



ELSEVIER

Available online at www.sciencedirect.com



Journal of Computational and Applied Mathematics 426 (2023) 1–22

Journal
Logo

On the convergence of orthogonalization-free conjugate gradient method for extreme eigenvalues of Hermitian matrices: a Riemannian optimization interpretation

Shixin Zheng^a, Haizhao Yang^b, Xiangxiong Zhang^c

^aDepartment of Mathematics, Purdue University, West Lafayette, IN, USA, zheng513@purdue.edu

^bDepartment of Mathematics, University of Maryland College Park, MD, USA, hzyang@umd.edu

^cDepartment of Mathematics, Purdue University, West Lafayette, IN, USA, zhan1966@purdue.edu

Abstract

In many applications, it is desired to obtain extreme eigenvalues and eigenvectors of large Hermitian matrices by efficient and compact algorithms. In particular, orthogonalization-free methods are preferred for large-scale problems for finding eigenspaces of extreme eigenvalues without explicitly computing orthogonal vectors in each iteration. For the top p eigenvalues, the simplest orthogonalization-free method is to find the best rank- p approximation to a positive semi-definite Hermitian matrix by algorithms solving the unconstrained Burer-Monteiro formulation. We show that the nonlinear conjugate gradient method for the unconstrained Burer-Monteiro formulation is equivalent to a Riemannian conjugate gradient method on a quotient manifold with the Bures-Wasserstein metric, thus its global convergence to a stationary point can be proven. Numerical tests suggest that it is efficient for computing the largest k eigenvalues for large-scale matrices if the largest k eigenvalues are nearly distributed uniformly.

Keywords:

Hermitian matrices, extreme eigenvalues, orthogonalization free, conjugate gradient, Riemannian optimization, quotient manifold, Bures-Wasserstein metric

1. Introduction

1.1. The eigenvalue problem of Hermitian positive definite matrices

In this paper, we are interested in solving the eigenvalue problem for a Hermitian matrix $B \in \mathbb{C}^{n \times n}$ to find its largest p eigenvalues and the corresponding eigenvectors. For large enough $\mu > 0$, $A := B + \mu I \in \mathbb{C}^{n \times n}$ is a positive definite Hermitian matrix with the same extreme eigenspaces. Thus we focus only on Hermitian positive definite or semi-definite matrices.

Extreme eigenvalue problems for Hermitian matrices naturally arise in many applications [1, 2, 3, 4, 5, 6, 7]. For example, many problems can be cast as a graph, for which the adjacency matrix and the graph Laplacian are real symmetric thus Hermitian [8]. The extreme eigenvalues and eigenvectors of these matrices contain information about the graph and the point cloud data such as diffusion maps [9]. Notice that the discussion in this paper also applies to the smallest k eigenvalues for a positive definite Hermitian matrix B by considering either $A = \mu I - B$ with large enough μ or $A = B^{-1}$ if an efficient implementation of linear system solver for $Bx = b$ is available, i.e., the matrix-vector multiplication $B^{-1}b$ can be efficiently implemented.

The extreme eigenvalue problem can be written as an optimization problem, with many different cost functions to consider. The most well-known one is to minimize the multicolumn Rayleigh quotient

$$\underset{x \in \mathbb{C}^{n \times p}}{\text{minimize}} \quad f(x) := \text{tr}((x^* x)^{-1} x^* A x). \quad (1)$$

If assuming the spectrum of x^*x is bounded by one and take the inverse of x^*x as the first order approximation of the Neumann series expansion, then as an approximation to multicolumn Rayleigh quotient, a popular method known as orbital minimization method (OMM) is to minimize the cost function [10]:

$$\underset{x \in \mathbb{C}^{n \times p}}{\text{minimize}} \quad f(x) := \text{tr}((2I - x^*x)x^*Ax) . \quad (2)$$

Another simple formulation is to consider optimization over the noncompact Stiefel manifold $\mathbb{C}_*^{n \times p} = \{X \in \mathbb{C}^{n \times p} : \text{rank}(X)=p\}$:

$$\underset{x \in \mathbb{C}_*^{n \times p}}{\text{minimize}} \quad f(x) := \frac{1}{2} \|xx^* - A\|_F^2 , \quad (3)$$

where $\|\cdot\|_F$ is the matrix Frobenius norm. Various orthogonalization-free algorithms for solving both (2) and (3) were considered and compared numerically in [11].

1.2. The real inner product and Fréchet derivatives

In this paper, we mainly focus on the cost function (3) and consider the nonlinear conjugate gradient (CG) methods solving (3).

Since $f(x)$ is real-valued and thus not holomorphic, $f(x)$ does not have a complex derivative with respect to $x \in \mathbb{C}^{n \times p}$. The linear spaces of complex matrices will therefore be regarded as vector spaces over \mathbb{R} . For any real vector space \mathcal{E} , the inner product on \mathcal{E} is denoted by $\langle \cdot, \cdot \rangle_{\mathcal{E}}$. For real matrices $A, B \in \mathbb{R}^{n \times p}$, the Hilbert–Schmidt inner product is $\langle A, B \rangle_{\mathbb{R}^{n \times p}} = \text{tr}(A^T B)$. Let $\Re(A)$ and $\Im(B)$ represent the real and imaginary parts of a complex matrix A . For $A, B \in \mathbb{C}^{n \times p}$, the real inner product for the real vector space $\mathbb{C}^{n \times p}$ then equals

$$\langle A, B \rangle_{\mathbb{C}^{n \times p}} := \Re(\text{tr}(A^* B)), \quad (4)$$

where $*$ is the conjugate transpose. We emphasize that (4) is a real inner product, rather than the complex Hilbert–Schmidt inner product. It is straightforward to verify that (4) can be written as

$$\langle A, B \rangle_{\mathbb{C}^{n \times p}} = \text{tr}(\Re(A)^T \Re(B)) + \text{tr}(\Im(A)^T \Im(B)) = \langle \Re(A), \Re(B) \rangle_{\mathbb{R}^{n \times p}} + \langle \Im(A), \Im(B) \rangle_{\mathbb{R}^{n \times p}} .$$

With the real inner product (4) for the real vector space $\mathbb{C}^{n \times p}$, a Fréchet derivative for the real-valued function $f(x)$ can be defined as

$$\nabla f(x) = \nabla f_{\Re(x)}(x) + i \nabla f_{\Im(x)}(x) \in \mathbb{C}^{n \times p}, \quad (5)$$

where $\nabla f_{\Re(x)}(x), \nabla f_{\Im(x)}(x) \in \mathbb{R}^{n \times p}$ are the gradient of the cost function f with respect to the real and imaginary parts of x , respectively. In particular, for $f(x) = \frac{1}{2} \|\mathcal{A}(xx^*) - b\|_F^2$ with a linear operator \mathcal{A} , the Fréchet derivative (5) becomes

$$\nabla f(x) = 2\mathcal{A}^*(\mathcal{A}(xx^*) - b)x,$$

where \mathcal{A}^* is the adjoint operator of \mathcal{A} . See Appendix in [12] for details.

1.3. The conjugate gradient method solving the Burer-Monteiro formulation

Notice that $\mathbb{C}_*^{n \times p}$ is an open set in the Euclidean space $\mathbb{C}^{n \times p}$, thus any line search method $x_{k+1} = x_k + \alpha_k \eta_k$ starting with the iterate $x_k \in \mathbb{C}_*^{n \times p}$ and a small enough step size α_k will give $x_{k+1} \in \mathbb{C}_*^{n \times p}$. Therefore, any such line search algorithm can be regarded as the same algorithm solving an unconstrained problem with a non-degenerate $x_k \in \mathbb{C}_*^{n \times p}$:

$$\underset{x \in \mathbb{C}^{n \times p}}{\text{minimize}} \quad f(x) := \frac{1}{2} \|xx^* - A\|_F^2 . \quad (6)$$

In the literature, the formulation (6) is often called the Burer-Monteiro method for Hermitian positive semi-definite (PSD) fixed rank p constraint, i.e., for minimizing $\|X - A\|_F^2$ where X is a Hermitian PSD matrix of rank p .

The nonlinear conjugate gradient method for (6) can be written as

$$\begin{cases} x_{k+1} &= x_k + \alpha_k \eta_k, \\ \eta_{k+1} &= -\nabla f(x_k) + \beta_k \eta_k = -2(xx^* - A)x + \beta_k \eta_k, \end{cases} \quad (7)$$

where α_k is the step size, β_k is a nonlinear coefficient computed by various formulae, and η_k is the search direction in CG method. In this paper, we only consider two variants for how to compute β_k : one is the Polak–Ribière CG method, and the other one is the Fletcher–Reeves CG method for computing the conjugate direction [13].

1.4. The main result: the convergence of Riemannian conjugate gradient method via quotient geometry

The CG method (7) for finding top p eigenvalues of Hermitian PSD matrix A has been considered in [11]. In particular, (7) does not require any orthogonalization operation in each iteration, and its performance is superior especially for uniformly distributed eigenvalues in numerical tests.

The landscape of (6) has been well studied in [11, 14, 15, 16] and its local minimizers must also be global minimizers. Theorem 2.1 in [11] implies that, if $\hat{x} \in \mathbb{C}_*^{n \times p}$ satisfies $\nabla f(\hat{x}) = 0$ for $f(x) = \frac{1}{2} \|xx^* - A\|_F^2$, then $\hat{x} = UO$ where $O \in \mathbb{C}^{p \times p}$ is a unitary matrix, and $U \in \mathbb{C}^{n \times p}$ has orthogonal columns as some eigenvectors of A . Furthermore, any local minimum is a global minimum, i.e., any local minimizer of (6) in $\mathbb{C}_*^{n \times p}$ has the form $\hat{x} = UO$ with columns of U being eigenvectors of a Hermitian PSD matrix A corresponding to its top p eigenvectors.

However, the convergence of CG method (7) for (6) has never been rigorously justified.

Notice that there is an ambiguity up to unitary matrices in both formulations (6) and (3), that is $f(xO) = f(x)$ for any $O \in \mathcal{O}_p$, where \mathcal{O}_p are all $p \times p$ unitary matrices. To this end, mathematically it is proper to consider an equivalence class for each $x \in \mathbb{C}_*^{n \times p}$:

$$[x] = \{xO : \forall O \in \mathcal{O}_p\},$$

and a quotient set

$$\mathbb{C}_*^{n \times p} / \mathcal{O}_p := \{[x] : \forall x \in \mathbb{C}_*^{n \times p}\}.$$

The quotient set with a proper metric becomes a quotient manifold. It is not uncommon to abuse notation by letting x denote the equivalent class $[x]$, and \bar{x} denote one representation of this equivalent class. So we can instead consider the optimization over the quotient manifold:

$$\underset{x \in \mathbb{C}_*^{n \times p} / \mathcal{O}_p}{\text{minimize}} \quad h(x) := f(\bar{x}) = \frac{1}{2} \|\bar{x}\bar{x}^* - A\|_F^2. \quad (8)$$

Following the recent progress in [12] for Riemannian optimization over Hermitian PSD fixed rank manifolds, we first show that the simple unconstrained Burer-Monteiro CG method (7) is equivalent to a Riemannian CG method solving (8) over the quotient manifold $\mathbb{C}_*^{n \times p} / \mathcal{O}_p$ with the Bures-Wasserstein metric [17] and proper retraction and vector transport operators. Then with existing Riemannian optimization convergence theory, we can establish the global convergence of the simple algorithm (7) to a stationary point of (3). We emphasize that the main result of this paper is the global convergence proof for the classical simple algorithm (7), and we do not modify the algorithm (7) at all. The Riemannian optimization is used only for proving convergence of (7), and (7) should not be implemented via much more complicated Riemannian optimization over a quotient manifold.

1.5. Related work and contributions

To be more specific, we will show that both the Polak–Ribière CG method and the Fletcher-Reeves CG method in (7) are equivalent to their Riemannian variants over the quotient manifold $\mathbb{C}_*^{n \times p} / \mathcal{O}_p$ with the Bures-Wasserstein metric [17].

Moreover, this equivalence allows us to establish the global convergence of the conventional Fletcher-Reeves CG method (7) to a stationary point of (3), following the convergence of the Riemannian Fletcher-Reeves CG method in [18]. For the problem (6), it has been well known that local minima are also global minima [14, 15, 16, 11], e.g., critical points are either global minima or saddle points. Combined with the result that first-order methods almost always avoid strict saddle points [19], we obtain a justification of the global convergence of the conventional Fletcher-Reeves CG method (7) to the global minimizer of (3). For the Polak–Ribière CG method, the convergence is much harder to establish, but its numerical performance is often superior.

In the literature, notable convergence results for orthogonalization-free methods include global convergence of perturbed gradient descent for (6) in [15] and global convergence of TriOFM in [20].

The same CG algorithm (7) was also considered in [11] for real symmetric matrices. Both our algorithm and convergence proof also apply to the Hermitian matrices. We also verify the numerical performance of the discussed algorithms on large matrices of the size millions by millions. In particular, our numerical tests for large matrices are consistent with the observation in [11] that the simple CG method (7) is superior for nearly uniformly distributed extreme eigenvalues.

