

# Triangularized Orthogonalization-free Method for Solving Extreme Eigenvalue Problems

Weiguo Gao<sup>†‡§</sup>, Yingzhou Li<sup>#</sup>, Bichen Lu<sup>§†</sup>

<sup>†</sup> School of Mathematical Sciences, Fudan University

<sup>‡</sup> School of Data Science, Fudan University

<sup>§</sup> Shanghai Center for Mathematical Sciences

<sup>#</sup> Department of Mathematics, Duke University

May 26, 2020

## Abstract

A novel orthogonalization-free method together with two specific algorithms are proposed to solve extreme eigenvalue problems. On top of gradient-based algorithms, the proposed algorithms modify the multi-column gradient such that earlier columns are decoupled from later ones. Global convergence to eigenvectors instead of eigenspace is guaranteed almost surely. Locally, algorithms converge linearly with convergence rate depending on eigengaps. Momentum acceleration, exact linesearch, and column locking are incorporated to further accelerate both algorithms and reduce their computational costs. We demonstrate the efficiency of both algorithms on several random matrices with different spectrum distribution and matrices from practice.

**Keywords.** eigenvalue problem; orthogonalization-free; iterative eigensolver; full configuration interaction;

## 1 Introduction

This paper proposes a novel triangularized orthogonalization-free method (TriOFM) for solving extreme eigenvalue problems. Given a symmetric matrix  $A$ , the extreme eigenvalue problem is defined as,

$$AX = X\Lambda, \tag{1}$$

where  $A \in \mathbb{R}^{n \times n}$ ,  $A^T = A$ ,  $\Lambda \in \mathbb{R}^{p \times p}$  is a diagonal matrix with  $A$ 's smallest (largest)  $p$  eigenvalues being its diagonal entries in ascending (descending) order, and the columns of  $X$  are the corresponding eigenvectors. The proposed methods are targeting some specific applications in computational physics and computational chemistry, in which areas smallest eigenpairs are desired as the ground-state and low-lying excited-states. Hence, in the following, we describe and analyze methods for smallest  $p$  eigenpairs. All eigensolvers in this paper can be applied to a linearly transformed  $A$  to obtain either its smallest or its largest  $p$  eigenpairs.

---

\*Authors are listed in alphabetical order.

Solving extreme eigenvalue problem is the fundamental computational step in a wide range of applications, including but not limited to principle component analysis, dimension reductions, electronic structure calculation, quantum many-body problems, etc. In this paper, we specifically concern extreme eigenvalue problems with two properties:

- (i) Orthogonalization of  $X$  is not permitted;
- (ii) Eigenvectors are sparse vectors.

At least two important applications, linear scaling density functional theory (DFT) [24] and full configuration interaction (FCI) [13] for low-lying excited states from electronic structure calculation, admit these two properties. In DFT with spacial local basis sets, the eigenvectors are sparse in general, which are also known as Wannier functions [3, 27]. Hence, the second property holds in linear scaling DFT. Regarding the first property, since the number of desired eigenpairs is on the same order as the problem size in DFT, any orthogonalization on  $X$  would make the computational complexity go beyond linear scaling. Hence the first property is needed in the algorithm design. FCI for low-lying excited states is very different from DFT problems. In FCI,  $p$  is usually a small constant, e.g.,  $p = 10$ . However, the size of the matrix  $n$  grows factorially as the number of electrons and orbitals in the system increases. Luckily, the eigenvectors of the low-lying excited states are extremely sparse, which makes the problem tackleable. Coordinate descent algorithm is applied to the FCI problem to reveal the sparse pattern efficiently [29]. Orthogonalization of  $X$ , however, is incompatible with the coordinate descent algorithm. Hence it is not permitted. For both applications, the sparsity of the eigenvectors is the key feature to preserve. For algorithms converging to the eigenspace instead of eigenvectors directly would destroy the sparsity and are not favored by these applications. Hence, we need an orthogonalization-free method that converges to eigenvectors directly.

