eigenvalues of the clear matrix pair (A_0, B_0). The white dots and the red 'x' sign present the iterative algorithm results and the optimal eigenvalue computed by Algorithm 4.3, respectively. The black dots and the pink '+' sign present the iterative algorithm results and the optimal eigenvalue computed by BEGM algorithm, respectively. Some more computational results are also reported in Table 3, in which the mean iteration numbers, the mean computational time in seconds, the mean value of $\|A - \hat{A}\|^2 + \|B - \hat{B}\|^2$ and the mean value of $\|\hat{A}\mathbf{v} - \lambda\hat{B}\mathbf{v}\|$ for the computed optimal solution $(\hat{A}, \hat{B}, \{(\lambda, \mathbf{v})\})$ for the compared two algorithms are reported based on 10 repeated tests. We can see from Table 3 that, in most cases, Algorithm 4.3 runs faster that BEGM algorithm in terms of computational time and iteration steps.

An average of results of 10 random tests of Algorithm 4.3 for the case $1 \leq l \leq n$ is reported in Table 4, where parts of the items are the same as those of Tables 2 and 3. The items "$\text{grad}_V f$" and "$\text{grad}_P f$" represent the mean value of norm of gradient of f with respect to V and P based on 10 repeated tests, respectively. The case $l = 1$ and $l = n$ are reported again in Table 4 just for comparison only, which show that Algorithm 4.3 is relatively less efficient for the case $1 < l < n$.

5.3 Further Numerical Performance

We compared the performance of Algorithm 4.2 with the Riemannian conjugate gradient methods used in [12–14,16] which are all applicable to the problem (2.2) with necessary modifications. The known matrices $A, B \in \mathbb{C}^{m \times n}$ in (2.2) are similar to the ones given in last example with $\sigma = 0.5$ but different problem size parameters (m, n, l). The numerical comparison results are reported in Table 5, where the terms "CT.", "IT.", "$\text{grad}_V f$" and "$\text{grad}_P f$" are the same as those in Table 4, "Obj." means the objective function value $\|[AV^k \;\; BV^k]P^k\|$, "feasi_V" and "feasi_P" mean respectively the feasibility $\|V^{kH}V^k - I_l\|$ and $\|P^{kH}P^k - I_l\|$ at the final iterate by implementing the proposed algorithms. In Table 5,

- "RCG-PR" stands for the Riemannian conjugate gradient method used in [14] designed to solve the problem (2.2), whose line search is the strong Wolfe line search, and the parameter β_{k+1} in (3.1) is the Riemannian version of the Polak–Ribière-type formula by combining the vector transport proposed. The retraction adapted in [14] is the QR-based retraction, and the vector transport is constructed by using the orthogonal projection operator onto the tangent space. On the real Stiefel manifold $\text{St}(n, l, \mathbb{R})$, we often use a retraction $\mathcal{R}^{\mathbb{R}}$ based on the real QR decomposition. However, the map $\mathcal{R}^{\mathbb{R}}$ cannot be a retraction on $\text{Stp}(n, l)$ and $\text{Stp}(2l, l)$ (see [10] for more details). Alternatively, we can use a retraction $\mathcal{R}^{\text{Stp}}$ defined by

  $$\mathcal{R}^{\text{Stp}}_{\widetilde{V}}(\widetilde{\xi}) = \phi\left(qf(\phi^{-1}(\widetilde{V} + \widetilde{\xi}))\right) = \phi(qf(V + \xi)), \quad \widetilde{V} \in \text{Stp}(n, l), \quad \widetilde{\xi} \in \mathcal{T}_{\widetilde{V}}\text{Stp}(n, l),$$

  where the $qf(\cdot)$ denotes the Q-factor of the complex QR decomposition of the complex matrix in parentheses. That is, if a full-rank $n \times l$ complex matrix M is decomposed into

  $$M = QR, \quad Q \in \text{St}(n, l, \mathbb{C}), \quad R \in S^{+}_{upp},$$

  then $qf(M) = Q$, where S^{+}_{upp} denotes the set of all $l \times l$ upper triangular matrices with strictly positive diagonal entries. Therefore, a retraction $\mathcal{R}$ on the product manifold $\text{Stp}(n, l) \times \text{Stp}(2l, l)$ is immediately defined as

  $$\mathcal{R}_{(\widetilde{V}, \widetilde{P})}(\widetilde{\xi}, \widetilde{\eta}) = \left(\mathcal{R}^{\text{Stp}}_{\widetilde{V}_l}(\widetilde{\xi}), \;\; \mathcal{R}^{\text{Stp}}_{\widetilde{P}}(\widetilde{\eta})\right)$$

Table 3 Comparison results by Algorithms 4.3 and BEGM algorithm based on 10 repeated tests

σ	rel_A	rel_B	Algorithm 4.3				BEGM			
			IT.	Obj.	CT.	Res.	IT.	Obj.	CT.	Res.
0	0	0	387.9	0	0.2787	7.09×10^{-14}	67.0	408.444	0.217	2.08×10^{-14}
0.25	3.87%	3.38%	333.9	37.628	0.2326	5.73×10^{-14}	70.7	443.682	0.217	2.38×10^{-14}
0.50	7.82%	6.64%	260.5	146.010	0.1753	4.35×10^{-14}	301.4	647.288	0.956	2.28×10^{-14}
0.75	11.65%	10.05%	262.6	335.360	0.1843	5.49×10^{-14}	484.3	769.888	1.534	2.85×10^{-14}
1.00	15.53%	13.32%	310.0	585.073	0.2118	6.65×10^{-14}	654.6	1054.084	2.060	2.16×10^{-14}
1.25	19.14%	16.47%	200.3	885.991	0.1387	4.69×10^{-14}	1244.6	1898.528	3.930	3.33×10^{-14}