This paper mainly focuses on the convergence analysis of the simplest orthogonalization-free method (7) which is fully scalable in parallel computing. Developing distributed and parallel numerical implementation will be left as future work. In the literature, most numerical solvers for eigenvalue problems rely on orthogonalization to achieve high efficiency in sequential computing. Well-developed algorithms with orthogonalization include [21, 22, 23, 24]. To achieve better parallel efficiency for a full eigendecomposition, spectrum slicing can be applied to estimate different eigenpairs in different spectrum regions simultaneously [25, 26, 27, 28, 29, 30].

1.6. Outline of this paper

We first review basic concepts and known results for Riemannian quotient manifolds $\mathbb{C}_*^{n \times p}/O_p$ in Section 2. Then we review the equivalence of the conventional CG method to the Riemannian CG method in Section 3. The convergence proof of the Riemannian CG method is provided in Section 4. In Section 5, we show that the simple coordinate descent method of minimizing (6) is also equivalent to a coordinate Riemannian gradient descent method. Section 6 includes numerical tests. Concluding remarks are given in Section 7.

2. Preliminaries: Riemannian Quotient Manifold $\mathbb{C}_*^{n \times p}/O_p$

In this section, we briefly review some known results of the Riemannian geometry of $\mathbb{C}_*^{n \times p}/O_p$ that will be used in this paper. Any missing details can be found in [12].

2.1. $\mathbb{C}_*^{n \times p}/O_p$ as a quotient manifold

Define $\mathbb{C}_*^{n \times p} = \{X \in \mathbb{C}^{n \times p} : \text{rank}(X) = p\}$ and an equivalence relation on $\mathbb{C}_*^{n \times p}$ through the smooth Lie group action of unitary matrices O_p on the manifold $\mathbb{C}_*^{n \times p}$:

$$\mathbb{C}_*^{n \times p} \times O_p \rightarrow \mathbb{C}_*^{n \times p}, \quad (\bar{x}, O) \mapsto \bar{x}O.$$

This action defines an equivalence relation on $\mathbb{C}_*^{n \times p}$ by setting $\bar{x}_1 \sim \bar{x}_2$ if there exists an $O \in O_p$ such that $\bar{x}_1 = \bar{x}_2 O$. Hence we have constructed a quotient space $\mathbb{C}_*^{n \times p}/O_p$ that removes this ambiguity. The set $\mathbb{C}_*^{n \times p}$ is called the *total space* of $\mathbb{C}_*^{n \times p}/O_p$.

Denote the natural projection as

$$\pi : \mathbb{C}_*^{n \times p} \rightarrow \mathbb{C}_*^{n \times p}/O_p, \quad \bar{x} \mapsto x.$$

We denote the equivalence class containing \bar{x} as

$$[\bar{x}] = \pi^{-1}(x) = \{\bar{x}O | O \in O_p\}.$$

Following Corollary 21.6 and Theorem 21.10 of [31], $\mathbb{C}_*^{n \times p}/O_p$ is a smooth manifold as stated in the following theorem.

Theorem 2.1. *The quotient space $\mathbb{C}_*^{n \times p}/O_p$ is a quotient manifold over \mathbb{R} of dimension $2np - p^2$ and has a unique smooth structure such that the natural projection π is a smooth submersion.*

2.2. Vertical space

The equivalence class $[\bar{x}] = \pi^{-1}(x)$ is an embedded submanifold of $\mathbb{C}_*^{n \times p}$ ([32, Prop. 3.4.4]). The tangent space of $[\bar{x}]$ at \bar{x} is therefore a subspace of $\mathbb{C}_*^{n \times p}$ called the *vertical space* at \bar{x} and is denoted by $\mathcal{V}_{\bar{x}}$. The following proposition characterizes $\mathcal{V}_{\bar{x}}$.

Proposition 2.2. *The vertical space at $\bar{x} \in [\bar{x}] = \{\bar{x}O | O \in O_p\}$, which is the tangent space of $[\bar{x}]$ at \bar{x} is*

$$\mathcal{V}_{\bar{x}} = \{\bar{x}\Omega | \Omega^* = -\Omega, \Omega \in \mathbb{C}^{p \times p}\}.$$

2.3. Riemannian metric

A *Riemannian metric* g is a smoothly varying inner product defined on the tangent space. That is, $g_{\bar{x}}(\cdot, \cdot)$ is an inner product on $T_{\bar{x}}\mathbb{C}_*^{n \times p}$. Once we choose a Riemannian metric g for $\mathbb{C}_*^{n \times p}$, we can obtain the orthogonal complement in $T_{\bar{x}}\mathbb{C}_*^{n \times p}$ of $\mathcal{V}_{\bar{x}}$ with respect to the metric. In other words, we choose the *horizontal distribution* as orthogonal complement w.r.t. Riemannian metric, see [32, Section 3.5.8]. This orthogonal complement to $\mathcal{V}_{\bar{x}}$ is called *horizontal space* at \bar{x} and is denoted by $\mathcal{H}_{\bar{x}}$. We thus have

$$T_{\bar{x}}\mathbb{C}_*^{n \times p} = \mathcal{H}_{\bar{x}} \oplus \mathcal{V}_{\bar{x}}. \quad (9)$$

Once we have the horizontal space, there exists a unique vector $\bar{\xi}_{\bar{x}} \in \mathcal{H}_{\bar{x}}$ that satisfies $D\pi(\bar{x})[\bar{\xi}_{\bar{x}}] = \xi_x$ for each $\xi_x \in T_x\mathbb{C}_*^{n \times p}/\mathcal{O}_p$. This $\bar{\xi}_{\bar{x}}$ is called the *horizontal lift* of ξ_x at \bar{x} .

In this paper, we consider the Riemannian metric on $\mathbb{C}_*^{n \times p}$ to be the canonical Euclidean inner product on $\mathbb{C}^{n \times p}$ defined by

$$g_{\bar{x}}(A, B) := \langle A, B \rangle_{\mathbb{C}^{n \times p}} = \Re(\text{tr}(A^* B)), \quad \forall A, B \in T_{\bar{x}}\mathbb{C}_*^{n \times p} = \mathbb{C}^{n \times p}. \quad (10)$$

Proposition 2.3. *Under metric g defined in (10), the horizontal space at \bar{x} satisfies*

$$\mathcal{H}_{\bar{x}} = \{z \in \mathbb{C}^{n \times p} : \bar{x}^* z = z^* \bar{x}\} = \{\bar{x}(\bar{x}^* \bar{x})^{-1} S + \bar{x}_{\perp} K | S^* = S, S \in \mathbb{C}^{p \times p}, K \in \mathbb{C}^{(n-p) \times p}\}.$$

2.4. Projections onto vertical space and horizontal space

Due to the direct sum property (9), for our choices of \mathcal{H}_Y , there exist projection operators for any $z \in T_Y\mathbb{C}_*^{n \times p} = \mathbb{C}^{n \times p}$ to \mathcal{H}_Y as

$$z = P_{\bar{x}}^{\mathcal{V}}(z) + P_{\bar{x}}^{\mathcal{H}}(z).$$

It is straightforward to verify the following formulae for projection operators $P_Y^{\mathcal{V}}$ and $P_Y^{\mathcal{H}}$.

Proposition 2.4. *The orthogonal projections of any $z \in \mathbb{C}^{n \times p}$ to $\mathcal{V}_{\bar{x}}$ and $\mathcal{H}_{\bar{x}}$ are*

$$P_{\bar{x}}^{\mathcal{V}}(z) = \bar{x}\Omega, \quad P_{\bar{x}}^{\mathcal{H}}(z) = z - \bar{x}\Omega,$$

where Ω is the skew-symmetric matrix that solves the Lyapunov equation

$$\Omega \bar{x}^* \bar{x} + \bar{x}^* \bar{x} \Omega = \bar{x}^* z - z^* \bar{x}.$$

Remark 2.5. *The solution X to the Lyapunov equation $XE + EX = Z$ for a Hermitian E is unique if E is Hermitian positive-definite [33, Section 2.2]. Let $E = U\Lambda U^*$ be the SVD, then the Lyapunov equation $XE + EX = Z$ becomes*

$$(U^* X U) \Lambda + \Lambda (U^* X U) = U^* Z U,$$

which gives the solution

$$(U^* X U)_{i,j} = (U^* Z U)_{i,j} / (\Lambda_{i,i} + \Lambda_{j,j}).$$

2.5. $\mathbb{C}_*^{n \times p}/\mathcal{O}_p$ as Riemannian quotient manifold

First, we show in the following lemma the relationship between the horizontal lifts of the quotient tangent vector ξ_x lifted at different representatives in $[\bar{x}]$.

Lemma 2.6. *Let η be a vector field on $\mathbb{C}_*^{n \times p}/\mathcal{O}_p$, and let $\bar{\eta}$ be the horizontal lift of η . Then for each $\bar{x} \in \mathbb{C}_*^{n \times p}$, we have*

$$\bar{\eta}_{\bar{x}\mathcal{O}} = \bar{\eta}_{\bar{x}} \mathcal{O}, \quad \forall \mathcal{O} \in \mathcal{O}_p.$$

Proof. See [33, Prop. A.8] □

Recall from [32, Section 3.6.2] that if the expression $g_{\bar{x}}(\bar{\xi}_{\bar{x}}, \bar{\xi}_{\bar{x}})$ does not depend on the choice of $\bar{x} \in \pi^{-1}(\bar{x})$ for every $x \in \mathbb{C}_*^{n \times p}/\mathcal{O}_p$ and every $\xi_x, \zeta_x \in T_x\mathbb{C}_*^{n \times p}/\mathcal{O}_p$, then

$$g_x(\xi_x, \zeta_x) := g_{\bar{x}}(\bar{\xi}_{\bar{x}}, \bar{\zeta}_{\bar{x}}) \quad (11)$$

defines a Riemannian metric on the quotient manifold $\mathbb{C}_*^{n \times p}/\mathcal{O}_p$. By Lemma 2.6, it is straightforward to verify that the Riemannian metric (10) on $\mathbb{C}_*^{n \times p}$ induces a Riemannian metric on $\mathbb{C}_*^{n \times p}/\mathcal{O}_p$ defined as (11). The quotient manifold $\mathbb{C}_*^{n \times p}/\mathcal{O}_p$ endowed with a Riemannian metric defined in (11) is called a *Riemannian quotient manifold*. By abuse of notation, we use g for denoting Riemannian metrics on both total space $\mathbb{C}_*^{n \times p}$ and quotient space $\mathbb{C}_*^{n \times p}/\mathcal{O}_p$. This particular metric is also called the Bures-Wasserstein metric for PSD matrices of fixed-rank [17].

2.6. Riemannian gradient

The cost function of (6) induces a cost function on $\mathbb{C}_*^{n \times p} / \mathcal{O}_p$.

$$h : \mathbb{C}_*^{n \times p} / \mathcal{O}_p \rightarrow \mathbb{C}, \quad x \mapsto f(\bar{x}). \quad (12)$$

That is, $f = h \circ \pi$. Notice when we solve (3), we restrict f on the noncompact Stiefel manifold $\mathbb{C}_*^{n \times p}$, which is a submanifold of $\mathbb{C}^{n \times p}$. Hence the Riemannian gradient of f on $\mathbb{C}_*^{n \times p}$ at \bar{x} is the projection of the Fréchet gradient of f on $\mathbb{C}^{n \times p}$, denoted by $\nabla f(\bar{x})$, onto the tangent space $T_{\bar{x}}\mathbb{C}_*^{n \times p} = \mathbb{C}^{n \times p}$. Since ∇f is already in $\mathbb{C}^{n \times p}$, the projection is identity. That is,

$$\text{grad } f(\bar{x}) = \nabla f(\bar{x}). \quad (13)$$

Remark 2.7. One can refer to [12, Appendix A] for more details about Fréchet derivative. A Fréchet gradient for any real-valued function $f(X)$ at $X \in \mathbb{C}^{m \times n}$ can be defined as

$$\nabla f(X) = \nabla f_{\Re(X)}(X) + i \nabla f_{\Im(X)}(X) \in \mathbb{C}^{m \times n}, \quad (14)$$

where $\nabla f_{\Re(X)}(X), \nabla f_{\Im(X)}(X) \in \mathbb{R}^{m \times n}$ are the gradient of f with respect to the real and imaginary parts of X , respectively. In particular, for the cost function considered in this paper $f(\bar{x}) = \frac{1}{2} \|\bar{x}\bar{x}^* - A\|_F^2$, the Fréchet gradient (14) becomes

$$\nabla f(\bar{x}) = 2(\bar{x}\bar{x}^* - A)\bar{x}.$$

Now consider the Riemannian gradient of h at $x \in \mathbb{C}_*^{n \times p} / \mathcal{O}_p$. $\text{grad } h(x)$ is a tangent vector in $T_x\mathbb{C}_*^{n \times p} / \mathcal{O}_p$. The next theorem shows that the horizontal lift of $\text{grad } h(x)$ can be obtained from the Riemannian gradient of f .

Theorem 2.8. The horizontal lift of the Riemannian gradient of h at \bar{x} is the Riemannian gradient of f at \bar{x} . That is,

$$\overline{\text{grad } h(x)}_{\bar{x}} = \text{grad } f(\bar{x}).$$

Therefore, although $\text{grad } f(\bar{x})$ belongs in $\mathbb{C}^{n \times p}$, it is automatically in $\mathcal{H}_{\bar{x}}$.