## 1.1 Related Work

For linear eigenvalue problems as (1), there are many classical eigensolvers from textbooks of numerical linear algebra. Readers are referred to [11] for references. In electronic structure calculation, variants of classical eigensolvers, like Davidson [7], locally optimal block preconditioned conjugate gradient method (LOBPCG) [14], projected preconditioned conjugate gradient (PPCG) [28], Chebyshev filtering [1, 35], pole expansion [20, 26], are widely used in the self consistent field iteration in DFT. All these methods are related to Krylov subspace methods to certain degree. A recent software ELSI [33, 32] provides an interface to many of these eigensolvers for DFT calculation.

Besides Krylov subspace methods, another family of methods view the eigenvalue problem as a constrained optimization problem and solve it using either first-order or second-order optimization methods [6, 9, 12, 31, 34]. These methods usually targeting more general objective functions with orthonormal constraint. But the linear eigenvalue problem is always one of their potential applications. Since the feasible domain of the orthonormality constraint is known as Stiefel manifold, these methods are also known as manifold optimization methods. They take either Euclidean gradient or Riemannian gradient step with certain strategies in calculating the stepsize. A retraction or projection step is needed to maintain the feasibility of the iterator  $X$ . Recently, in order to enhance the parallelizability, the retraction step is avoided through either augmented Lagrangian method [10, 30] or extend gradient [5].

Linear eigenvalue problems can also be written as an unconstrained optimization problem. The most well-known one is minimizing the Rayleigh quotient, which can be extended to multicolumn

case as well. Another two unconstrained objective functions are

$$\min_{X \in \mathbb{R}^{n \times p}} \|A + XX^\top\|_F^2, \tag{Obj1}$$

and

$$\min_{X \in \mathbb{R}^{n \times p}} \text{tr}((2I - X^\top X)X^\top AX), \tag{Obj2}$$

where  $\|\cdot\|_F$  denotes the Frobenius norm and  $\text{tr}(\cdot)$  denotes the trace operation. They are processors of the TriOFM proposed in this paper. (Obj1) has been adopted to address the extreme eigenvalue problems arising from several areas [16, 19, 21], including FCI [29, 18]. (Obj2) is widely known in the orbital minimization method (OMM) [4, 22, 23, 25, 24], which is popular in the area of (linear scaling) DFT. More details about (Obj1) and (Obj2) are deferred to Section 2.

For all methods afore mentioned in this section, some of them are orthogonalization-free, and some of them converge to eigenvectors directly. But none of them holds two properties simultaneously.

## 1.2 Contribution

In this paper, a novel iterative framework, named triangularized orthogonalization-free method (TriOFM), is proposed, which has orthogonalization-free property and converges to eigenvectors directly. The framework is inspired by the unconstrained optimization methods while the updating direction is no longer a gradient of any energy functional. Instead of viewing the iterative framework as an optimization method, it is more proper to view it as a discrete dynamical system or discrete time flow in a vector field. Under the novel iterative framework, locking technique can be activated, which is of important practical usage but not available in other orthogonalization-free methods. Two iterative schemes are proposed under TriOFM framework for (Obj1) and (Obj2), namely TriOFM-(Obj1) and TriOFM-(Obj2).

The convergence analysis of TriOFM-(Obj1) is carried out in detail. The global convergence is proved without convergence rate. And the key in the global convergence proof, not surprisingly, is the stable manifold theorem. Then we also provide local convergence analysis with convergence rate, where the rate is carried out through a careful analysis on the accumulated error term. All these analyses are carried over to TriOFM-(Obj2) without detailed proof.

After the analysis, we propose a few algorithmic strategies to further accelerate the convergence and reduce the computational cost. Conjugate gradient direction and a few line search strategies are suggested to accelerate both iterative schemes. Locking technique is incorporated to reduce the computational cost.

Numerical examples are provided to demonstrate the effectiveness of the novel framework. All suggested algorithmic strategies are first tested on randomly generated matrices and then applied to practical examples, one from DFT and another one from FCI. In both practical examples, we observe that the proposed framework achieves both the orthogonalization-free and converging to eigenvectors properties while not losing much efficiency comparing with their original versions (gradient-based versions).