Table 4 Numerical results for Algorithms 4.3 based on 10 repeated tests

	σ	rel_A	rel_B	Algorithm 4.3					
				IT.	Obj.	CT.	$\text{grad}_V f$	$\text{grad}_P f$	Res.
$l = 1$	0	0	0	387.9	0.00	0.28	2.85×10^{-3}	1.33×10^{-3}	5.99×10^{-14}
	0.25	3.87%	3.38%	333.9	37.63	0.23	5.79×10^{-3}	2.53×10^{-3}	4.52×10^{-14}
	0.50	7.82%	6.64%	260.5	146.01	0.18	7.43×10^{-3}	3.11×10^{-3}	6.53×10^{-14}
	0.75	11.65%	10.05%	262.6	335.36	0.17	1.12×10^{-2}	4.61×10^{-3}	5.74×10^{-14}
	1.00	15.53%	13.32%	310.0	585.07	0.20	2.64×10^{-2}	1.47×10^{-2}	4.95×10^{-14}
	1.25	19.14%	16.47%	200.3	885.99	0.13	2.19×10^{-2}	5.55×10^{-3}	5.30×10^{-14}
$l = 2$	0	0	0	636.1	0.00	0.45	1.06×10^{-3}	7.30×10^{-4}	6.80×10^{-13}
	0.25	3.89%	3.35%	496.1	73.71	0.35	8.59×10^{-3}	7.00×10^{-3}	7.62×10^{-13}
	0.50	7.77%	6.72%	551.2	295.57	0.39	2.30×10^{-2}	1.90×10^{-2}	5.85×10^{-13}
	0.75	11.68%	10.02%	468.4	671.32	0.33	2.56×10^{-2}	2.30×10^{-2}	1.06×10^{-12}
	1.00	15.49%	13.38%	486.8	1190.03	0.34	4.54×10^{-2}	3.88×10^{-2}	2.17×10^{-12}
	1.25	19.06%	16.45%	580.3	1821.17	0.41	3.91×10^{-2}	3.00×10^{-2}	4.76×10^{-12}
$l = 3$	0	0	0	683.9	0.00	0.55	2.95×10^{-3}	2.24×10^{-3}	2.80×10^{-12}
	0.25	3.88%	3.35%	696.3	110.36	0.56	9.93×10^{-3}	7.58×10^{-3}	2.21×10^{-12}
	0.50	7.79%	6.67%	725.3	444.61	0.59	3.85×10^{-2}	2.88×10^{-2}	2.77×10^{-12}
	0.75	11.66%	10.05%	582.9	1000.99	0.48	3.70×10^{-2}	2.75×10^{-2}	2.01×10^{-12}
	1.00	15.43%	13.24%	589.9	1796.88	0.48	4.55×10^{-2}	3.63×10^{-2}	2.06×10^{-12}
	1.25	19.33%	16.49%	463.8	2802.96	0.38	4.32×10^{-2}	3.14×10^{-2}	1.60×10^{-12}

Table 4 continued

	σ	rel_A	rel_B	Algorithm 4.3					
				IT.	Obj.	CT.	$\text{grad}_V f$	$\text{grad}_P f$	Res.
$l = 4$	0	0	0	812.1	0.00	0.79	2.03×10^{-3}	1.97×10^{-3}	5.89×10^{-12}
	0.25	3.90%	3.34%	533.9	146.57	0.53	1.54×10^{-2}	1.28×10^{-2}	4.44×10^{-12}
	0.50	7.76%	6.67%	590.7	596.75	0.58	3.84×10^{-2}	3.41×10^{-2}	3.57×10^{-12}
	0.75	11.60%	9.99%	488.7	1332.42	0.47	5.03×10^{-2}	4.61×10^{-2}	3.20×10^{-12}
	1.00	15.39%	13.22%	542.1	2359.72	0.53	4.28×10^{-2}	4.16×10^{-2}	4.67×10^{-12}
	1.25	19.14%	16.46%	387.3	3672.97	0.38	3.73×10^{-2}	3.27×10^{-2}	2.51×10^{-12}
$l = 5$	0	0	0	83.8	0.00	0.08	5.76×10^{-4}	8.44×10^{-4}	1.74×10^{-12}
	0.25	3.90%	3.35%	74.6	186.63	0.07	3.88×10^{-3}	5.80×10^{-3}	2.09×10^{-12}
	0.50	7.80%	6.66%	67.4	738.68	0.06	5.52×10^{-3}	9.63×10^{-3}	2.02×10^{-12}
	0.75	11.65%	9.96%	71.5	1663.94	0.06	1.36×10^{-2}	1.96×10^{-2}	2.02×10^{-12}
	1.00	15.39%	13.32%	72.3	2956.09	0.07	2.25×10^{-2}	3.37×10^{-2}	1.82×10^{-12}
	1.25	19.18%	16.59%	66.2	4640.95	0.06	2.50×10^{-2}	3.64×10^{-2}	2.01×10^{-12}

Table 5 Numerical results for Algorithms 4.2 based on 10 repeated tests

m, n, l	Algorithm	CT.	IT.	Obj.	$\text{grad}_V f$	$\text{grad}_P f$	feasi_V	feasi_P
30,15,1	Algorithm 4.2	0.045	137	2.938	9.00×10^{-4}	1.34×10^{-4}	3.33×10^{-16}	1.55×10^{-15}
	RCG-PR	0.109	267	2.938	2.41×10^{-4}	1.05×10^{-5}	0	0
	RCG-MPRP	0.141	383	2.938	2.70×10^{-4}	7.03×10^{-5}	3.33×10^{-16}	1.11×10^{-16}
	RCG-FR	0.235	652	2.938	5.05×10^{-4}	1.92×10^{-4}	2.22×10^{-16}	2.22×10^{-16}
	RCGQR	0.055	143	2.938	3.08×10^{-4}	6.61×10^{-5}	1.11×10^{-16}	2.22×10^{-16}
	RCGPol	0.055	145	2.938	1.73×10^{-4}	4.49×10^{-5}	6.66×10^{-15}	8.88×10^{-16}
30,15,5	Algorithm 4.2	0.119	322	16.147	2.77×10^{-4}	8.98×10^{-5}	1.59×10^{-14}	1.69×10^{-14}
	RCG-PR	1.354	2323	19.431	5.82×10^{-4}	3.02×10^{-4}	9.41×10^{-16}	6.02×10^{-16}
	RCG-MPRP	0.786	1693	14.367	1.76×10^{-3}	1.03×10^{-3}	8.48×10^{-16}	9.97×10^{-16}
	RCG-FR	1.881	3000	1.69×10^{4}	3.01×10^{4}	7.70×10^{3}	8.27×10^{-16}	9.06×10^{-16}
	RCGQR	0.229	358	20.409	1.24×10^{-3}	8.27×10^{-4}	1.19×10^{-15}	9.09×10^{-16}
	RCGPol	2.996	3000	4248.3	1.05×10^{1}	4.96	5.43×10^{-1}	1.45×10^{-12}
30,15,15	Algorithm 4.2	0.065	83	54.105	9.40×10^{-4}	6.53×10^{-4}	1.41×10^{-14}	1.69×10^{-14}
	RCG-PR	0.186	235	54.105	1.72×10^{-3}	2.34×10^{-3}	2.19×10^{-15}	2.01×10^{-15}
	RCG-MPRP	0.117	143	54.105	6.55×10^{-4}	1.68×10^{-3}	1.83×10^{-15}	1.66×10^{-15}
	RCG-FR	0.118	176	54.105	8.82×10^{-4}	1.70×10^{-3}	2.14×10^{-15}	2.04×10^{-15}
	RCGQR	0.096	87	54.105	4.87×10^{-4}	7.10×10^{-4}	2.85×10^{-15}	1.79×10^{-15}
	RCGPol	5.019	3000	2.33×10^{4}	2.95×10^{1}	2.30×10^{1}	1.73	1.72×10^{-10}
300,15,1	Algorithm 4.2	0.057	146	67.665	2.60×10^{-4}	4.73×10^{-5}	4.88×10^{-15}	2.22×10^{-16}
	RCG-PR	0.369	705	71.075	1.07×10^{-3}	1.80×10^{-4}	2.22×10^{-16}	3.26×10^{-16}
	RCG-MPRP	0.444	910	71.075	1.33×10^{-3}	5.18×10^{-4}	4.44×10^{-16}	3.25×10^{-16}
	RCG-FR	0.292	598	71.075	1.32×10^{-3}	3.87×10^{-4}	2.22×10^{-16}	2.22×10^{-16}
	RCGQR	0.132	225	71.075	7.43×10^{-4}	3.23×10^{-4}	6.66×10^{-16}	0
	RCGPol	0.104	197	71.075	1.22×10^{-3}	6.07×10^{-5}	4.44×10^{-16}	0