Proof. See [32, Section 3.6.2]. □

2.7. Retraction

The retraction on the quotient manifold $\mathbb{C}_*^{n \times p} / \mathcal{O}_p$ can be defined using the retraction on the total space $\mathbb{C}_*^{n \times p}$. Let $Y \in \mathbb{C}_*^{n \times p}$, for any $Z \in \mathbb{C}^{n \times p}$ and a step size $\tau > 0$,

$$\bar{R}_Y(\tau Z) := Y + \tau Z,$$

is a retraction on $\mathbb{C}_*^{n \times p}$ if $Y + \tau Z$ remains full rank, which is ensured for small enough τ . Then Lemma 2.6 indicates that \bar{R} satisfies the conditions of [32, Prop. 4.1.3], which implies that

$$R_x(\tau \eta_x) := \pi(\bar{R}_{\bar{x}}(\tau \bar{\eta}_{\bar{x}})) = \pi(\bar{x} + \tau \bar{\eta}_{\bar{x}}) \quad (15)$$

defines a retraction on the quotient manifold $\mathbb{C}_*^{n \times p} / \mathcal{O}_p$ for a small enough step size $\tau > 0$.

2.8. Vector transport

We use differentiated retraction as our vector transport [32, Section 8.1.4].

$$\mathcal{T}_{\eta_x}(\xi_x) := D R_x(\eta_x)[\xi_x] = \left. \frac{d}{dt} \right|_{t=0} R_x(\eta_x + t \xi_x). \quad (16)$$

Notice that

$$\begin{aligned} \mathcal{T}_{\eta_x}(\xi_x) &= D R_x(\eta_x)[\xi_x] = D \pi(\bar{R}_{\bar{x}}(\bar{\eta}_{\bar{x}})) [D \bar{R}_{\bar{x}}(\bar{\eta}_{\bar{x}})[\bar{\xi}_{\bar{x}}]] = D \pi(\bar{x} + \bar{\eta}_{\bar{x}}) \left[\left. \frac{d}{dt} \right|_{t=0} \bar{R}_{\bar{x}}(\bar{\eta}_{\bar{x}} + t \bar{\xi}_{\bar{x}}) \right] \\ &= D \pi(\bar{x} + \bar{\eta}_{\bar{x}}) \left[\left. \frac{d}{dt} \right|_{t=0} (\bar{x} + \bar{\eta}_{\bar{x}} + t \bar{\xi}_{\bar{x}}) \right] = D \pi(\bar{x} + \bar{\eta}_{\bar{x}}) [\bar{\xi}_{\bar{x}}] = D \pi(\bar{x} + \bar{\eta}_{\bar{x}}) [P_{\bar{x} + \bar{\eta}_{\bar{x}}}^{\mathcal{H}}(\bar{\xi}_{\bar{x}})]. \end{aligned}$$

Hence the horizontal lift of a transported vector is simply the projection of the original horizontal lift to the new horizontal space, as shown in the following formula.

$$\overline{\mathcal{T}_{\eta_x}(\xi_x)}_{\bar{x} + \bar{\eta}_{\bar{x}}} = P_{\bar{x} + \bar{\eta}_{\bar{x}}}^{\mathcal{H}}(\bar{\xi}_{\bar{x}}). \quad (17)$$

3. The Conjugate Gradient Methods

We first recall the traditional conjugate gradient method for solving (6), which is summarized as Algorithm 1. We present the abstract Riemannian conjugate gradient method for solving (8) over the quotient manifold as Algorithm 2, with Wolfe conditions

$$h(R_{x_k}(\alpha_k \eta_k)) \leq h(x_k) + c_1 \alpha_k g_{x_k}(\text{grad } h(x_k), \eta_k), \quad (18)$$

$$\left| g_{R_{x_k}(\alpha_k \eta_k)}(\text{grad } h(R_{x_k}(\alpha_k \eta_k)), D R_{x_k}(\alpha_k \eta_k)[\eta_k]) \right| \leq c_2 \left| g_{x_k}(\text{grad } h(x_k), \eta_k) \right|. \quad (19)$$

$0 < c_1 < c_2 < 1$.

The abstract Algorithm 2 can be implemented as Algorithm 3, in which each tangent vector is treated as horizontal lift and each iterate is a representative of its equivalence class, and it is independent of the choice of the representative of the equivalent class.

Algorithm 1 (Polak–Ribière or Fletcher–Reeves) Conjugate Gradient on $\mathbb{C}^{n \times p}$

Require: initial iterate $x_0 \in \mathbb{C}^{n \times p}$, tolerance $\varepsilon > 0$, initial descent direction as negative gradient $\eta_0 = -\nabla f(x_0) = -2(x_0 x_0^* - A)x_0$

1: **for** $k = 0, 1, 2, \dots$ **do**

2: Use backtracking to compute the step size $\alpha_k > 0$ satisfying the strong Wolfe conditions

3: Obtain the new iterate by

$$x_{k+1} = x_k + \alpha_k \eta_k$$

4: Compute the gradient

$$\xi_{k+1} := \nabla f(x_{k+1})$$

5: Check for convergence

 if $\|\xi_{k+1}\|_F < \varepsilon$, then break

6: Compute a conjugate direction by the Polak–Ribière method or the Fletcher–Reeves method

$$\eta_{k+1} = -\xi_{k+1} + \beta_{k+1} \eta_k$$

$$\text{where } \beta_{k+1} = \begin{cases} \max\left(0, \frac{\langle \nabla f(x_{k+1}), \nabla f(x_{k+1}) - \nabla f(x_k) \rangle}{\langle \nabla f(x_k), \nabla f(x_k) \rangle}\right) & \text{if using Polak–Ribière} \\ \frac{\langle \nabla f(x_{k+1}), \nabla f(x_{k+1}) \rangle}{\langle \nabla f(x_k), \nabla f(x_k) \rangle} & \text{if using Fletcher–Reeves.} \end{cases}$$

7: **end for**

Algorithm 2 Riemannian Conjugate Gradient on the quotient manifold $\mathbb{C}_*^{n \times p} / \mathcal{O}_p$ with metric g **Require:** initial iterate $x_0 \in \mathbb{C}_*^{n \times p} / \mathcal{O}_p$, tolerance $\varepsilon > 0$, tangent vector $\eta_0 = -\text{grad } h(x_0)$ 1: **for** $k = 0, 1, 2, \dots$ **do**2: Compute the step size $\alpha_k > 0$ satisfying the strong Wolfe conditions (18) and (19)

3: Obtain the new iterate by retraction

$$x_{k+1} = R_{x_k}(\alpha_k \eta_k)$$

4: Compute the gradient

$$\xi_{k+1} := \text{grad } h(x_{k+1})$$

5: Check for convergence

$$\text{if } \|\xi_{k+1}\| := \sqrt{g_{x_{k+1}}(\xi_{k+1}, \xi_{k+1})} < \varepsilon, \text{ then break}$$

6: Compute a conjugate direction by the Polak–Ribière (PR₊) method or the Fletcher–Reeves (FR) method, and vector transport

$$\eta_{k+1} = -\xi_{k+1} + \beta_{k+1} \mathcal{T}_{\alpha_k \eta_k}(\eta_k)$$

$$\text{where } \beta_{k+1} = \begin{cases} \max \left(0, \frac{g_{x_{k+1}}(\text{grad } h(x_{k+1}), \text{grad } h(x_{k+1}) - \mathcal{T}_{\alpha_k \eta_k}(\xi_k))}{g_{x_k}(\text{grad } h(x_k), \text{grad } h(x_k))} \right) & \text{PR}_+ \\ \frac{g_{x_{k+1}}(\text{grad } h(x_{k+1}), \text{grad } h(x_{k+1}))}{g_{x_k}(\text{grad } h(x_k), \text{grad } h(x_k))} & \text{FR} \end{cases}$$

7: **end for****Algorithm 3** Implementation for Riemannian Conjugate Gradient on the quotient manifold $\mathbb{C}_*^{n \times p} / \mathcal{O}_p$ with metric g **Require:** initial iterate $\bar{x}_0 \in \mathbb{C}_*^{n \times p}$, tolerance $\varepsilon > 0$, initial descent direction as $\bar{\eta}_0 = -\text{grad } f(\bar{x}_0) = -2(\bar{x}_0 \bar{x}_0^* - A) \bar{x}_0$ 1: **for** $k = 0, 1, 2, \dots$ **do**2: Compute the step size $\alpha_k > 0$ satisfying the strong Wolfe conditions

3: Obtain the new iterate by retraction

$$\bar{x}_{k+1} = \bar{R}_{\bar{x}_k}(\alpha_k \bar{\eta}_k) = \bar{x}_k + \alpha_k \bar{\eta}_k$$

4: Compute the horizontal lift of gradient

$$\bar{\xi}_{k+1} := \text{grad } f(\bar{x}_{k+1}) = 2(\bar{x}_{k+1} \bar{x}_{k+1}^* - A) \bar{x}_{k+1}$$

5: Check for convergence

$$\text{if } \|\bar{\xi}_{k+1}\| := \sqrt{g_{\bar{x}_{k+1}}(\bar{\xi}_{k+1}, \bar{\xi}_{k+1})} < \varepsilon, \text{ then break}$$

6: Compute a conjugate direction by PR₊ or by FR and vector transport

$$\bar{\eta}_{k+1} = -\bar{\xi}_{k+1} + \beta_{k+1} \mathcal{T}_{\alpha_k \bar{\eta}_k}(\bar{\eta}_k)_{\bar{x}_{k+1}}$$

$$\text{where } \beta_{k+1} = \begin{cases} \max \left(0, \frac{g_{\bar{x}_{k+1}}(\text{grad } f(\bar{x}_{k+1}), \text{grad } f(\bar{x}_{k+1}) - \mathcal{T}_{\alpha_k \bar{\eta}_k}(\bar{\xi}_k)_{\bar{x}_{k+1}})}{g_{\bar{x}_k}(\text{grad } f(\bar{x}_k), \text{grad } f(\bar{x}_k))} \right) & \text{PR}_+ \\ \frac{g_{\bar{x}_{k+1}}(\text{grad } f(\bar{x}_{k+1}), \text{grad } f(\bar{x}_{k+1}))}{g_{\bar{x}_k}(\text{grad } f(\bar{x}_k), \text{grad } f(\bar{x}_k))} & \text{FR} \end{cases}$$

7: **end for**

The following results were first proven in [12]. For completeness, we include a detailed proof.

Lemma 3.1. *Let η_k be the descent direction generated by Algorithm 2. Then we have*

$$\mathcal{T}_{\alpha_k \eta_k}(\eta_k)_{\bar{x}_{k+1}} = P_{\bar{x}_k + \alpha_k \bar{\eta}_k}^{\mathcal{H}}(\bar{\eta}_k) = \bar{\eta}_k. \quad (20)$$

Proof. The first equality follows from (17). Recall the projection formula given in proposition 2.4. Denote $\bar{x}_{k+1} = \bar{x}_k + \alpha_k \bar{\eta}_k$. Then we have

$$P_{\bar{x}_k + \alpha_k \bar{\eta}_k}^{\mathcal{H}}(\bar{\eta}_k) = \bar{\eta}_k - \bar{x}_{k+1} \Omega_k. \quad (21)$$

Hence in order to show $P_{\bar{x}_k + \alpha_k \bar{\eta}_k}^{\mathcal{H}}(\bar{\eta}_k) = \bar{\eta}_k$, it is equivalent to show the Lyapunov equation

$$\Omega_k \bar{x}_{k+1}^* \bar{x}_{k+1} + \bar{x}_{k+1}^* \bar{x}_{k+1} \Omega_k = \bar{x}_{k+1}^* \bar{\eta}_k - \bar{\eta}_k^* \bar{x}_{k+1} \quad (22)$$

only has trivial solution $\Omega_k = 0$ for all $k \geq 0$.

The solution X to the Lyapunov equation $XE + EX = Z$ for a Hermitian E is unique if E is Hermitian positive-definite [33, Section 2.2]. Thus (22) has a unique solution if $\bar{x}_{k+1} \in \mathbb{C}_*^{n \times p}$. Thus we only need to show the right-hand side of the equation is zero. We prove this by induction.