## 1.3 Organization

Section 2 provides detail introductions to both (Obj1) and (Obj2) with an analysis on the energy landscape of the multicolumn case. Section 3 introduces the novel iterative framework and its two iterative schemes in detail. The global and local convergence analysis are carried out in Section 4 and Section 5 respectively. Algorithmic strategies are suggested in Section 6. In Section 7, all iterative

schemes and algorithmic strategies are numerically tested on random matrices and matrices from practice. Finally, Section 8 concludes the paper with discussion on future directions.

## 2 Preliminary

We introduce optimization based eigensolvers using (Obj1) and (Obj2) in detail in this section. These eigensolvers are closely related to our proposed methods. Notations used throughout the paper is summarized in Table 1, which would be used without further explanation.

Notation	Explanation
$n$	The size of the matrix.
$p$	The number of desired eigenpairs.
$q$	The number of negative eigenvalues of the matrix.
$A$	The $n$ -by- $n$ symmetric matrix.
$\Lambda$	A diagonal matrix with diagonal entries being eigenvalues of $A$ in increasing ordering.
$\lambda_i$	The $i$ -th smallest eigenvalue of $A$ .
$\Lambda_i$	The first $i$ -by- $i$ principal submatrix of $\Lambda$ .
$U$	The orthogonal matrix denoting the eigenvectors of $A$ .
$u_i$	The eigenvector of $A$ corresponding to $\lambda_i$ .
$U_i$	The first $i$ columns of $U$ .
$\rho$	The two norm of $A$ , <i>i.e.</i> , $\rho = \ A\ _2$ .
$X^{(t)}$	An $n$ -by- $p$ matrix denoting the vectors at $t$ -th iteration.
$x_i^{(t)}$	The $i$ -th column of $X^{(t)}$ .
$X_i^{(t)}$	The first $i$ columns of $X^{(t)}$ , <i>i.e.</i> $X_i^{(t)} = \begin{pmatrix} x_1^{(t)} & x_2^{(t)} & \dots & x_i^{(t)} \end{pmatrix}$ .
$f(X)$	The objective function of $X$ , either (Obj1) or (Obj2).
$f_1(X), f_2(X)$	The objective function of (Obj1) and (Obj2) respectively.
$\nabla f(X)$	The gradient of $f(X)$ .
$\nabla f_1(X), \nabla f_2(X)$	The gradient of $f_1(X), f_2(X)$ .
$\alpha$	The stepsize used in optimization algorithms.
$e_i$	The $i$ -th standard basis, <i>i.e.</i> , a vector of length $n$ with one on the $i$ -th entry and zero elsewhere.

Table 1: Notations used throughout the paper.

As mentioned in Section 1, there have already been many well-known methods for solving linear symmetric eigenvalue problems, including the traditional ones like power method, QR iteration, Lanczos, etc. All these methods can be viewed as solvers for the constrained optimization problem,

$$\min_{\substack{X \in \mathbb{R}^{n \times p} \\ X^T X = I}} \text{tr}(X^T A X). \quad (2)$$

Hence a projection back onto the constraint space is needed after each iteration or after a few iterations. Such a projection is necessary to guarantee that different columns of  $X$  convergence to different eigenvectors, but is difficult to parallelize efficiently. Unconstrained optimization based

eigsolvers, in contrast to those traditional ones, require matrix-matrix product without projection. Hence the parallel communication cost is significantly reduced and the parallel efficiency is preserved, which is plausible for solving large problems on massive parallel environment.

The most well-known unconstrained optimization based eigensolver for a single eigenpair is to minimize the Rayleigh quotient, whose multicolumn extension is

$$\min_{X \in \mathbb{R}^{n \times p}} r(X) = \text{tr} \left( (X^\top X)^{-1} X^\top A X \right). \quad (3)$$

The minimizer of (3) is any point in the eigenspace spanned by  $\{u_1, u_2, \dots, u_p\}$ . However, if we minimize (3) with gradient based optimization methods, the gradient at each iteration requires solving a matrix inverse  $(X^\top X)^{-1}$ , which is computation-wise expensive and lacks parallel efficiency. (Obj1) and (Obj2) are two unconstrained optimization problems for multicolumn symmetric eigenvalue problems. More importantly, the gradient based iteration for both objective functions are inverse-free and projection-free.