Table 5 continued

m, n, l	Algorithm	CT.	IT.	Obj.	$\text{grad}_V f$	$\text{grad}_P f$	feasi_V	feasi_P
300,15,5	Algorithm 4.2	0.237	446	341.628	2.18×10^{-3}	5.05×10^{-4}	2.72×10^{-14}	1.75×10^{-14}
	RCG-PR	2.018	2163	339.253	6.37×10^{-3}	3.83×10^{-3}	6.28×10^{-16}	7.83×10^{-16}
	RCG-MPRP	1.705	2167	354.497	5.76×10^{-3}	2.94×10^{-3}	5.05×10^{-16}	8.98×10^{-16}
	RCG-FR	3.370	3000	8.39×10^{5}	9.32×10^{4}	5.84×10^{4}	7.51×10^{-16}	9.29×10^{-16}
	RCGQR	0.559	526	355.603	1.59×10^{-3}	1.15×10^{-3}	8.78×10^{-16}	6.97×10^{-16}
	RCGPol	3.775	3000	1.12×10^{5}	6.61×10^{-2}	3.82×10^{-2}	2.00	6.73×10^{-10}
300,15,15	Algorithm 4.2	0.087	69	1059.542	2.70×10^{-4}	2.66×10^{-4}	1.51×10^{-14}	1.33×10^{-14}
	RCG-PR	0.493	210	1059.542	7.84×10^{-3}	1.07×10^{-2}	2.10×10^{-15}	2.43×10^{-15}
	RCG-MPRP	0.417	170	1059.542	1.18×10^{-3}	3.16×10^{-3}	2.35×10^{-15}	1.81×10^{-15}
	RCG-FR	10.467	3000	2.58×10^{5}	1.12×10^{5}	2.33×10^{5}	2.14×10^{-15}	2.00×10^{-15}
	RCGQR	0.317	120	1059.542	3.64×10^{-3}	5.95×10^{-3}	1.77×10^{-15}	2.12×10^{-15}
	RCGPol	12.756	3000	2.24×10^{5}	1.36×10^{4}	1.67×10^{3}	2.89	2.74×10^{-9}
1000,30,1	Algorithm 4.2	0.359	543	246.944	1.45×10^{-3}	1.71×10^{-4}	5.55×10^{-15}	2.66×10^{-15}
	RCG-PR	1.040	1208	237.425	3.01×10^{-3}	7.23×10^{-4}	2.22×10^{-16}	4.44×10^{-16}
	RCG-MPRP	1.812	2097	237.425	7.62×10^{-3}	1.49×10^{-3}	4.44×10^{-16}	2.22×10^{-16}
	RCG-FR	2.542	3000	1788.785	9.48×10^{3}	4.61×10^{1}	2.22×10^{-16}	2.22×10^{-16}
	RCGQR	0.820	976	265.148	5.93×10^{-3}	1.37×10^{-3}	2.22×10^{-16}	0
	RCGPol	0.607	699	314.849	6.41×10^{-11}	5.95×10^{-4}	0	2.22×10^{-16}

Table 5 continued

m, n, l	Algorithm	CT.	IT.	Obj.	$\text{grad}_V f$	$\text{grad}_P f$	feasi_V	feasi_P
1000,30,5	Algorithm 4.2	2.272	1719	1255.312	2.65×10^{-2}	1.38×10^{-2}	6.07×10^{-14}	1.08×10^{-13}
	RCG-PR	7.153	3000	1234.414	7.07×10^{1}	4.39×10^{1}	7.99×10^{-16}	6.42×10^{-16}
	RCG-MPRP	9.110	3000	1240.582	7.16×10^{1}	2.85×10^{1}	1.50×10^{-15}	1.11×10^{-15}
	RCG-FR	11.768	3000	2.07×10^{5}	2.41×10^{5}	1.05×10^{5}	5.73×10^{-16}	1.77×10^{-15}
	RCGQR	3.953	1358	1202.042	2.25×10^{-2}	1.17×10^{-2}	1.02×10^{-15}	7.58×10^{-16}
	RCGPol	11.065	3000	2.15×10^{5}	4.23×10^{3}	2.10×10^{3}	2.11	1.50×10^{-7}
1000,30,30	Algorithm 4.2	0.496	91	7317.251	7.23×10^{-2}	5.78×10^{-2}	3.14×10^{-14}	2.69×10^{-14}
	RCG-PR	3.994	334	7317.251	3.43×10^{-2}	4.68×10^{-2}	3.95×10^{-15}	3.43×10^{-15}
	RCG-MPRP	3.286	264	7317.251	1.20×10^{-2}	2.37×10^{-2}	3.51×10^{-15}	3.66×10^{-15}
	RCG-FR	54.356	3000	1.61×10^{6}	3.86×10^{5}	7.99×10^{5}	4.43×10^{-15}	3.33×10^{-15}
	RCGQR	2.694	199	7317.251	2.27×10^{-2}	3.43×10^{-2}	3.35×10^{-15}	3.51×10^{-15}
	RCGPol	58.507	3000	1.49×10^{6}	6.34×10^{4}	8.31×10^{3}	5.06	6.17×10^{-8}

where $(\widetilde{V}_l, \widetilde{P}) \in \mathrm{Stp}(n,l) \times \mathrm{Stp}(2l,l)$ and $(\widetilde{\xi}, \widetilde{\eta}) \in \mathcal{T}_{(\widetilde{V}_l,\widetilde{P})}(\mathrm{Stp}(n,l) \times \mathrm{Stp}(2l,l))$. Further, the retraction on $\mathrm{Stp}(n,l) \times \mathrm{Stp}(2l,l)$ corresponds to the retraction $\mathcal{R}$ on $\mathrm{St}(n,l,\mathbb{C}) \times \mathrm{St}(2l,l,\mathbb{C})$ defined by