When $k = 0$, the right hand side of (22) is

$$\bar{x}_1^* \bar{\eta}_0 - \bar{\eta}_0^* \bar{x}_1 = (\bar{x}_0 + \alpha_0 \bar{\eta}_0)^* \bar{\eta}_0 - \bar{\eta}_0^* (\bar{x}_0 + \alpha_0 \bar{\eta}_0) = \bar{x}_0^* \bar{\eta}_0 - \bar{\eta}_0^* \bar{x}_0 = -2\bar{x}_0^* (\bar{x}_0 \bar{x}_0^* - A) \bar{x}_0 + 2\bar{x}_0^* (\bar{x}_0 \bar{x}_0^* - A^*) \bar{x}_0 = 0.$$

Now suppose $\bar{x}_k^* \bar{\eta}_{k-1} - \bar{\eta}_{k-1}^* \bar{x}_k = 0$ and hence $P_{\bar{x}_k}^{\mathcal{H}}(\bar{\eta}_{k-1}) = \bar{\eta}_{k-1}$. Then

$$\begin{aligned} \bar{x}_{k+1}^* \bar{\eta}_k - \bar{\eta}_k^* \bar{x}_{k+1} &= (\bar{x}_k + \alpha_k \bar{\eta}_k)^* \bar{\eta}_k - \bar{\eta}_k^* (\bar{x}_k + \alpha_k \bar{\eta}_k) = \bar{x}_k^* \bar{\eta}_k - \bar{\eta}_k^* \bar{x}_k \\ &= \bar{x}_k^* \left(-\bar{\xi}_k + \beta_k P_{\bar{x}_k}^{\mathcal{H}}(\bar{\eta}_{k-1}) \right) - \left(-\bar{\xi}_k + \beta_k P_{\bar{x}_k}^{\mathcal{H}}(\bar{\eta}_{k-1}) \right)^* \bar{x}_k = \bar{x}_k^* \left(-\bar{\xi}_k + \beta_k \bar{\eta}_{k-1} \right) - \left(-\bar{\xi}_k + \beta_k \bar{\eta}_{k-1} \right)^* \bar{x}_k \\ &= -\bar{x}_k^* \bar{\xi}_k + \bar{\xi}_k^* \bar{x}_k = -2\bar{x}_k^* (\bar{x}_k \bar{x}_k^* - A) \bar{x}_k + 2\bar{x}_k^* (\bar{x}_k \bar{x}_k^* - A^*) \bar{x}_k = 0. \end{aligned}$$

Hence $P_{\bar{x}_{k+1}}^{\mathcal{H}}(\bar{\eta}_k) = \bar{\eta}_k$ also holds and we have proved this lemma. \square

We can now state our first main result:

Theorem 3.2. *Algorithm 3 is equivalent to Algorithm 1, which is the conjugate gradient method solving (6), in the sense that they produce exactly the same iterates if started from the same initial point.*

Proof. By (13), the gradients generated by Algorithm 1 and Algorithm 3 are the same. By Lemma 3.1 and the equivalence between the Riemannian metric on $\mathbb{C}_*^{n \times p}$ and the inner product on $\mathbb{C}^{n \times p}$, we see that β_k generated by these two algorithms are also equivalent. Hence the conjugate directions are also the same. So the two algorithms generate the same iterates. \square

4. The Convergence of the Fletcher-Reeves Conjugate Gradient Method

In this section, we will prove that the Riemannian Fletcher-Reeves Conjugate Gradient method converges to a stationary point thus Algorithm 1 also converges by the equivalence Theorem 3.2.

The discussion in this section follows the same lines as in standard convergence theory, e.g., [18]. The cost function and vector transport considered in this paper satisfy the conditions for convergence analysis in [18]. Many results in this section are standard convergence results for a line search method, see [13]. For completeness, we include the full proof.

Let $\eta_k \in T_{x_k} \mathbb{C}_*^{n \times p} / \mathcal{O}_p$ be a descent direction. Define the angle θ_k between $-\text{grad } h(x_k)$ and η_k by

$$\cos \theta_k = -\frac{g_{x_k}(\text{grad } h(x_k), \eta_k)}{\|\text{grad } h(x_k)\|_{x_k} \|\eta_k\|_{x_k}}. \quad (23)$$

Let $\mathcal{L} := \{x \in \mathbb{C}_*^{n \times p} / \mathcal{O}_p : 0 \leq h(x) \leq h(x_0)\}$ and $\pi^{-1}(\mathcal{L}) = \{\bar{x} \in \mathbb{C}_*^{n \times p} : 0 \leq f(\bar{x}) \leq f(\bar{x}_0)\}$. We can show that $\pi^{-1}(\mathcal{L})$ is bounded.

Lemma 4.1. *There is a constant C such that $\|\bar{x}\|_F \leq C$, $\forall \bar{x} \in \pi^{-1}(\mathcal{L})$.*

Proof. Assume it is not true, then $\forall n \in \mathbb{N}, \exists \bar{x}_n \in \pi^{-1}(\mathcal{L})$ such that $\|\bar{x}_n\|_F \geq n$. Let $y_n = \frac{\bar{x}_n}{\|\bar{x}_n\|_F}$, then $\|y_n\|_F = 1$ and $\bar{x}_n = \|\bar{x}_n\|_F y_n = a_n y_n$ with $a_n \geq n$. Thus $f(\bar{x}_n) = \frac{1}{2} \|a_n^2 y_n y_n^* - A\|_F^2 \rightarrow \infty$ since $a_n \rightarrow \infty$ and $\|y_n\|_F = 1$. On the other hand, $\bar{x}_n \in \pi^{-1}(\mathcal{L})$ implies that $f(\bar{x}_n)$ should be bounded, which is a contradiction. \square

Lemma 4.2. *The Riemannian gradient of f , i.e., $\text{grad } f(\bar{x}) = 2(\bar{x}\bar{x}^* - A)\bar{x}$ is Lipschitz continuous on $\pi^{-1}(\mathcal{L})$. That is, there exists a constant $L > 0$ such that*

$$\|\text{grad } f(\bar{y}) - \text{grad } f(\bar{x})\|_F \leq L \|\bar{y} - \bar{x}\|_F, \quad \text{for all } \bar{x}, \bar{y} \in \pi^{-1}(\mathcal{L}). \quad (24)$$

Proof. It suffices to show that $q : \bar{x} \mapsto \bar{x}\bar{x}^*\bar{x}$ is Lipschitz continuous on $\pi^{-1}(\mathcal{L})$. Let $\bar{x}, \bar{y} \in \pi^{-1}(\mathcal{L})$. Then $\|\bar{x}\|_F \leq C, \|\bar{y}\|_F \leq C$ by Lemma 4.2.

$$\begin{aligned} \|q(\bar{x}) - q(\bar{y})\|_F &= \|\bar{x}\bar{x}^*\bar{x} - \bar{y}\bar{y}^*\bar{y}\|_F = \|\bar{x}\bar{x}^*\bar{x} - \bar{x}\bar{x}^*\bar{y} + \bar{x}\bar{x}^*\bar{y} - \bar{y}\bar{y}^*\bar{y}\|_F \\ &\leq \|\bar{x}\bar{x}^*\bar{x} - \bar{x}\bar{x}^*\bar{y}\|_F + \|\bar{x}\bar{x}^*\bar{y} - \bar{y}\bar{y}^*\bar{y}\|_F = \|\bar{x}\bar{x}^*\bar{x} - \bar{x}\bar{x}^*\bar{y}\|_F + \|\bar{x}\bar{x}^*\bar{y} - \bar{y}\bar{x}^*\bar{y} + \bar{y}\bar{x}^*\bar{y} - \bar{y}\bar{y}^*\bar{y}\|_F \\ &\leq \|\bar{x}\bar{x}^*\bar{x} - \bar{x}\bar{x}^*\bar{y}\|_F + \|\bar{x}\bar{x}^*\bar{y} - \bar{y}\bar{x}^*\bar{y}\|_F + \|\bar{y}\bar{x}^*\bar{y} - \bar{y}\bar{y}^*\bar{y}\|_F \\ &\leq \|\bar{x}\bar{x}^*\| \|\bar{x} - \bar{y}\|_F + \|\bar{x} - \bar{y}\|_F \|\bar{x}^*\|_F \|\bar{y}\|_F + \|\bar{y}\|_F \|\bar{x}^* - \bar{y}^*\|_F \|\bar{y}\|_F \leq 3C^2 \|\bar{x} - \bar{y}\|_F. \end{aligned}$$

□

Theorem 4.3 (Zoutendijk's theorem on manifold). *Let η_k be a descent direction and let α_k satisfy the strong Wolfe conditions (18) and (19). Then for the cost function h defined in 12, the following series converges.*

$$\sum_k \cos^2 \theta_k \|\text{grad } h(x_k)\|_{x_k}^2 < \infty.$$

Proof. From the strong Wolfe condition (19) we have

$$\begin{aligned} (c_2 - 1)g_{x_k}(\text{grad } h(x_k), \eta_k) &\leq g_{x_{k+1}}((\text{grad } h(R_{x_k}(\alpha_k \eta_k), D R_{x_k}(\alpha_k \eta_k)[\eta_k]) - g_{x_k}(\text{grad } h(x_k), \eta_k) \\ &= g_{\bar{x}_{k+1}}(\text{grad } f(\bar{x}_k + \alpha_k \bar{\eta}_k), P_{\bar{x}_k + \alpha_k \bar{\eta}_k}^H(\bar{\eta}_k)) - g_{\bar{x}_k}(\text{grad } f(\bar{x}_k), \bar{\eta}_k) \\ &= g_{\bar{x}_{k+1}}(\text{grad } f(\bar{x}_k + \alpha_k \bar{\eta}_k), \bar{\eta}_k) - g_{\bar{x}_k}(\text{grad } f(\bar{x}_k), \bar{\eta}_k). \end{aligned}$$

Notice that our Riemannian metric g is simply the inner product on the Euclidean space $\mathbb{C}^{n \times p}$, hence

$$g_{\bar{x}_{k+1}}(\text{grad } f(\bar{x}_k + \alpha_k \bar{\eta}_k), \bar{\eta}_k) - g_{\bar{x}_k}(\text{grad } f(\bar{x}_k), \bar{\eta}_k) = \langle \text{grad } f(\bar{x}_k + \alpha_k \bar{\eta}_k) - \text{grad } f(\bar{x}_k), \bar{\eta}_k \rangle. \quad (25)$$

From Lemma 4.2 we know

$$\langle \text{grad } f(\bar{x}_k + \alpha_k \bar{\eta}_k) - \text{grad } f(\bar{x}_k), \bar{\eta}_k \rangle \leq \alpha_k L \|\bar{\eta}_k\|_F^2.$$

Hence for any k we have

$$\alpha_k \geq \frac{(c_2 - 1)g_{x_k}(\text{grad } h(x_k), \eta_k)}{L \|\bar{\eta}_k\|_F^2}. \quad (26)$$

Now it follows from (18) and (26) that

$$\begin{aligned} 0 \leq h(x_{k+1}) &\leq h(x_k) + c_1 \alpha_k g_{x_k}(\text{grad } h(x_k), \eta_k) \\ &\leq h(x_k) - \frac{c_1(1 - c_2)}{L} \cos^2 \theta_k \|\text{grad } h(x_k)\|_{x_k}^2 \leq h(x_0) - \frac{c_1(1 - c_2)}{L} \sum_{j=0}^k \cos^2 \theta_j \|\text{grad } h(x_j)\|_{x_j}^2. \end{aligned}$$

Hence

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \|\text{grad } h(x_k)\|_{x_k}^2 \leq \frac{L}{c_1(1 - c_2)} h(x_0) < \infty. \quad (27)$$

□

Lemma 4.4. *If using Fletcher-Reeves method in Algorithm 2, then for $0 < c_1 < c_2 < 1/2$, the search direction η_k is a descent direction satisfying*

$$-\frac{1}{1 - c_2} \leq \frac{g_{x_k}(\text{grad } h(x_k), \eta_k)}{\|\text{grad } h(x_k)\|_{x_k}^2} \leq \frac{2c_2 - 1}{1 - c_2}. \quad (28)$$

Proof. We prove it by induction on k .

When $k = 0$, (28) holds since

$$\frac{g_{x_0}(\text{grad } h(x_0), \eta_0)}{\|\text{grad } h(x_0)\|_{x_0}^2} = \frac{g_{x_0}(\text{grad } h(x_0), -\text{grad } h(x_0))}{\|\text{grad } h(x_0)\|_{x_0}^2} = -1.$$

Now suppose (28) holds for some $k \geq 0$.

Recall that we use differentiated retraction as our vector transport:

$$\mathcal{T}_{\alpha_k \eta_k}(\eta_k) = \text{D } R_{x_k}(\alpha_k \eta_k)[\eta_k].$$

And the β_{k+1} in Fletcher-Reeves method is defined as

$$\beta_{k+1} = \frac{g_{x_{k+1}}(\text{grad } h(x_{k+1}), \text{grad } h(x_{k+1}))}{g_{x_k}(\text{grad } h(x_k), \text{grad } h(x_k))}.$$

Hence the middle term in (28) for $k + 1$ is

$$\begin{aligned} \frac{g_{x_{k+1}}(\text{grad } h(x_{k+1}), \eta_{k+1})}{\|\text{grad } h(x_{k+1})\|_{x_{k+1}}^2} &= \frac{g_{x_{k+1}}(\text{grad } h(x_{k+1}), -\text{grad } h(x_{k+1}) + \beta_{k+1} \mathcal{T}_{\alpha_k \eta_k}(\eta_k))}{\|\text{grad } h(x_{k+1})\|_{x_{k+1}}^2} \\ &= \frac{g_{x_{k+1}}(\text{grad } h(x_{k+1}), -\text{grad } h(x_{k+1}) + \beta_{k+1} \text{D } R_{x_k}(\alpha_k \eta_k)[\eta_k])}{\|\text{grad } h(x_{k+1})\|_{x_{k+1}}^2} \\ &= -1 + \frac{g_{x_{k+1}}(\text{grad } h(x_{k+1}), \text{D } R_{x_k}(\alpha_k \eta_k)[\eta_k])}{\|\text{grad } h(x_k)\|_{x_k}^2}. \end{aligned} \quad (29)$$

From the strong Wolfe condition (19) we have

$$c_2 g_{x_k}(\text{grad } h(x_k), \eta_k) \leq g_{x_{k+1}}(\text{grad } h(x_{k+1}), \text{D } R_{x_k}(\alpha_k \eta_k)[\eta_k]) \leq -c_2 g_{x_k}(\text{grad } h(x_k), \eta_k). \quad (30)$$

Hence from (29) and (30) we have

$$-1 + c_2 \frac{g_{x_k}(\text{grad } h(x_k), \eta_k)}{\|\text{grad } h(x_k)\|_{x_k}^2} \leq \frac{g_{x_{k+1}}(\text{grad } h(x_{k+1}), \eta_{k+1})}{\|\text{grad } h(x_{k+1})\|_{x_{k+1}}^2} \leq -1 - c_2 \frac{g_{x_k}(\text{grad } h(x_k), \eta_k)}{\|\text{grad } h(x_k)\|_{x_k}^2}.$$

And the result (28) follows from the induction hypothesis. \square

Theorem 4.5. For cost function h in (12), the Algorithm 2 with Fletcher-Reeves method generates iterates x_k such that

$$\liminf_{k \rightarrow \infty} \|\text{grad } h(x_k)\|_{x_k} = 0. \quad (31)$$

Proof. If $\text{grad } h(x_k) = 0$ for some $k = k_0$. Then $\text{grad } h(x_k) = 0$ for all $k \geq k_0$.