## 2.1 Objective Function One

The intuition behind (Obj1) is simple. Since  $A$  is symmetric and we aim to compute the smallest negative eigenpairs,  $A + XX^\top$  is the residual of the symmetric low-rank approximation. Hence, (Obj1) minimizes the Frobenius norm square of the residual. (Obj1) has an equivalent trace form similar to (Obj1), *i.e.*,

$$\min_{X \in \mathbb{R}^{n \times p}} \|A + XX^\top\|_F^2 \Leftrightarrow \min_{X \in \mathbb{R}^{n \times p}} \text{tr} (2X^\top A X + X^\top X X^\top X). \quad (4)$$

With an extra penalty parameter, (Obj1) is also equivalent to a trace-penalty minimization model [21].

The gradient of (Obj1) using the trace form can be derived as

$$\nabla f_1(X) = 4AX + 4XX^\top X. \quad (5)$$

In both [19, 21], the energy landscape of (Obj1) has been analyzed using the properties of its gradient and Hessian. (Obj1) does not have any spurious local minimum. We summarize energy landscape analysis related theorems in [19, 21] as follows.

**Theorem 2.1.** *Assume that  $A$  has  $q$  negative eigenpairs with  $q \geq p$ . All **stationary points** of (Obj1) are of form  $X = U_q \sqrt{-\Lambda_q} P S Q$  and all **local minima** are of form  $X = U_p \sqrt{-\Lambda_p} Q$ , where  $\sqrt{\cdot}$  is applied entry-wise,  $P \in \mathbb{R}^{q \times p}$  is the first  $p$  columns of an arbitrary  $q$ -by- $q$  permutation matrix,  $S \in \mathbb{R}^{p \times p}$  is a diagonal matrix with diagonal entries being 0 or 1, and  $Q \in \mathbb{R}^{p \times p}$  is an arbitrary orthogonal matrix. Further, any local minimum is also a global minimum.*

Notice that in Theorem 2.1,  $A$  is assumed to have at least  $p$  negative eigenpairs, which is not necessary for the theorem to hold. When  $A$  has less number of negative eigenpairs, both stationary points and global minima can be modified to include zero columns. In order to simply the presentation in this paper, we stick to this assumption for (Obj1) throughout the paper. For practical problems from physics, this assumption usually holds and no diagonal shift of the matrix is needed.

## 2.2 Objective Function Two

For (Obj2), the intuition is more complicated. There are two ways to understand the construction of the objective function: approximation of inverse and Lagrange multiplier.

Recalling the objective function in Rayleigh quotient (3), the matrix inverse leads to expensive computational cost. Assuming the spectrum of  $X^\top X$  is bounded by one, we have the Neumann series expansion and its first order approximation as,

$$(X^\top X)^{-1} = (I - (I - X^\top X))^{-1} = \sum_{i=1}^{\infty} (I - X^\top X)^i \approx 2I - X^\top X, \quad (6)$$

which leads to the approximation of the objective function as

$$\text{tr}((X^\top X)^{-1} X^\top A X) \approx \text{tr}((2I - X^\top X) X^\top A X). \quad (7)$$

Another way to understand the construction of (Obj2) is through applying the Lagrange multiplier method to (2), where the Lagrangian function is

$$\mathcal{L}(X, \Xi) = \text{tr}(X^\top A X) - \text{tr}(\Xi(X^\top X - I)) \quad (8)$$

and  $\Xi$  denotes the Lagrange multiplier. Using the first order optimality condition,  $\frac{\partial \mathcal{L}}{\partial X} = 0$ , and left multiplying both sides by  $X$ , we obtain  $\Xi = X^\top A X$ . Substituting the expression of  $\Xi$  back into  $\mathcal{L}$  leads to (Obj2).