$$\mathcal{R}_{(V,P)}(\alpha(\xi,\eta)) = (\mathcal{R}_V^{\mathbb{C}}(\alpha\xi),\ \ \mathcal{R}_P^{\mathbb{C}}(\alpha\eta)) = (qf(V+\alpha\xi),\ \ qf(P+\alpha\eta)), \quad (5.2)$$

for any $\alpha \in \mathbb{R}$, where $(V,P) \in \mathrm{St}(n,l,\mathbb{C}) \times \mathrm{St}(2l,l,\mathbb{C})$ and $(\xi,\eta) \in \mathcal{T}_{(V,P)}\mathrm{St}(n,l,\mathbb{C}) \times \mathrm{St}(2l,l,\mathbb{C})$. The vector transport used in [14] is defined by

$$\begin{aligned}\mathcal{T}_{\alpha(\xi,\eta)}(\zeta,\chi) &= \mathbf{P}_{\mathcal{R}_{(V_l,P)}(\alpha(\xi,\eta))}(\zeta,\chi)\\ &= \Big(\zeta - qf(V+\alpha\xi)\mathrm{her}(qf(V+\alpha\xi)^H\zeta),\ \ \chi - qf(P+\alpha\eta)\mathrm{her}(qf(P+\alpha\eta)^H\chi)\Big)\end{aligned} \quad (5.3)$$

where $(\xi,\eta), (\zeta,\chi) \in \mathbf{T}_{(V_l,P)}\mathrm{St}(n,l,\mathbb{C}) \times \mathrm{St}(2l,l,\mathbb{C})$.

- "RCG-MPRP" stands for the Riemannian geometric modified Polak–Ribière–Polay conjugate gradient method used in [12] designed to solve to the problem (2.2), with the same QR-based retraction and vector transport as "RCG-PR".
- "RCG-FR" stands for the Riemannian Fletcher–Reeves conjugate gradient method used in [13] designed to solve the problem (2.2), with the same QR-based retraction and vector transport as "RCG-PR".
- "RCGQR" stands for a modification of Algorithm 4.2, by using the QR-based retraction (5.3) and the corresponding differentiated retraction as a vector transport (see p. 173 of [11]).
- "RCGPol" stands for another modification of Algorithm 4.2, by the polar decomposition to construct a retraction (see p. 59 of [11]), and naturally using the corresponding differentiated retraction as a vector transport (see p. 173 of [11]).

We see from Table 5 that, all the compared algorithms work very efficient for solving the problem (2.2) in the case $l = 1$, while the performance of Algorithm 4.2 is slightly better than that of the other algorithms in term of computational time and iteration steps. In the case $1 < l \leq n$ Algorithm 4.2 perform much better than others. In addition, we present in Fig. 4 the plots of optimality (quadratic sum of norm of the gradient) versus iteration of one particular test to compare further the efficiency of these algorithms. We see that Algorithm 4.2 outperformed the other five competitors in general.

6 Concluding Remarks

In this paper, we have considered the generalized eigenvalue problem for nonsquare pencils, by finding the minimal perturbation to the pencil such that the perturbed pencil has l distinct eigenpairs with $1 \leq l \leq n$, which was first shown in theoretical study by Chu et al. [2] that can be reduced to a special form of optimization problem (1.4). We have focused on solving the general optimization problem (1.4) from numerical point of view, and proposed an efficient algorithm based on Riemannian nonlinear conjugate gradient method with nonmonotone line search technique. It is confirmed numerically that the proposed algorithm is effective for any $1 \leq l \leq n$. In particular, for the case $l = n$, the algorithm is less affected by the data noise and guaranteed to yield n eigenpairs, and the algorithm generated exactly the same optimal solution as Ito and Murota's algorithm [5], whose approach is essentially a direct method for designing for solving the problem (1.1), and was proposed by Boutry et al. in [1].

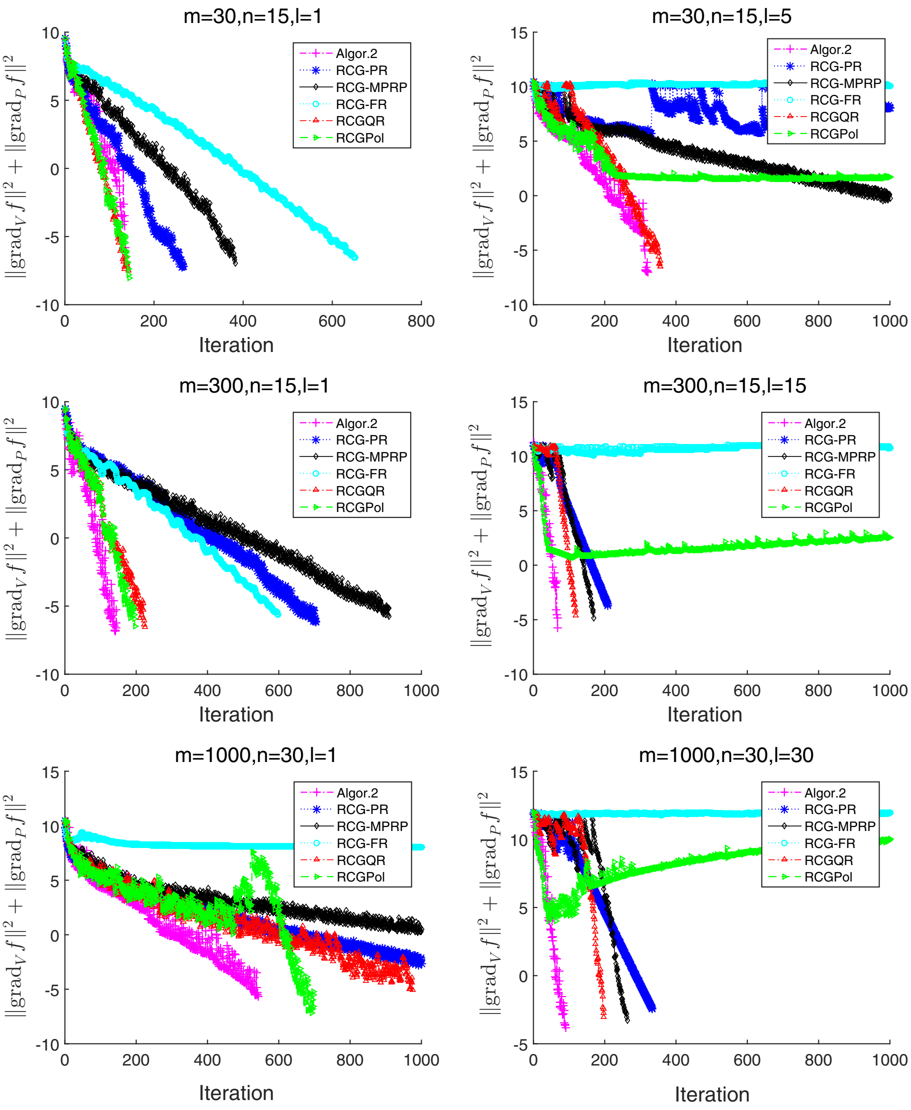

Fig. 4 Evolutions of norm of the gradients versus iterations for different problem sizes

Moreover, the algorithm runs faster than Boutry et al.'s algorithm for the case with $l = 1$, in the sense that the perturbed pair with minimal perturbation admits at least one engenpair.