So we consider $\text{grad } h(x_k) \neq 0$ for all k . We shall prove (31) by contradiction. Suppose (31) does not hold. Then there exists a constant $c > 0$ such that

$$\|\text{grad } h(x_k)\|_{x_k} \geq c > 0, \quad \forall k \geq 0. \quad (32)$$

From (23) and (28) we have

$$\cos \theta_k \geq \frac{1 - 2c_2}{1 - c_2} \frac{\|\text{grad } h(x_k)\|_{x_k}}{\|\eta_k\|_{x_k}}. \quad (33)$$

It follows by Theorem 4.3 that the following series converges.

$$\sum_{k=0}^{\infty} \frac{\|\text{grad } h(x_k)\|_{x_k}^4}{\|\eta_k\|_{x_k}^2} < \infty. \quad (34)$$

For $k \geq 1$, the strong Wolfe condition (19) and (28) gives rise to

$$\left| g_{x_k} \left(\text{grad } h(x_k), \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1}) \right) \right| \leq -c_2 g_{x_{k-1}} \left(\text{grad } h(x_{k-1}), \eta_{k-1} \right) \leq \frac{c_2}{1-c_2} \|\text{grad } h(x_{k-1})\|_{x_{k-1}}^2.$$

Hence we have the following recurrence equation for $\|\eta_k\|_{x_k}^2$.

$$\begin{aligned} \|\eta_k\|_{x_k}^2 &= \left\| -\text{grad } h(x_k) + \beta_k \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1}) \right\|_{x_k}^2 \\ &\leq \left\| \text{grad } h(x_k) \right\|_{x_k}^2 + 2\beta_k \left| g_{x_k} \left(\text{grad } h(x_k), \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1}) \right) \right| + \beta_k^2 \left\| \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1}) \right\|_{x_k}^2 \\ &\leq \left\| \text{grad } h(x_k) \right\|_{x_k}^2 + \frac{2c_2}{1-c_2} \beta_k \left\| \text{grad } h(x_{k-1}) \right\|_{x_{k-1}}^2 + \beta_k^2 \left\| \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1}) \right\|_{x_k}^2 \\ &= \left\| \text{grad } h(x_k) \right\|_{x_k}^2 + \frac{2c_2}{1-c_2} \left\| \text{grad } h(x_k) \right\|_{x_k}^2 + \beta_k^2 \left\| \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1}) \right\|_{x_k}^2 \\ &= \frac{1+c_2}{1-c_2} \left\| \text{grad } h(x_k) \right\|_{x_k}^2 + \beta_k^2 \left\| \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1}) \right\|_{x_k}^2. \end{aligned} \quad (35)$$

Recall that we use differentiated retraction as our vector transport:

$$\mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1}) = \text{D}R_{x_{k-1}}(\alpha_{k-1}\eta_{k-1})[\eta_{k-1}] = \text{D}\pi(\bar{x}_{k-1} + \alpha_{k-1}\bar{\eta}_{k-1}) \left[P_{\bar{x}_{k-1} + \alpha_{k-1}\bar{\eta}_{k-1}}^{\mathcal{H}}(\bar{\eta}_{k-1}) \right].$$

Hence

$$\begin{aligned} \left\| \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1}) \right\|_{x_k}^2 &= g_{x_k} \left(\mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1}), \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1}) \right) = g_{\bar{x}_k} \left(\overline{\mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1})}_{\bar{x}_k}, \overline{\mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1})}_{\bar{x}_k} \right) \\ &= g_{\bar{x}_k} \left(P_{\bar{x}_{k-1} + \alpha_{k-1}\bar{\eta}_{k-1}}^{\mathcal{H}}(\bar{\eta}_{k-1}), P_{\bar{x}_{k-1} + \alpha_{k-1}\bar{\eta}_{k-1}}^{\mathcal{H}}(\bar{\eta}_{k-1}) \right) = g_{\bar{x}_{k-1}}(\bar{\eta}_{k-1}, \bar{\eta}_{k-1}) = \|\eta_{k-1}\|_{x_{k-1}}^2. \end{aligned}$$

Hence (35) becomes the following recurrence formula for $\|\eta_k\|_{x_k}^2$.

$$\|\eta_k\|_{x_k}^2 \leq \frac{1+c_2}{1-c_2} \left\| \text{grad } h(x_k) \right\|_{x_k}^2 + \beta_k^2 \|\eta_{k-1}\|_{x_{k-1}}^2. \quad (36)$$

By recursively using (35) and recall the definition of β_k in Fletcher-Reeves method we obtain

$$\begin{aligned} \|\eta_k\|_{x_k}^2 &\leq \frac{1+c_2}{1-c_2} \left(\left\| \text{grad } h(x_k) \right\|_{x_k}^2 + \beta_k^2 \left\| \text{grad } h(x_{k-1}) \right\|_{x_{k-1}}^2 + \cdots + \beta_k^2 \beta_{k-1}^2 \cdots \beta_2^2 \left\| \text{grad } h(x_1) \right\|_{x_1}^2 \right) \\ &\quad + \beta_k^2 \beta_{k-1}^2 \cdots \beta_0^0 \|\eta_0\|_{x_0}^2 \\ &= \frac{1+c_2}{1-c_2} \left\| \text{grad } h(x_k) \right\|_{x_k}^4 \left(\left\| \text{grad } h(x_k) \right\|_{x_k}^{-2} + \left\| \text{grad } h(x_k) \right\|_{x_{k-1}}^{-2} + \cdots + \left\| \text{grad } h(x_k) \right\|_{x_1}^{-2} \right) \\ &\quad + \left\| \text{grad } h(x_k) \right\|_{x_k}^4 \left\| \text{grad } h(x_0) \right\|_{x_0}^{-2} \\ &< \frac{1+c_2}{1-c_2} \left\| \text{grad } h(x_k) \right\|_{x_k}^4 \sum_{j=0}^k \left\| \text{grad } h(x_j) \right\|_{x_j}^{-2} \leq \frac{1+c_2}{1-c_2} \left\| \text{grad } h(x_k) \right\|_{x_k}^4 \frac{k+1}{c^2}, \end{aligned}$$

where we have used the contradiction assumption (32) in the last inequality. (37) results in the divergence of the following series.

$$\sum_{k=0}^{\infty} \frac{\left\| \text{grad } h(x_k) \right\|_{x_k}^4}{\|\eta_k\|_{x_k}^2} \geq c^2 \frac{1-c_2}{1+c_2} \sum_{k=0}^{\infty} \frac{1}{k+1} = \infty. \quad (37)$$

This contradicts to (34) and hence we have completed the proof. \square

In general, it is more difficult to prove the convergence of the Riemannian PR_+ CG method. It is possible to extend the convergence proof of PR_+ CG method in [34] to Riemannian PR_+ CG method, but it is beyond the scope of this paper.

5. Coordinate Riemannian Gradient Descent (CRGD)

The orthogonalization-free methods are preferred for large scale problems. For much larger problems, the coordinate descent method is favored, since the full gradient can be too large to even store. For instance, the coordinate gradient descent method for finding leading eigenvalue in [16] is the coordinate descent method for minimizing (6) with rank $p = 1$. In this section, following the same Riemannian manifold notation as in previous sections, we show that the a Riemmanian coordinate descent method is also equivalent to the coordinate descent method for minimizing (6) with any rank $p > 0$, which is the generalization of the algorithm in [16].

In [35], a method called the tangent subspace descent method was proposed: this method generalized the block coordinate descent method to manifold settings. Instead of updating the full gradient at each iteration, the tangent direction in each update is a projected vector of the full Riemannian gradient to a subspace of the tangent space by some subspace selection rule P_k . In the specific case of $\mathbb{C}_*^{n \times p} / \mathcal{O}_p$ considered in this paper, this method is written as Algorithm 4 and we denote it as Coordinate Riemannian Gradient Descent (CRGD).

Since the horizontal lift of $\text{grad } h(x_k)$ is a n -by- p matrix, we can simply choose the subspace selection rule by cyclically selecting the N -column block of the n -by- p matrix $\text{grad } f(\bar{x}_k)$. Let M_k denote the mask that evaluates the k -th N -column block of a n -by- p matrix cyclically. That is, if Z is a n -by- p matrix, then

$$M_k(Z) = Z_{kN+1:(k+1)N,:} \quad (38)$$

where $Z_{kN+1:(k+1)N,:}$ denotes the N -by- p matrix that takes the $(kN + 1)$ -th to $(k + 1)N$ -th columns of Z . And the index that exceeds the matrix range is understood as modulo by the matrix size, namely, cyclically. Then our update to \bar{x}_k is written through the following

$$\bar{x}_{k+1} = \bar{R}_{\bar{x}_k}(\alpha M_k(\text{grad } f(\bar{x}_k))), \quad (39)$$

where α is a constant step size.

With the simple retraction as in Section 2.7, (39) simply reduces to

$$\bar{x}_{k+1} = \bar{x}_{k+1} - \alpha M_k(2(\bar{x}_k \bar{x}_k^* - A)\bar{x}_k). \quad (40)$$

Notice that (40) with $p = 1$ and $N = 1$ reduces to the coordinate descent method for the leading eigenvalue in [16]. In particular, if $p = 1$ and we set $N = 1$ and P_k in Algorithm 4 to be M_k , defined in (38), then Algorithm 4 is equivalent to Algorithm 2 in [16]. So the generalization of the method in [16] to top p eigenvalues can be equivalently written as (40) or (39), which is a Riemannian coordiante descent method.

To take the advantage of CRGD to solve large-scaled problems, one should implement it through compact implementation. That is, each update should only depend on the block size N and should be independent of the problem size n . In the case of eigenvalue problem, $f(\bar{x}) = \frac{1}{2} \|\bar{x} \bar{x}^* - A\|_F^2$. If we assume that A is a sparse matrix such that we can achieve $M_k(Av)$ in $O(N)$, then we can indeed achieve a compact implementation of CRGD as in Algorithm 5.

Algorithm 4 Coordinate Riemannian gradient descent (CRGD) on the quotient manifold $\mathbb{C}_*^{n \times p} / \mathcal{O}_p$ with metric g

Require: initial iterate $x_0 \in \mathbb{C}_*^{n \times p} / \mathcal{O}_p$, tolerance $\varepsilon > 0$, tangent vector $\xi_0 = -\text{grad } h(x_0)$, subspace selection rule P_k , $\delta_0 := P_0(\xi_0)$, stepsize $\alpha > 0$.

1: **for** $k = 0, 1, 2, \dots$ **do**

2: Obtain the new iterate by retraction

$$x_{k+1} = R_{x_k}(\alpha \delta_k)$$

3: Compute the projection of $\xi_{k+1} := -\text{grad } h(x_{k+1})$ to a subspace of $T_{x_{k+1}} \mathbb{C}_*^{n \times p} / \mathcal{O}_p$

$$\delta_{k+1} := P_{k+1}(\xi_{k+1})$$

4: Check for convergence

$$\text{if } \|\delta_{k+1}\| := \sqrt{g_{x_{k+1}}(\delta_{k+1}, \delta_{k+1})} < \varepsilon, \text{ then break}$$

5: **end for**

Algorithm 5 Compact implementation for cyclic coordinate Riemannian gradient descent on the quotient manifold $\mathbb{C}_*^{n \times p} / \mathcal{O}_p$ with metric g

Require: initial iterate $\bar{x}_0 \in \mathbb{C}_*^{n \times p}$, $\bar{\eta}_0 = -\text{grad } f(\bar{x}_0) \in \mathbb{C}^{n \times p}$, first N columns of $\bar{\eta}_0$: $\bar{\delta}_0 = \mathcal{M}_0(\bar{\eta}_0)$, $a_0 = x_0^* x_0$, $b_0 = \delta_0^* x_0$, $c_0 = \delta_0^* \delta_0$, stepsize $\alpha > 0$, $s_0 = a_0 + \alpha b_0 + \alpha b_0^* + \alpha^2 c_0$, tolerance $\varepsilon > 0$.