The gradient of (Obj2) can be derived through a multivariable calculus as,

$$\nabla f_2(X) = 4AX - 2XX^\top AX - 2AXX^\top X. \quad (9)$$

Analyzing the properties of the gradient and Hessian, previous work [22] shows that (Obj2) has simple form of the stationary points and does not have any spurious local minimum. The theorems therein are summarized as follows.

**Theorem 2.2.** *Let  $A$  be a negative definite matrix. All **stationary points** of (Obj2) are of form  $X = UPSQ$  and all **local minima** are of form  $X = U_p Q$ , where  $P \in \mathbb{R}^{n \times p}$  is the first  $p$  columns of an arbitrary  $n$ -by- $n$  permutation matrix,  $S \in \mathbb{R}^{p \times p}$  is a diagonal matrix with diagonal entries being 0 or 1, and  $Q \in \mathbb{R}^{p \times p}$  is an arbitrary orthogonal matrix. Further, any local minimum is also a global minimum.*

Notice that the matrix  $A$  in (Obj2) must be negative definite. Otherwise,  $X$  can be selected as scaled eigenvectors corresponding to the positive eigenvalues and  $f_2(X)$  can be arbitrarily negative and unbounded from below. For eigenvalue problems, the matrix can be shifted to be negative definite, which requires an estimation of the largest eigenvalue. Later in this paper, we will always assume  $A$  is negative definite in the case of (Obj2).

Based on the analysis of the energy landscape of both (Obj1) and (Obj2), any algorithm avoiding saddle points converges to the global minimum. Such algorithms include but not limit to regular gradient descent [15], conjugate gradient descent, stochastic gradient descent [2, 17], etc. Using the notation defined in Table 1 and gradients defined as (5) and (9), the gradient descent iterations for (Obj1) and (Obj2) are defined as,

$$X^{(t+1)} = X^{(t)} - \alpha \left( AX^{(t)} + X^{(t)} \left( X^{(t)} \right)^\top X^{(t)} \right), \quad (10)$$

and

$$X^{(t+1)} = X^{(t)} - \alpha \left( 2AX^{(t)} - AX^{(t)} \left( X^{(t)} \right)^\top X^{(t)} - X^{(t)} \left( X^{(t)} \right)^\top AX^{(t)} \right), \quad (11)$$

where we have absorbed the constant common factor into the stepsize  $\alpha$ . Unfortunately, the Hessian of both (Obj1) and (Obj2) are unbounded from above, hence the valid set for the choice of the stepsize over the entire domain is empty. One solution which has been applied before [19] is to find a huge bounded domain of  $X$  such that iterations are guaranteed to stay within this domain and the objective functions have bounded Hessian and nonempty choice for the stepsize. In Section 3, we need a similar theorem as well.

As shown in Theorem 2.1 and Theorem 2.2, the global minima are generated as a linear combination of eigenvectors corresponding to low-lying eigenvalues. Further computation is required to obtain the low-lying eigenpairs, which is needed in many practical computations like DFT for metallic systems, FCI, etc. We introduce two ways to compute eigenpairs from the global minimum  $X$ . The first way works only for (Obj1). We can apply singular value decomposition (SVD) to  $X$  and obtain the explicit decomposition  $X = U_p \sqrt{-\Lambda_p} Q$ . Hence the eigenpairs can be recovered explicitly. The second way works for both (Obj1) and (Obj2). Applying eigenvalue decomposition to  $X^\top A X$  results,  $X^\top A X = Q^\top \Lambda Q$ , which gives eigenvalues explicitly. Then computing  $U_p = X Q^\top \sqrt{-\Lambda_p}^{-1}$  and  $U_p = X Q^\top$  gives the eigenvectors for (Obj1) and (Obj2) respectively. Both ways for computing explicit eigenpairs cost  $O(np^2)$  operations, which is negligible when  $p \ll n$ . When  $p \sim n$  as in DFT calculation, the post processing cost is still cubic scaling in  $n$  and lacks parallel efficiency.

### 3 Triangularized Optimization Eigensolvers

We propose triangularized orthogonalization-free methods (TriOFM) as eigensolvers based on (Obj1) and (Obj2), which are denoted as TriOFM-(Obj1) and TriOFM-(Obj2).