Acknowledgements The authors would like to thank both reviewers for their very constructive suggestions that lead to the improvement of this manuscript.

7 Appendix

7.1 Proof of Lemma 4.1

Proof For any point $(\widetilde{V}, \widetilde{P}) \in \Phi$, we have $\|\widetilde{V}\| = \|\widetilde{P}\| = \sqrt{l}$, which implies that the level set Φ is bounded. This together with the fact that the objection function $f(\widetilde{V}, \widetilde{P})$ is a continuous function with respect to $\widetilde{V}$ and $\widetilde{P}$, shows that Φ is also closed. Thus the level set Φ is compact.

□

7.2 Proof of Lemma 4.2

Proof We proceed by induction. Since $(\widetilde{\xi}_0, \widetilde{\eta}_0) = -\mathrm{grad} f(\widetilde{V}^0, \widetilde{P}^0)$, (35) holds immediately for $k = 0$. Suppose (4.11) holds for some k. Then $\beta^D_{k+1} > 0$ according to (32) and therefore β_{k+1} is well defined and the ratio $r_{k+1} = \frac{\beta_{k+1}}{\beta^D_{k+1}}$ lies in [0, 1]. By (4.2), (4.3) and (4.4), we have

$$\begin{aligned}
&\langle \mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ (\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1})\rangle \\
&\quad = r_{k+1}\beta^D_{k+1}\left\langle \mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ \mathcal{T}_{\alpha_k(\widetilde{\xi}_k,\widetilde{\eta}_k)}(\widetilde{\xi}_k, \widetilde{\eta}_k)\right\rangle - \|\mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1})\|^2 \\
&\quad = \frac{r_{k+1}\langle \mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ \mathcal{T}_{\alpha_k(\widetilde{\xi}_k,\widetilde{\eta}_k)}(\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle - \max\{Y_{k+1},\ -\langle \mathrm{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle\}}{\max\{Y_{k+1},\ -\langle \mathrm{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle\}} \\
&\qquad \cdot \|\mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1})\|^2. \qquad (7.1)
\end{aligned}$$

If $\langle \mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ \mathcal{T}_{\alpha_k(\widetilde{\xi}_k,\widetilde{\eta}_k)}(\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle \geq 0$, it follows from the induction hypothesis and (7.1) that $Y_{k+1} > 0$ and

$$\begin{aligned}
&\langle \mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ (\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1})\rangle \\
&\quad = \frac{(r_{k+1} - 1)\langle \mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ \mathcal{T}_{\alpha_k(\widetilde{\xi}_k,\widetilde{\eta}_k)}(\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle + \langle \mathrm{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle}{Y_{k+1}} \\
&\qquad \|\mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1})\|^2,
\end{aligned}$$

which show that $\langle \mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ (\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1})\rangle < 0$.

If $\langle \mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ \mathcal{T}_{\alpha_k(\widetilde{\xi}_k,\widetilde{\eta}_k)}(\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle < 0$, it follows from (7.1) that

$$\begin{aligned}
&\langle \mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ (\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1})\rangle \\
&\quad = -\frac{\langle \mathrm{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle + r_{k+1}\langle \mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ \mathcal{T}_{\alpha_k(\widetilde{\xi}_k,\widetilde{\eta}_k)}(\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle}{\langle \mathrm{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle} \\
&\qquad \|\mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1})\|^2,
\end{aligned}$$

which also gives $\langle \mathrm{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ (\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1})\rangle < 0$. Therefore (4.11) holds for $k + 1$ in both cases. By induction, we conclude that (4.11) is true for all k. □

7.3 Proof of Lemma 4.3

Proof It follows from the nonmonotone linesearch step, Step 2, of Algorithm 4.1, that when k is sufficiently large, $\rho^{-1}\alpha_k$ does not satisfy (4.1). This means,

$$\begin{aligned}
& f(R_{(\widetilde{V}^k,\widetilde{P}^k)}(\rho^{-1}\alpha_k(\widetilde{\xi}_k,\widetilde{\eta}_k))) \\
& \quad > \max\{f(\widetilde{V}^k,\widetilde{P}^k),\ldots,f(\widetilde{V}^{k-h(k)},\widetilde{P}^{k-h(k)})\} + \delta\rho^{-1}\alpha_k\langle \mathrm{grad} f(\widetilde{V}^k,\widetilde{P}^k),(\widetilde{\xi}_k,\widetilde{\eta}_k)\rangle \\
& \quad \geq f(\widetilde{V}^k,\widetilde{P}^k) + \delta\rho^{-1}\alpha_k\langle \mathrm{grad} f(\widetilde{V}^k,\widetilde{P}^k),(\widetilde{\xi}_k,\widetilde{\eta}_k)\rangle. \qquad (7.2)
\end{aligned}$$

We note that the pullback function $\widehat{f}(\widetilde{\xi},\widetilde{\eta})$ defined in (3.26) is continuously differentiable since both the cost function $f(\widetilde{V},\widetilde{P})$ and the retraction mapping $\mathcal{R}_{(\widetilde{V},\widetilde{P})}$ defined in (3.8) are continuously differentiable. Then there exist two constants $\kappa > 0$ and $L > 0$ such that

$$\|\mathrm{grad}\widehat{f}_{(\widetilde{V},\widetilde{P})}(\widetilde{\xi},\widetilde{\eta}) - \mathrm{grad}\widehat{f}_{(\widetilde{V},\widetilde{P})}(\widetilde{\zeta},\widetilde{\chi})\| \leq L\|(\widetilde{\xi},\widetilde{\eta}) - (\widetilde{\zeta},\widetilde{\chi})\| \qquad (7.3)$$

for any $(\widetilde{V},\widetilde{P}) \in \Phi$ and $(\widetilde{\xi},\widetilde{\eta}),(\widetilde{\zeta},\widetilde{\chi}) \in \mathbf{T}_{(\widetilde{V},\widetilde{P})}\mathrm{Stp}(n,l)\times\mathrm{Stp}(2l,l)$ with $\|(\widetilde{\xi},\widetilde{\eta})\|$, $\|(\widetilde{\zeta},\widetilde{\chi})\| \leq \kappa$.