1: **for** $k = 0, 1, 2, \dots$ **do**

2: Obtain the new iterate by retraction

$$\bar{x}_{k+1} = \bar{R}_{\bar{x}_k}(\alpha \bar{\delta}_k) = \bar{x}_k + \alpha \bar{\delta}_k$$

3: Cyclically compute the next N columns of $\bar{\eta}_{k+1} = -\text{grad } f(\bar{x}_{k+1})$

$$\bar{\delta}_{k+1} := -2M_{k+1}(\bar{x}_k s_k) - 2\alpha M_{k+1}(\bar{\delta}_k s_k) + 2M_{k+1}(A\bar{x}_k) + 2\alpha M_{k+1}(A\bar{\delta}_k)$$

4: Check for convergence

$$\text{if } \|\bar{\delta}_{k+1}\| := \sqrt{g_{\bar{x}_{k+1}}(\bar{\delta}_{k+1}, \bar{\delta}_{k+1})} < \varepsilon, \text{ then break}$$

5: Compute and update $a_{k+1}, b_{k+1}, c_{k+1}$

$$a_{k+1} = a_k + \alpha \bar{x}_k^* \bar{\delta}_k + \alpha \bar{\delta}_k^* x_k + \alpha^2 \bar{\delta}_k^* \bar{\delta}_k$$

$$b_{k+1} = \bar{\delta}_{k+1}^* \bar{x}_{k+1}$$

$$c_{k+1} = \bar{\delta}_{k+1}^* \delta_{k+1}$$

6: Compute temporary variable $s_{k+1} \in \mathbb{C}^{p \times p}$

$$s_{k+1} = a_{k+1} + \alpha b_{k+1} + \alpha b_{k+1}^* + \alpha^2 c_{k+1}$$

7: **end for**

6. Numerical Experiments

The numerical performance of the simple CG methods (7) has been well studied in the literature, e.g., see [11] for a comparison with other orthogonalization-free methods. In general, the performance of (7) for solving (6) depends on the spectrum of the matrix A . For completeness, in this section we verify the numerical performance of the simple CG methods (7) on large matrices A .

6.1. Real symmetric PSD matrices

We consider two types of matrices A . The first type is a 2D Laplacian matrix, which has a nearly uniform eigenvalue gap for a few top eigenvalues. Consider the discretization of a 2D Poisson equation with homogeneous Dirichlet boundary conditions on $[0, 1] \times [0, 1]$ using m -by- m interior grid points. Then the matrix representing the Laplacian operator is a 2D Laplacian matrix A of size m^2 -by- m^2 given as

$$A = \frac{1}{\Delta x^2} K \otimes I_m + I_m \otimes \frac{1}{\Delta y^2} K, \quad (41)$$

where $\Delta x = \Delta y = \frac{1}{m+1}$ and K is a m -by- m tridiagonal matrix.

$$K = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & \ddots & \ddots & \ddots \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix} \quad (42)$$

The second type is constructed by eigenvalue decomposition $A = V \Lambda V^{-1}$ where eigenvectors V are given by discrete cosine transform. We assign Λ so that the eigenvalues λ_i have four types of distribution of eigenvalues, similar to the numerical experiments considered in [11] but with a much larger matrix size:

1. (random) $\lambda_i \sim |\mathcal{N}(0, 1)|$, where $\mathcal{N}(0, 1)$ is standard normal distribution.
2. (uniform) $\lambda_i = 1 - \frac{i-1}{n}$, $1 \leq i \leq r$.
3. (u-shape) $\lambda_1 = \frac{14}{16}, \lambda_2 = \frac{10}{16}, \lambda_3 = \frac{8}{16}, \lambda_4 = \frac{7}{16}, \lambda_5 = \frac{5}{16}, \lambda_i = \frac{1}{16}$.
4. (logarithm) $\lambda_i = \frac{2^{1+\lfloor \log_2 n \rfloor}}{n} \frac{1}{2^i}$, $1 \leq i \leq r$.

The comparison of simple CG methods (7) with the TriOFM method in [20] for a 2D discrete Laplacian matrix is shown in Figure 1.

The comparison is shown for randomly distributed eigenvalues in Figure 2, uniformly distributed eigenvalues in Figure 3, U-shape distribution of eigenvalues in Figure 4, and log distribution of eigenvalues in Figure 5. Notice that the simple CG-PR method is much less efficient than the TriOFM method for the log distribution of eigenvalues. However, this slowness is due to the eigenvalue gap between σ_p and σ_{p+1} . In Figure 6, the top p eigenvalues with $p = 5$ have a log distribution but the gap between σ_p and σ_{p+1} is enlarged by shifting the top p eigenvalues from the same matrix in Figure 5, and we observe that the simple CG-PR method is efficient in this scenario. In other words, the matrix in Figure 5 has eigen values $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$, and the matrix in Figure 6 has eigenvalues $\lambda_1 + C \geq \lambda_2 + C \geq \dots \geq \lambda_{p+1} \geq \dots \geq \lambda_n$.

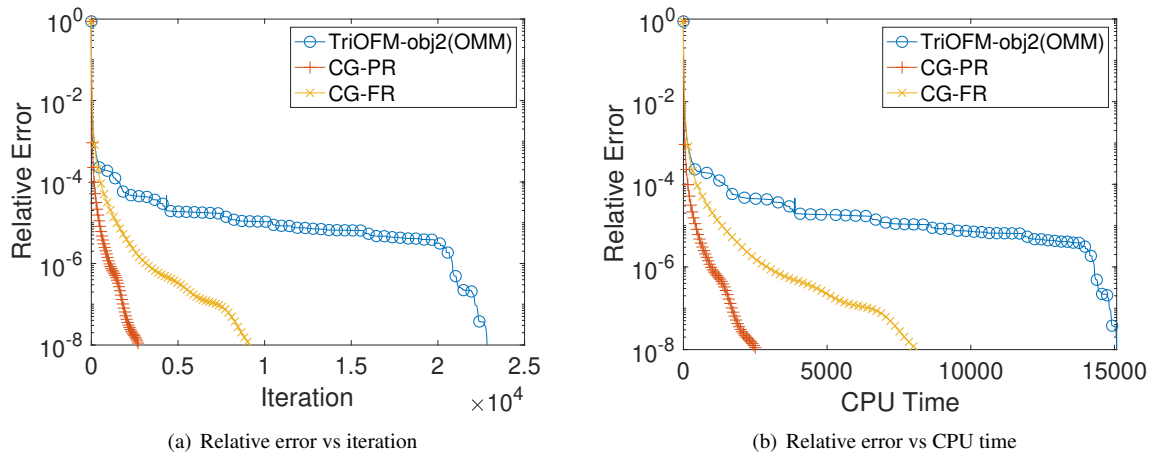


Figure 1. Comparison for computing the top-10 eigenvalues of a 2D Laplacian matrix of size $10^6 \times 10^6$.

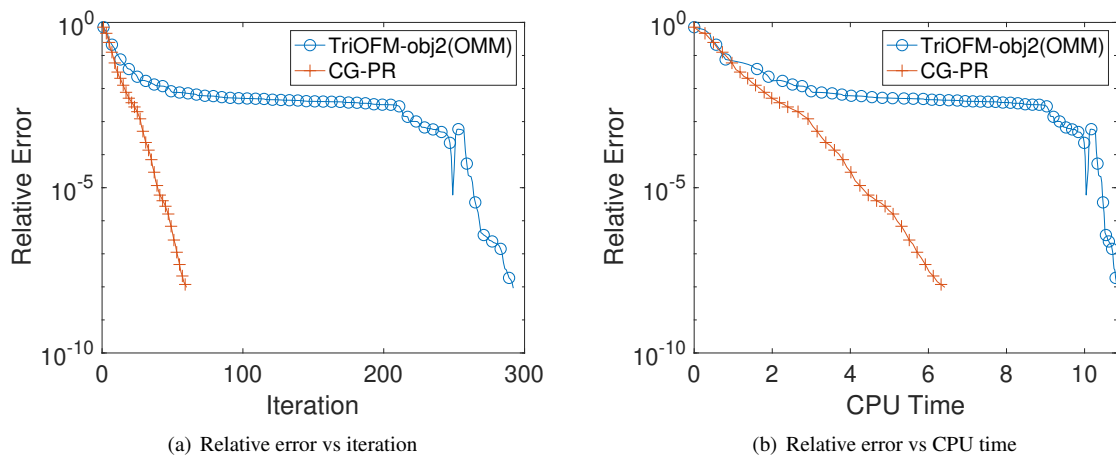


Figure 2. Comparison for computing the top-10-eigenvalue problem of a 10^4 -by- 10^4 matrix with randomly distributed eigenvalues.

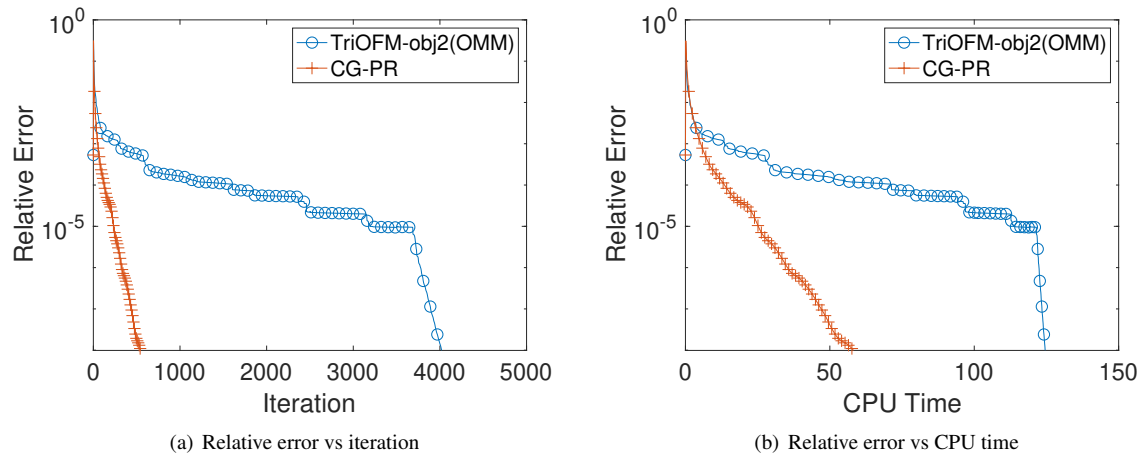


Figure 3. Comparison for computing the top-10-eigenvalue problem of a 10^4 -by- 10^4 matrix with uniformly distributed eigenvalues.

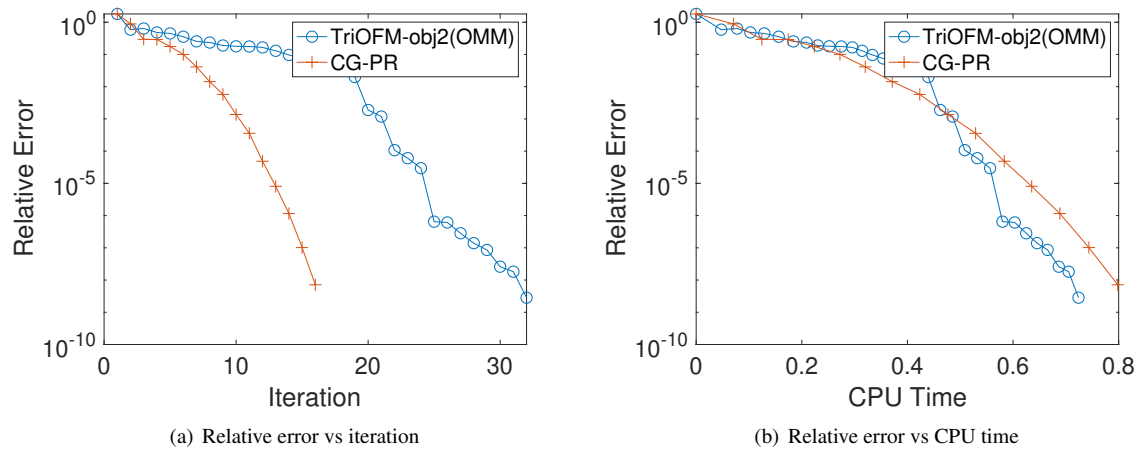


Figure 4. Comparison for computing the top-10-eigenvalue problem of a 10^4 -by- 10^4 matrix with U-shape distributed eigenvalues.

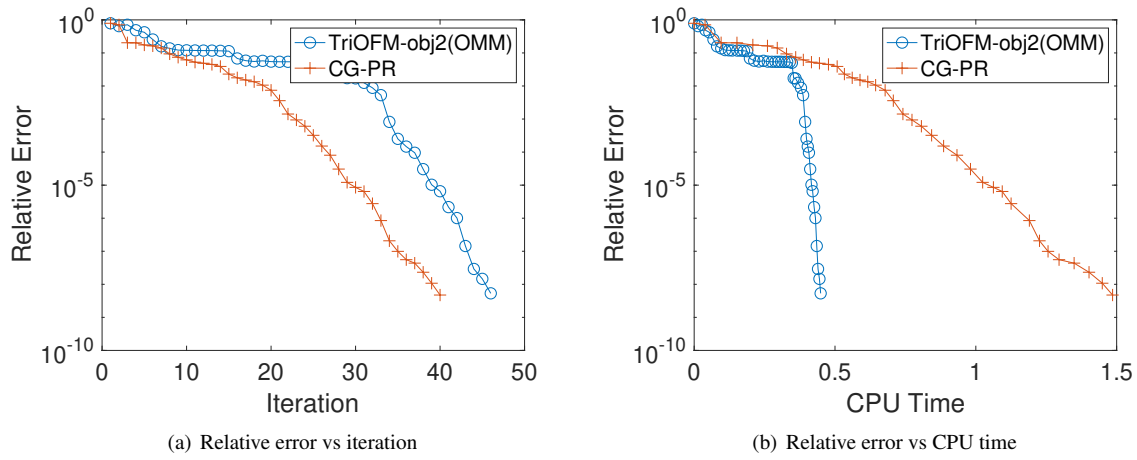


Figure 5. Comparison for computing the top-5-eigenvalue problem of a 10^4 -by- 10^4 matrix with logarithm distributed eigenvalues.