Our goal as mentioned in Section 1 is to find  $p$  extreme eigenpairs with two properties: (i). Orthogonalization of  $X$  is not permitted; (ii). Eigenvectors are sparse vectors. Optimizing (Obj1) and (Obj2) almost achieves the first required property except the post processing part. While, the second property is totally ignored. Due to the existence of the arbitrary orthogonal matrix  $Q$ , the convergent point for both (Obj1) and (Obj2) almost surely destroy the sparsity in the original eigenvectors. Adding  $\ell_1$  penalty to (Obj2) [23] is proposed to achieve the sparsity as much as possible in DFT problems, which is not likely to be applicable to FCI problems.

Another way of explicitly getting the eigenpairs rather than a point in the eigenspace is to solve the single column version of (Obj1) or (Obj2) multiple times. For the first time, we solve the single column version of (Obj1) or (Obj2) for  $A_1 = A$  and obtain the smallest eigenpair  $\lambda_1$  and  $u_1$ . Then we apply either method to  $A_2 = A_1 - \lambda_1 u_1 u_1^\top$  and obtain  $\lambda_2$  and  $u_2$ . At  $k$ -th time, either method is applied to  $A_k = A_{k-1} - \lambda_{k-1} u_{k-1} u_{k-1}^\top = A - \sum_{i=1}^{k-1} \lambda_i u_i u_i^\top$  and  $\lambda_k$  and  $u_k$  are computed. Such a procedure has two drawbacks. First, single column operation is known as BLAS2-level (matrix-vector) operation, which is significantly slower than BLAS3-level (matrix-matrix) operation due to the memory hierarchy in modern computer architecture. This drawback would mostly impact the performance of DFT calculations. The second drawback is due to the transformed matrix  $A_k$ . The sparsity in  $A$  plays crucial role in designing coordinate descent algorithms for FCI problems. However, it is destroyed in  $A_k$  by the additive low-rank matrix. Hence the second drawback makes the proposed procedure infeasible for FCI problems.

Although the afore mentioned procedure is not working satisfactorily for our problems, it does inspire TriOFM-(Obj1) and TriOFM-(Obj2). We first motivate and derive TriOFM-(Obj1). Then TriOFM-(Obj2) can be derived in an analogy way.

In the above procedure, the single column version of (Obj1) is applied to  $A_k = A - \sum_{i=1}^{k-1} \lambda_i u_i u_i^\top$ . Notice that if the column-by-column procedure is applied, the convergent point of  $x_i$  is  $\pm \sqrt{-\lambda_i} u_i$ . Hence,  $A_k$  can be viewed as the summation of  $A$  and the outer product of convergent vector of  $x_1$ ,

$x_2, \dots, x_{k-1}$ . If we assume all columns iterate together, and the single column version of (Obj1) is applied to the most closed approximation of  $A_k$ , *i.e.*,  $A_k \approx \tilde{A}_k = A + \sum_{i=1}^{k-1} x_i x_i^\top$ , then we obtain the following iterative schemes,

$$\begin{aligned}
x_1^{(t+1)} &= x_1^{(t)} - \alpha \left( Ax_1^{(t)} + x_1^{(t)} \left( x_1^{(t)} \right)^\top x_1^{(t)} \right), \\
x_2^{(t+1)} &= x_2^{(t)} - \alpha \left( Ax_2^{(t)} + x_1^{(t)} \left( x_1^{(t)} \right)^\top x_2^{(t)} + x_2^{(t)} \left( x_2^{(t)} \right)^\top x_2^{(t)} \right), \\
&\dots \\
x_k^{(t+1)} &= x_k^{(t)} - \alpha \left( Ax_k^{(t)} + \sum_{i=1}^k x_i^{(t)} \left( x_i^{(t)} \right)^\top x_k^{(t)} \right), \\
&\dots
\end{aligned} \tag{12}$$