By applying the mean-value theorem, together with (7.3), there exists a constant $\omega_k \in (0,1)$ such that for all k sufficiently large,

$$\begin{aligned}
& f\left(R_{(\widetilde{V}^k,\widetilde{P}^k)}(\rho^{-1}\alpha_k(\widetilde{\xi}_k,\widetilde{\eta}_k))\right) - f(\widetilde{V}^k,\widetilde{P}^k) \\
& = f\left(R_{(\widetilde{V}^k,\widetilde{P}^k)}(\rho^{-1}\alpha_k(\widetilde{\xi}_k,\widetilde{\eta}_k))\right) - f\left(R_{(\widetilde{V}^k,\widetilde{P}^k)}(0_{(\widetilde{V}^k,\widetilde{P}^k)})\right) \\
& = \widehat{f}_{R_{(\widetilde{V}^k,\widetilde{P}^k)}}(\rho^{-1}\alpha_k(\widetilde{\xi}_k,\widetilde{\eta}_k)) - \widehat{f}_{R_{(\widetilde{V}^k,\widetilde{P}^k)}}\left(0_{(\widetilde{V}^k,\widetilde{P}^k)}\right) \\
& = \rho^{-1}\alpha_k\left\langle \mathrm{grad}\widehat{f}_{R_{(\widetilde{V}^k,\widetilde{P}^k)}}\left(\omega_k\rho^{-1}\alpha_k(\widetilde{\xi}_k,\widetilde{\eta}_k)\right),\ (\widetilde{\xi}_k,\widetilde{\eta}_k)\right\rangle \\
& = \rho^{-1}\alpha_k\left\langle \mathrm{grad}\widehat{f}_{R_{(\widetilde{V}^k,\widetilde{P}^k)}}(0_{(\widetilde{V}^k,\widetilde{P}^k)}),\ (\widetilde{\xi}_k,\widetilde{\eta}_k)\right\rangle - \rho^{-1}\alpha_k\left\langle(\widetilde{\xi}_k,\widetilde{\eta}_k),\ \mathrm{grad}\widehat{f}_{R_{(\widetilde{V}^k,\widetilde{P}^k)}}(0_{(\widetilde{V}^k,\widetilde{P}^k)})\right\rangle \\
& \quad + \rho^{-1}\alpha_k\left\langle(\widetilde{\xi}_k,\widetilde{\eta}_k),\ \mathrm{grad}\widehat{f}_{R_{(\widetilde{V}^k,\widetilde{P}^k)}}\left(\omega_k\rho^{-1}\alpha_k(\widetilde{\xi}_k,\widetilde{\eta}_k)\right)\right\rangle \\
& \leq \rho^{-1}\alpha_k\left\langle \mathrm{grad}\widehat{f}_{R_{(\widetilde{V}^k,\widetilde{P}^k)}}(0_{(\widetilde{V}^k,\widetilde{P}^k)}),\ (\widetilde{\xi}_k,\widetilde{\eta}_k)\right\rangle + \omega_k L\rho^{-2}\alpha_k^2\|(\widetilde{\xi}_k,\widetilde{\eta}_k)\|^2 \\
& \leq \rho^{-1}\alpha_k\left\langle \mathrm{grad} f(\widetilde{V}^k,\widetilde{P}^k),\ (\widetilde{\xi}_k,\widetilde{\eta}_k)\right\rangle + L\rho^{-2}\alpha_k^2\|(\widetilde{\xi}_k,\widetilde{\eta}_k)\|^2. \qquad (7.4)
\end{aligned}$$

Combining (7.2) with (7.4) together gives for all k sufficiently large,

$$\alpha_k > \frac{(\delta-1)\rho\langle \mathrm{grad} f(\widetilde{V}^k,\widetilde{P}^k),\ (\widetilde{\xi}_k,\widetilde{\eta}_k)\rangle}{L\|(\widetilde{\xi}_k,\widetilde{\eta}_k)\|^2}.$$

Then we get (4.12) by setting $\mu = (1-\delta)\rho/L$.

The proof of the second part is the same as that of Theorem 3.2 in [7]. To make this lemma self-contained, we prove it as follows.

For $j \geq 0$, define

$$F_j = \max\{f(\widetilde{V}^{hj},\widetilde{P}^{hj}), f(\widetilde{V}^{hj+1},\widetilde{P}^{hj+1})\ldots, f(\widetilde{V}^{hj+h-1},\widetilde{P}^{hj+h-1})\}.$$

We show by induction that

$$f(\widetilde{V}^{hj+i-1},\widetilde{P}^{hj+i-1}) \leq F_{j-1} + \delta\alpha_{hj+i-2}\langle \mathrm{grad} f(\widetilde{V}^{hj+i-2},\widetilde{P}^{hj+i-2}),\ (\widetilde{\xi}_{hj+i-2},\widetilde{\eta}_{hj+i-2})\rangle \qquad (7.5)$$

for all $i = 1,\ldots,h$. In fact, due to (4.1), (7.5) holds for $i = 1$. Assume (7.5) holds for all $i = 1,\ldots,t$ for some $1 \leq t \leq h-1$. Then it follows from (4.11) that

$$f(\widetilde{V}^{hj+i-2},\widetilde{P}^{hj+i-2}) \leq F_{j-1}, \quad \text{for all } i = 1,\ldots,t.$$

This, together with (4.1), implies that

$$
\begin{aligned}
& f(\widetilde{V}^{hj+t}, \widetilde{P}^{hj+t}) \\
& \quad \le \max\{f(\widetilde{V}^{hj+t-1}, \widetilde{P}^{hj+t-1}), f(\widetilde{V}^{hj+t-2}, \widetilde{P}^{hj+t-2}), \ldots, f(\widetilde{V}^{hj+t-h}, \widetilde{P}^{hj+t-h})\} \\
& \qquad + \delta\alpha_{hj+t-1}\langle \operatorname{grad} f(\widetilde{V}^{hj+t-1}, \widetilde{P}^{hj+t-1}),\ (\widetilde{\xi}_{hj+t-1}, \widetilde{\eta}_{hj+t-1})\rangle \\
& \quad \le F_{j-1} + \delta\alpha_{hj+t-1}\langle \operatorname{grad} f(\widetilde{V}^{hj+t-1}, \widetilde{P}^{hj+t-1}),\ (\widetilde{\xi}_{hj+t-1}, \widetilde{\eta}_{hj+t-1})\rangle.
\end{aligned}
$$