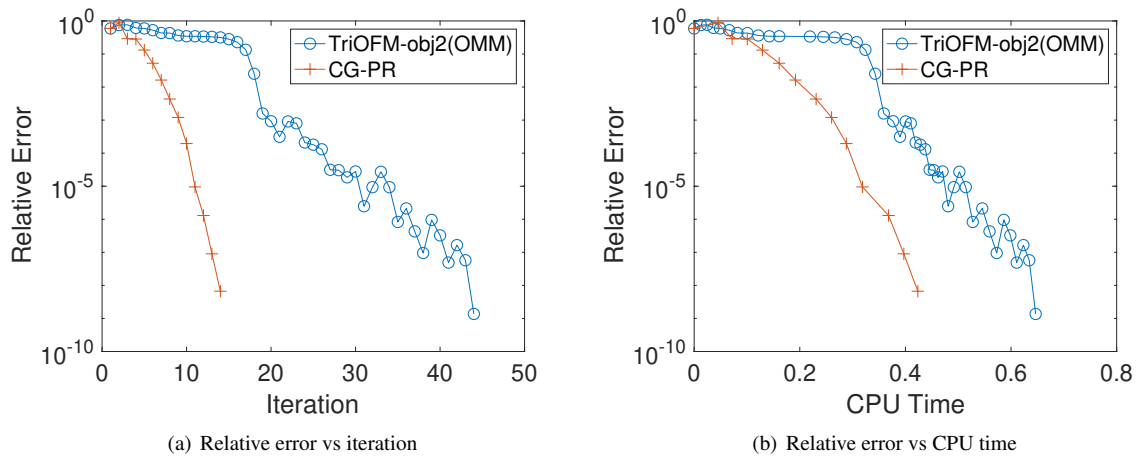


Figure 6. Comparison for computing the top-5-eigenvalue problem of a 10^4 -by- 10^4 matrix with eigenvalues $\lambda_1 + C \geq \lambda_2 + C \geq \lambda_5 + C \geq \lambda_{5+1} \geq \dots \geq \lambda_n$, where $C = \lambda_1$ and $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ has a log distribution.

6.2. Hermitian PSD matrices

It is shown in [12] that Algorithm 2 can be used for finding the top eigenvalues of a Hermitian PSD matrix. We test Algorithm 2 on 6 for a matrix A with eigenvectors defined by 2D Fast Fourier Transform. Namely, the linear operator of applying A to a 2D array u is defined by

$$Au = \text{fft2}(\Sigma .* \text{fft2}(u)),$$

where $.*$ denotes the entrywise product and Σ is a 2D array consisting of nonnegative eigenvalues of A .

The performance of the CG-PR method is shown in Figure 7 for four kinds of eigenvalue distributions in such a Hermitian PSD matrix.

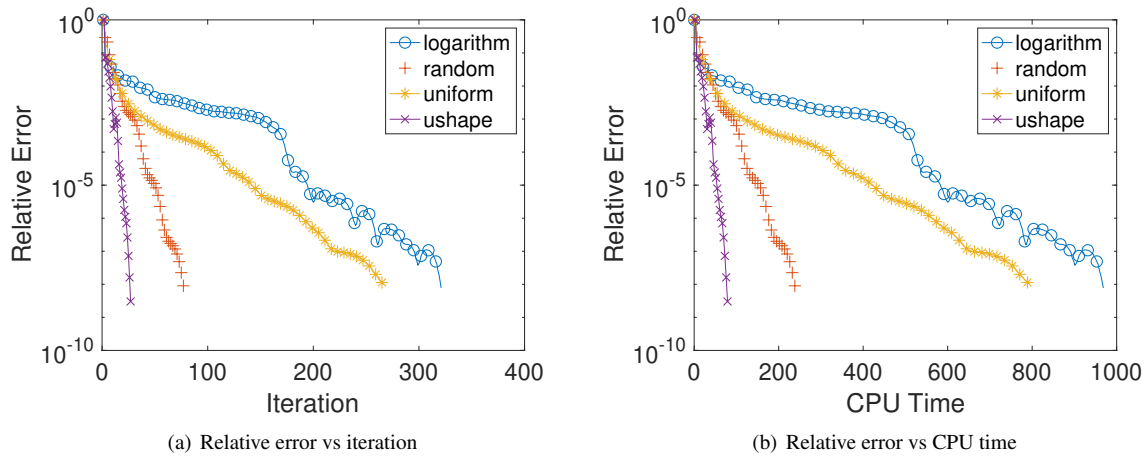


Figure 7. The CG-PR method for the top-10-eigenvalue problem with rank-1000 Hermitian matrices of 10^6 -by- 10^6 with different distributions of eigenvalues.

6.3. Smallest eigenvalues

6.3.1. Inverse 2D Laplacian matrix

One technique to find the smallest eigenvalues of a given invertible matrix A is through the shift-and-inverse method. That is, to find the largest eigenvalues of $(A + \mu I)^{-1}$, where $\mu > 0$ is a shift constant such that $A + \mu I$ becomes positive definite. We use this method to find the smallest eigenvalues of the 2D Laplacian matrix A as in (41).

Notice that the top eigenvalues of A^{-1} almost follow a logarithm distribution. Based on our observation, we can choose μ appropriately to make the top eigenvalues of $(A + \mu I)^{-1}$ have a uniform distribution to accelerate the convergence of the CG method. Since we know the true eigenvalues of A , we shift it by choosing μ to be the smallest desired eigenvalue. That is, suppose the smallest r eigenvalues of A is $\sigma_1 \leq \sigma_2 \leq \dots \leq \sigma_r$. Then we choose $\mu = \sigma_1$. As a result the top eigenvalues of $(A + \mu I)^{-1}$ would be $\frac{1}{\sigma_1 + \sigma_1} \geq \frac{1}{\sigma_2 + \sigma_1} \geq \dots \geq \frac{1}{\sigma_r + \sigma_1}$ that almost follows a uniform distribution. A fast matrix inversion is implemented by using the eigendecomposition of the matrix. The performance is shown in Figure 8 and Figure 9.

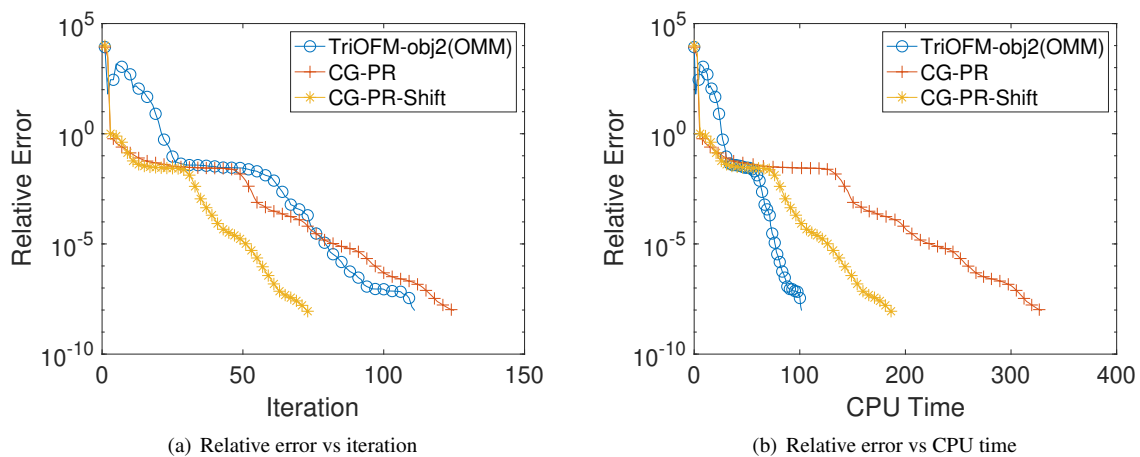
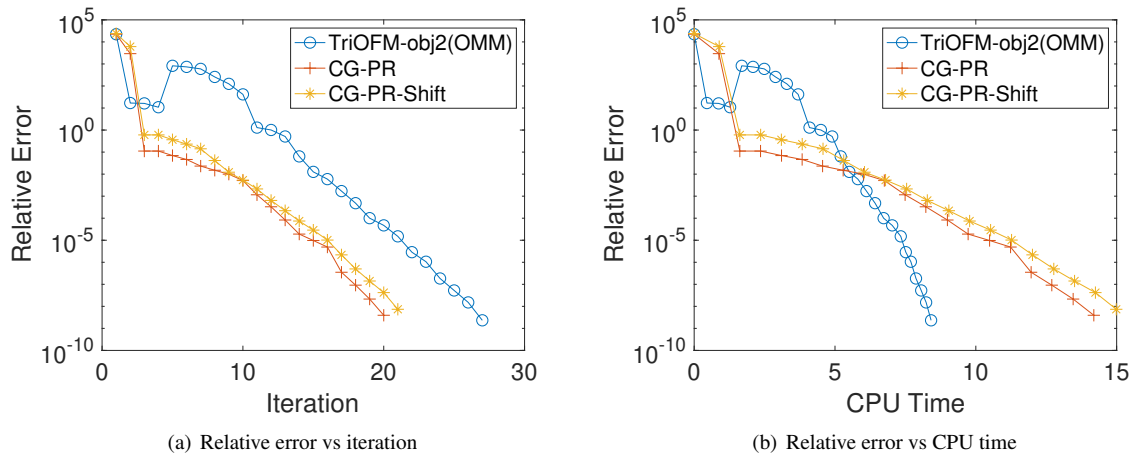


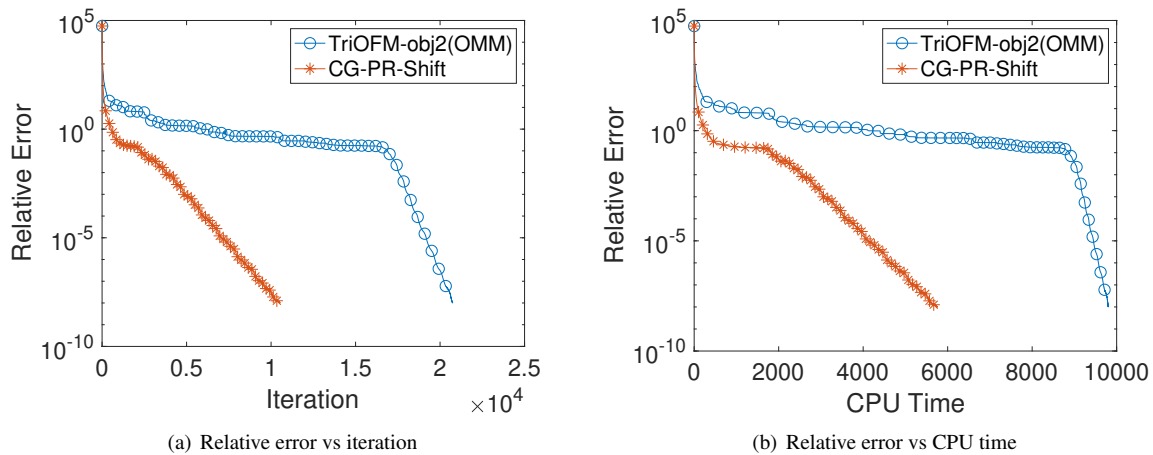
Figure 8. The shift-and-inverse method on the smallest-10-eigenvalue problem of a 10^6 -by- 10^6 2D-Laplacian matrix.

Figure 9. The shift-and-inverse method on the smallest-3-eigenvalue problem of a 10^6 -by- 10^6 2D-Laplacian matrix.

6.3.2. Negative 2D Laplacian matrix

Another way to find the smallest eigenvalues of a given matrix A is through the negative-shift method. That is, to consider finding the largest eigenvalues of $\mu I - A$, where $\mu > 0$ is a shift constant such that $\mu I - A$ is positive semi-definite. We use this method to find the smallest eigenvalues of the 2D Laplacian matrix defined in (41).

Notice we need to shift at least the largest eigenvalue of A to ensure that $\mu I - A$ is PSD. And once we find the top eigenvalues of $\mu I - A$ we need to shift back and extract the smallest eigenvalues of A by computing $\mu - (\mu - \sigma)$, where σ 's are the smallest eigenvalues of A . Hence when the condition number of A is bad, i.e., if $\mu \gg \sigma$, then we might lose a significant number of digits of accuracy for computing $\mu - (\mu - \sigma)$. In our numerical tests, we did not encounter this numerical accuracy issue. The performance is shown in Figure 10. Notice that the negative-shift method is much slower than the shift-and-inverse method, because of the different distributions of the largest eigenvalues of $\mu I - A$ and $(A + \mu I)^{-1}$.

Figure 10. The negative-shift method on the smallest-10-eigenvalue problem of a 10^6 -by- 10^6 2D-Laplacian matrix.

6.4. Negative 3D Laplacian matrix

We repeat the same test as in previous subsection for a larger problem of finding the smallest eigenvalues of a 3D discrete Laplacian on a 500^3 grid, which corresponds to a matrix of size $1.25E8 \times 1.25E8$. We implement both the simple CG method (7) and TriOFM method on a Nvidia GPU A100 80G.

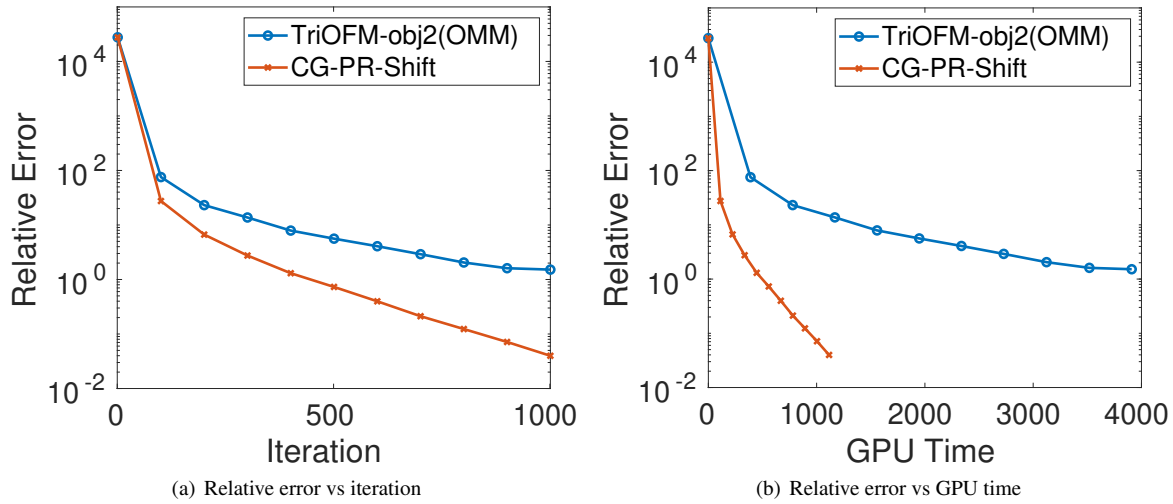


Figure 11. The shift-and-inverse method on the smallest-3-eigenvalue problem of a 3D-Laplacian matrix on a 500^3 grid. The matrix size is $1.25E8 \times 1.25E8$. Computation was done on Nvidia GPU A100 80G.

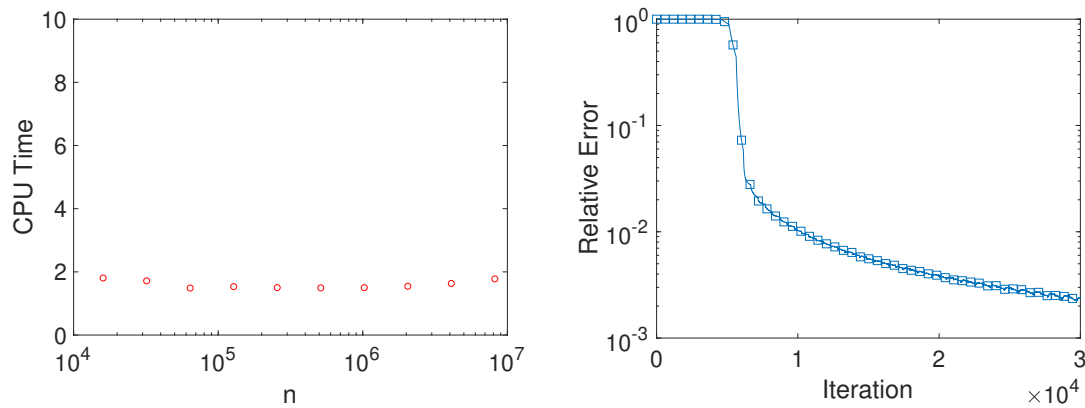
6.5. Coordinate Riemannian gradient descent

We consider applying the coordinate Riemannian gradient descent method described in Section 5 to a 1D Laplacian matrix of size n -by- n given by $A = \frac{1}{\Delta x^2} K$, where $\Delta x = \frac{1}{n+1}$ and K is the tridiagonal matrix defined in (42). This example is only for the demonstration purpose of the coordinate gradient descent method. Choosing this simple A makes it easy for the compact implementation of the matrix-vector multiplication of Au . One can also apply this method to any sparse matrix A as long as one has the compact implementation of $M_k(Au)$ in $O(N)$, where N is a constant independent of the problem size n .

As we can see from Figure 12, the CPU time for running the first 3000 iterations is independent of problem size. This demonstrated the $O(1)$ computational complexity of the coordinate Riemannian gradient descent method for leading eigenpairs.

7. Conclusions

In this paper we have studied the orthogonalization-free method to find leading eigenpairs of a positive semi-definite Hermitian matrix via an unconstrained Burer-Monteiro formulation. For this optimization problem, we have shown the equivalence between the nonlinear conjugate gradient method and a Riemannian conjugate gradient method on a quotient manifold with the Bures-Wasserstein metric, leading to a new understanding of the global convergence of the nonlinear conjugate gradient method in Burer-Monteiro formulation to a stationary point. We have also shown that the simple coordinate descent method in Burer-Monteiro formulation is equivalent to a coordinate Riemannian gradient descent method. Numerical tests on large scale matrices have verified the numerical performance of the simple conjugate gradient method in Burer-Monteiro formulation for computing leading eigen-pairs, which is consistent with findings in the literature.



(a) CPU time of the first 3000 iterations vs problem size $n = 100 * 2^k$ for k goes from 4 to 13. Each iteration cyclically updates $N = 100$ columns with constant step size 10^{-10} . (b) Relative error vs iteration. Problem size $n = 100 * 2^9$. Each iteration cyclically updates $N = 100$ columns with constant step size 10^{-10} .

Figure 12. Coordinate Riemannian gradient descent for solving the top-10 eigenvalues of a Laplacian matrix.

Acknowledgement

S. Zheng and X. Zhang are supported by NSF DMS-2208518. H. Yang thanks Oracle Labs, part of Oracle America, Inc., for providing funding that supported research in the area of leading eigenvalue problems. The authors are grateful to Yingzhou Li for providing the MATLAB code of TirOFM.

References

- [1] J. Shi, J. Malik, Normalized cuts and image segmentation, *IEEE Transactions on pattern analysis and machine intelligence* 22 (8) (2000) 888–905.
- [2] J. Cheeger, A lower bound for the smallest eigenvalue of the laplacian, in: *Problems in analysis*, Princeton University Press, 2015, pp. 195–200.
- [3] W. E. Donath, A. J. Hoffman, Algorithms for partitioning of graphs and computer logic based on eigenvectors of connection matrices, *IBM Technical Disclosure Bulletin* 15 (3) (1972) 938–944.
- [4] M. Fiedler, Algebraic connectivity of graphs, *Czechoslovak mathematical journal* 23 (2) (1973) 298–305.
- [5] J. Lu, H. Yang, Preconditioning orbital minimization method for planewave discretization, *Multiscale Modeling & Simulation* 15 (1) (2017) 254–273.
- [6] J. Lu, H. Yang, A cubic scaling algorithm for excited states calculations in particle–particle random phase approximation, *Journal of Computational Physics* 340 (2017) 297–308.
- [7] Z. Wang, Y. Li, J. Lu, Coordinate descent full configuration interaction, *Journal of chemical theory and computation* 15 (6) (2019) 3558–3569.
- [8] Q. Pang, H. Yang, A distributed block chebyshev-davidson algorithm for parallel spectral clustering, *arXiv preprint arXiv:2212.04443*.
- [9] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, S. W. Zucker, Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps, *Proceedings of the national academy of sciences* 102 (21) (2005) 7426–7431.
- [10] F. Corsetti, The orbital minimization method for electronic structure calculations with finite-range atomic basis sets, *Computer Physics Communications* 185 (3) (2014) 873–883.
- [11] W. Gao, Y. Li, B. Lu, Triangularized orthogonalization-free method for solving extreme eigenvalue problems, *Journal of Scientific Computing* 93 (3) (2022) 1–28.
- [12] S. Zheng, W. Huang, B. Vandereycken, X. Zhang, Riemannian optimization using three different metrics for hermitian psd fixed-rank constraints: an extended version (2023).
- [13] J. Nocedal, S. J. Wright, *Numerical optimization*, Springer, 1999.
- [14] X. Liu, Z. Wen, Y. Zhang, An efficient Gauss–Newton algorithm for symmetric low-rank product matrix approximations, *SIAM Journal on Optimization* 25 (3) (2015) 1571–1608.
- [15] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, M. I. Jordan, How to escape saddle points efficiently, in: *International Conference on Machine Learning*, PMLR, 2017, pp. 1724–1732.
- [16] Y. Li, J. Lu, Z. Wang, Coordinatewise descent methods for leading eigenvalue problem, *SIAM Journal on Scientific Computing* 41 (4) (2019) A2681–A2716.
- [17] E. Massart, J. M. Hendrickx, P.-A. Absil, Curvature of the manifold of fixed-rank positive-semidefinite matrices endowed with the Bures–Wasserstein metric, in: *Geometric Science of Information: 4th International Conference, GSI 2019, Toulouse, France, August 27–29, 2019, Proc.*, Springer, 2019, pp. 739–748.

- [18] H. Sato, T. Iwai, A new, globally convergent Riemannian conjugate gradient method, *Optimization* 64 (4) (2015) 1011–1031.
- [19] J. D. Lee, I. Panageas, G. Piliouras, M. Simchowitz, M. I. Jordan, B. Recht, First-order methods almost always avoid strict saddle points, *Mathematical programming* 176 (1) (2019) 311–337.
- [20] W. Gao, Y. Li, B. Lu, Global Convergence of Triangularized Orthogonalization-free Method, arXiv preprint arXiv:2110.06212.
- [21] A. V. Knyazev, Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method, *SIAM Journal on Scientific Computing* 23 (2) (2001) 517–541.
- [22] Y. Zhou, Y. Saad, M. L. Tiago, J. R. Chelikowsky, Self-consistent-field calculations using chebyshev-filtered subspace iteration, *Journal of Computational Physics* 219 (1) (2006) 172–184.
- [23] K. Neymeyr, A geometric theory for preconditioned inverse iteration IV: On the fastest convergence cases, *Linear Algebra and its Applications* 415 (1) (2006) 114–139, special Issue on Large Scale Linear and Nonlinear Eigenvalue Problems.
- [24] E. S. Coakley, V. Rokhlin, A fast divide-and-conquer algorithm for computing the spectra of real symmetric tridiagonal matrices, *Applied and Computational Harmonic Analysis* 34 (3) (2013) 379–414.
- [25] H. M. Aktulga, L. Lin, C. Haine, E. G. Ng, C. Yang, Parallel eigenvalue calculation based on multiple shift–invert lanczos and contour integral based spectral projection method, *Parallel Computing* 40 (7) (2014) 195–212, 7th Workshop on Parallel Matrix Algorithms and Applications.
- [26] R. Li, Y. Xi, E. Vecharynski, C. Yang, Y. Saad, A thick-restart lanczos algorithm with polynomial filtering for hermitian eigenvalue problems, *SIAM Journal on Scientific Computing* 38 (4) (2016) A2512–A2534.
- [27] E. Polizzi, Density-matrix-based algorithm for solving eigenvalue problems, *Physical Review B* 79 (2009) 115112.
- [28] T. Sakurai, H. Tadano, CIRR: a Rayleigh-Ritz type method with contour integral for generalized eigenvalue problems, *Hokkaido Mathematical Journal* 36 (4) (2007) 745 – 757.
- [29] Y. Xi, Y. Saad, Computing partial spectra with least-squares rational filters, *SIAM Journal on Scientific Computing* 38 (5) (2016) A3020–A3045.
- [30] X. Ye, J. Xia, R. H. Chan, S. Cauley, V. Balakrishnan, A Fast Contour-Integral Eigensolver for Non-Hermitian Matrices, *SIAM Journal on Matrix Analysis and Applications* 38 (4) (2017) 1268–1297.
- [31] J. M. Lee, *Introduction to Smooth Manifolds*, Vol. 218 of Graduate Texts in Mathematics, Springer New York, New York, NY, 2012.
- [32] P.-A. Absil, R. Mahony, R. Sepulchre, *Optimization algorithms on matrix manifolds*, Princeton University Press, Princeton, N.J. ; Woodstock, 2008, oCLC: ocn174129993.
- [33] E. Massart, P.-A. Absil, Quotient Geometry with Simple Geodesics for the Manifold of Fixed-Rank Positive-Semidefinite Matrices, *SIAM Journal on Matrix Analysis and Applications* 41 (1) (2020) 171–198.
- [34] J. C. Gilbert, J. Nocedal, Global convergence properties of conjugate gradient methods for optimization, *SIAM Journal on optimization* 2 (1) (1992) 21–42.
- [35] D. H. Gutman, N. Ho-Nguyen, Coordinate descent without coordinates: Tangent subspace descent on riemannian manifolds, *Mathematics of Operations Research* 48 (1) (2022) 127–159.