Thus, (7.5) is true for $t+1$. Therefore, (7.5) is true for $i = 1, \ldots, h$. By (7.5) and the definition of F_j, we have

$$
F_j \le F_{j-1} + \delta \max_{i=1,\ldots,h} \left\{\alpha_{hj+i-2}\left\langle \operatorname{grad} f(\widetilde{V}^{hj+i-2}, \widetilde{P}^{hj+i-2}),\ (\widetilde{\xi}_{hj+i-2}, \widetilde{\eta}_{hj+i-2})\right\rangle\right\}. \tag{7.6}
$$

Since F_j is bounded below, we deduce by summing (7.6) over j that (7.5) is true. □

7.4 Proof of Theorem 4.1

Proof We prove this theorem by contradiction. Suppose that $\liminf_{k\to\infty} \|\operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k)\| \neq 0$ This means there exists a constant $\gamma \in (0, 1)$ such that

$$
\|\operatorname{grad}\ f(\widetilde{V}^k, \widetilde{P}^k)\| \ge \gamma, \quad \text{for all } k. \tag{7.7}
$$

From (7.1), we can see that the formula (4.4) of β^D_{k+1} can be rewritten as

$$
\beta^D_{k+1} = \frac{\langle \operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ (\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1})\rangle}{r_{k+1}\langle \operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ \mathcal{T}_{\alpha_k(\widetilde{\xi}_k, \widetilde{\eta}_k)}(\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle - \max\{Y_{k+1},\ -\langle \operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle\}}. \tag{7.8}
$$

By (4.3), it is easy to prove that the denominator of β^D_{k+1} in (7.8) is equivalent to

$$
\begin{aligned}
& \langle \operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle - \max\Big\{(1 - r_{k+1})\langle \operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ \mathcal{T}_{\alpha_k(\widetilde{\xi}_k, \widetilde{\eta}_k)}(\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle, \\
& \quad -r_{k+1}\langle \operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ \mathcal{T}_{\alpha_k(\widetilde{\xi}_k, \widetilde{\eta}_k)}(\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle\Big\}.
\end{aligned} \tag{7.9}
$$

Substituting (7.9) into the denominator of β^D_{k+1} in (7.8), together with the results of Lemma 4.2, we have

$$
0 \le \beta_{k+1} \le \beta^D_{k+1} \le \frac{\langle \operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ (\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1})\rangle}{\langle \operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle}. \tag{7.10}
$$

It follows from (4.2) that $(\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1}) + \operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}) = \beta_{k+1}\mathcal{T}_{\alpha_k(\widetilde{\xi}_k, \widetilde{\eta}_k)}(\widetilde{\xi}_k, \widetilde{\eta}_k)$. Hence

$$
\begin{aligned}
\|(\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1})\|^2 = & -2\langle \operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ (\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1})\rangle \\
& - \|\operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1})\|^2 + \beta^2_{k+1}\|\mathcal{T}_{\alpha_k(\widetilde{\xi}_k, \widetilde{\eta}_k)}(\widetilde{\xi}_k, \widetilde{\eta}_k)\|^2.
\end{aligned} \tag{7.11}
$$

Dividing (7.11) by $\langle \operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ (\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1})\rangle^2$, using (7.11), and together with the Ring-Wirth nonexpansive condition (3.15) of the presented vector transport defined in (3.12), we can get

$$
\begin{aligned}
&\frac{\|(\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1})\|^2}{\langle \operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ \ (\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1})\rangle^2}\\
&\quad \le -\frac{2}{\langle \operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ \ (\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1})\rangle}\\
&\qquad -\frac{\|\operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1})\|^2}{\langle \operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ \ (\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1})\rangle^2} + \frac{\|\mathcal{T}_{\alpha_k(\widetilde{\xi}_k, \widetilde{\eta}_k)}(\widetilde{\xi}_k, \widetilde{\eta}_k)\|^2}{\langle \operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ \ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle^2}\\
&\quad \le -\frac{2}{\langle \operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ \ (\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1})\rangle}\\
&\qquad -\frac{\|\operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1})\|^2}{\langle \operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ \ (\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1})\rangle^2} + \frac{\|(\widetilde{\xi}_k, \widetilde{\eta}_k)\|^2}{\langle \operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ \ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle^2}\\
&\quad = -\left(\frac{\|\operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1})\|}{\langle \operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1}),\ \ (\widetilde{\xi}_{k+1}, \widetilde{\eta}_{k+1})\rangle^2} + \frac{1}{\|\operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1})\|^2}\right)^2\\
&\qquad + \frac{1}{\|\operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1})\|^2} + \frac{\|(\widetilde{\xi}_k, \widetilde{\eta}_k)\|^2}{\langle \operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ \ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle^2}\\
&\quad \le \frac{1}{\|\operatorname{grad} f(\widetilde{V}^{k+1}, \widetilde{P}^{k+1})\|^2} + \frac{\|(\widetilde{\xi}_k, \widetilde{\eta}_k)\|^2}{\langle \operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ \ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle^2}.
\end{aligned}
\tag{7.12}
$$

Combining the recursion of (7.12) and the assumption (7.7) together gives

$$
\begin{aligned}
\frac{\|(\widetilde{\xi}_k, \widetilde{\eta}_k)\|^2}{\langle \operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ \ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle^2} &= \sum_{i=1}^{k} \frac{1}{\|\operatorname{grad} f(\widetilde{V}^i, \widetilde{P}^i)\|^2} + \frac{\|(\widetilde{\xi}_0, \widetilde{\eta}_0)\|^2}{\langle \operatorname{grad} f(\widetilde{V}^0, \widetilde{P}^0),\ \ (\widetilde{\xi}_0, \widetilde{\eta}_0)\rangle^2}\\
&= \sum_{i=1}^{k} \frac{1}{\|\operatorname{grad} f(\widetilde{V}^i, \widetilde{P}^i)\|^2} + 1 \le \frac{k+1}{\gamma^2}.
\end{aligned}
\tag{7.13}
$$

Then we have that

$$
\frac{\langle \operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ \ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle^2}{\|(\widetilde{\xi}_k, \widetilde{\eta}_k)\|^2} \ge \frac{\gamma^2}{k+1}.
\tag{7.14}
$$

On the other hand, from the second inequality in (7.13) and (7.12), we have

$$
\begin{aligned}
&\frac{\|\operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k)\|^2}{\langle \operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ \ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle^2} + \frac{2}{\langle \operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ \ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle}\\
&\quad \le \frac{\|(\widetilde{\xi}_{k-1}, \widetilde{\eta}_{k-1})\|^2}{\langle \operatorname{grad} f(\widetilde{V}^{k-1}, \widetilde{P}^{k-1}),\ \ (\widetilde{\xi}_{k-1}, \widetilde{\eta}_{k-1})\rangle^2} \le \frac{k}{\gamma^2},
\end{aligned}
$$

which yields

$$
\frac{k}{\gamma^2}\langle \operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ \ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle^2 \ge 2\langle \operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ \ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle + \|\operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k)\|^2.
\tag{7.15}
$$

If

$$
-\langle \operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ \ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle \le \frac{3}{8}\|\operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k)\|^2,
$$

then together with (7.15) gives

$$
\langle \operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k),\ \ (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle^2 \ge \frac{\gamma^2}{4k}\|\operatorname{grad} f(\widetilde{V}^k, \widetilde{P}^k)\|^2.
$$

Therefore, by (4.11) and (7.7), we have

$$-\langle \mathrm{grad} f(\widetilde{V}^k, \widetilde{P}^k), \quad (\widetilde{\xi}_k, \widetilde{\eta}_k)\rangle \geq \min\left\{\frac{3\gamma^2}{8}, \frac{\gamma^2}{2\sqrt{k}}\right\}. \tag{7.16}$$

Using $\overline{\alpha}_k \geq \alpha_{min}$, together with (4.12), (7.14) and (7.16), we have

$$\begin{aligned}
&\sum_{j\geq 1} \min_{i=1,\ldots,h} \left\{-\alpha_{hj+i-2}\left\langle \mathrm{grad}\ f(\widetilde{V}^{hj+i-2}, \widetilde{P}^{hj+i-2}), \quad (\widetilde{\xi}_{hj+i-2}, \widetilde{\eta}_{hj+i-2})\right\rangle\right\} \\
&\geq \sum_{j\geq 1} \min_{i=1,\ldots,h} \min\left\{-\overline{\alpha}_{hj+i-2}\left\langle \mathrm{grad}\ f(\widetilde{V}^{hj+i-2}, \widetilde{P}^{hj+i-2}), \quad (\widetilde{\xi}_{hj+i-2}, \widetilde{\eta}_{hj+i-2})\right\rangle,\right. \\
&\quad \left.\mu\frac{\langle \mathrm{grad} f(\widetilde{V}^{hj+i-2}, \widetilde{P}^{hj+i-2}), \quad (\widetilde{\xi}_{hj+i-2}, \widetilde{\eta}_{hj+i-2})\rangle}{\|(\widetilde{\xi}_{hj+i-2}, \widetilde{\eta}_{hj+i-2})\|^2}\right\} \\
&\geq \sum_{j\geq 1} \min_{i=1,\ldots,h} \min\left\{\frac{3\gamma^2\alpha_{\min}}{8}, \frac{\gamma^2\alpha_{\min}}{2\sqrt{hj+i-2}}, \frac{\mu\gamma^2}{hj+i-1}\right\} \\
&\geq \sum_{j\geq 1} \min\left\{\frac{3\gamma^2\alpha_{\min}}{8}, \frac{\gamma^2\alpha_{\min}}{2\sqrt{h(j+1)}}, \frac{\mu\gamma^2}{h(j+1)}\right\} = +\infty,
\end{aligned}$$

which contradicts with (4.13). This prove (4.14). The existence of a first-order critical accumulation point follows immediately from the compactness of the product manifold Stp$(n, l) \times$ Stp$(2l, l)$ in Lemma 4.1. □

References

1. Boutry, G., Elad, M., Golub, G.H., Milanfar, P.: The generalized eigenvalue problem for nonsquare pencils using a minimal perturbation approach. SIAM J. Matrix Anal. Appl. **27**(2), 582–601 (2005)
2. Chu, D., Golub, G.H.: On a generalized eigenvalue problem for nonsquare pencils. SIAM J. Matrix Anal. Appl. **28**(3), 770–787 (2006)
3. Kressner, D., Mengi, E., Nakić, I., Truhar, N.: Generalized eigenvalue problems with specified eigenvalues. IMA J. Numer. Anal. **34**(2), 480–501 (2013)
4. Lecumberri, P., Gómez, M., Carlosena, A.: Generalized eigenvalues of nonsquare pencils with structure. SIAM J. Matrix Anal. Appl. **30**(1), 41–55 (2008)
5. Ito, S., Murota, K.: An algorithm for the generalized eigenvalue problem for nonsquare matrix pencils by minimal perturbation approach. SIAM J. Matrix Anal. Appl. **37**(1), 409–419 (2016)
6. Golub, G.H., Loan, C.F.V.: An analysis of the total least squares problem. SIAM J. Numer. Anal. **17**(6), 883–893 (1980)
7. Dai, Y.: A nonmonotone conjugate gradient algorithm for unconstrained optimization. J. Syst. Sci. Complex. **15**, 139–145 (2002)
8. Sun, J.: Matrix Perturbation Analysis. Science Press, Beijing (2001)
9. Sato, H., Iwai, T.: A complex singular value decomposition algorithm based on the Riemannian Newton method. In: 52nd IEEE Conference on Decision and Control, pp. 2972–2978. IEEE (2013)
10. Sato, H.: Riemannian conjugate gradient method for complex singular value decomposition problem. In: 53rd IEEE Conference on Decision and Control, pp. 5849–5854. IEEE (2014)
11. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2009)
12. 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)
13. 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)
14. Sato, H., Iwai, T.: A Riemannian optimization approach to the matrix singular value decomposition. SIAM J. Optim. **23**(1), 188–212 (2013)

15. Wen, Z., Yin, W.: A feasible method for optimization with orthogonality constraints. Math. Program. **142**(1–2), 397–434 (2013)
16. Zhu, X.: A Riemannian conjugate gradient method for optimization on the Stiefel manifold. Comput. Optim. Appl. **67**(1), 73–110 (2017)
17. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl. **20**(2), 303–353 (1998)
18. Absil, P.A., Baker, C., Gallivan, K.: A truncated-CG style method for symmetric generalized eigenvalue problems. J. Comput. Appl. Math. **189**(1–2), 274–285 (2006)
19. Baker, C., Absil, P.A., Gallivan, K.A.: An implicit Riemannian trust-region method for the symmetric generalized eigenproblem. In: International Conference on Computational Science, pp. 210–217. Springer, Berlin (2006)
20. 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)
21. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA J. Numer. Anal. **8**(1), 141–148 (1988)

Affiliations

Jiao-fen Li[1] · Wen Li[2] · Seak-Weng Vong[3] · Qi-Lun Luo[4] · MingQing Xiao[4]

Jiao-fen Li
lixiaogui1290@163.com

Seak-Weng Vong
swvong@umac.mo

MingQing Xiao
mxiao@siu.edu

[1] School of Mathematics and Computing Science, Guangxi Colleges and Universities Key Laboratory of Data Analysis and Computation, Guilin University of Electronic Technology, Guilin, China

[2] School of Mathematical Sciences, South China Normal University, Guangzhou, China

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

[4] Department of Mathematics, Southern Illinois University, Carbondale, IL 62901-4408, USA