Using matrix notation, the above iterative schemes admit the following representation,

$$X^{(t+1)} = X^{(t)} - \alpha \left( AX^{(t)} + X^{(t)} \text{triu} \left( \left( X^{(t)} \right)^\top X^{(t)} \right) \right), \tag{13}$$

where  $\text{triu}(\cdot)$  denote the upper triangular matrix. The key difference between (10) and (13) is that the gradient is modified by an upper triangular version,

$$g_1(X) = AX + X \text{triu} \left( X^\top X \right). \tag{14}$$

Next we analyze the stationary points of (13) in Theorem 3.1

**Theorem 3.1.** *Assume that  $A$  has  $q$  negative eigenpairs with  $q \geq p$ . All **stationary points** of (13) are of form  $X = U_q \sqrt{-\Lambda_q} P S D$ , where  $\sqrt{\cdot}$  is applied entry-wise,  $P \in \mathbb{R}^{q \times p}$  is the first  $p$  columns of an arbitrary  $q$ -by- $q$  permutation matrix,  $S \in \mathbb{R}^{p \times p}$  is a diagonal matrix with diagonal entries being 0 or 1, and  $D \in \mathbb{R}^{p \times p}$  is a diagonal matrix with diagonal entries being +1 or -1. Within these points all **stable stationary points** are of form  $X = U_p \sqrt{-\Lambda_p} D$  and others are **unstable stationary points**.*

*Proof.* All stationary points of (13) satisfy  $g_1(X) = 0$  for  $g_1(X)$  being a  $n$ -by- $p$  matrix. We prove the theorem by induction. Here we introduce notations for four matrices:  $X_i$  denote the first  $i$  columns of  $X$ ,  $P_i \in \mathbb{R}^{q \times i}$  is the first  $i$  columns of an arbitrary  $q$ -by- $q$  permutation matrix,  $S_i \in \mathbb{R}^{i \times i}$  is a diagonal matrix with diagonal entries being 0 or 1, and  $D_i \in \mathbb{R}^{i \times i}$  is a diagonal matrix with diagonal entries being +1 or -1.

Consider the first column of  $g_1(X) = 0$ , which is

$$Ax_1 + x_1 x_1^\top x_1 = 0, \tag{15}$$

for  $x_1^\top x_1$  being a scalar. When  $x_1 = 0$ , (15) naturally holds. When  $x_1 \neq 0$ ,  $x_1$  must be a scalar multiple of an eigenvector of  $A$  and  $x_1^\top x_1$  is the negative number of the corresponding eigenvalue, which must be negative. Hence  $X_1 = x_1$  is of the form,  $X_1 = U_q \sqrt{-\Lambda_q} P_1 S_1 D_1$ .

Now assume the first  $i$  columns of  $X$  obeys  $X_i = U_q \sqrt{-\Lambda_q} P_i S_i D_i$ . Then the  $(i+1)$ -th column of  $g_1(X) = 0$  is

$$0 = Ax_{i+1} + X_i X_i^\top x_{i+1} + x_{i+1} x_{i+1}^\top x_{i+1} = \tilde{A} x_{i+1} + x_{i+1} x_{i+1}^\top x_{i+1}, \tag{16}$$



where  $\tilde{A} = A + X_i X_i^\top = A + U_q \sqrt{-\Lambda_q} P_i S_i P_i^\top \sqrt{-\Lambda_q} U_q^\top$ .  $\tilde{A}$  is the original matrix  $A$  zeroing out a few eigenvalues corresponding to the selected columns in  $P_i$  with 1 in  $S_i$ . Applying the similar analysis as in the case of (15) to (16), we conclude that  $X_{i+1}$  is of the form,  $X_{i+1} = U_q \sqrt{-\Lambda_q} P_{i+1} S_{i+1} D_{i+1}$ .

Since  $q \geq p$ , we have sufficient number of negative eigenpairs to be added to  $X$ . The induction can be processed until  $i = p$ , and we obtain the expression for all stationary points as in the theorem.

The stabilities of stationary points are determined by the spectrum of their Jacobian matrices of  $g_1$ , *i.e.*,  $Dg_1(X)$ . Since both  $g_1(X)$  and  $X$  are matrices, the Jacobian is a 4-way tensor, which is unfolded as a matrix here. In order to avoid over complicated index in subscripts, we denote the matrix  $g_1(X)$  as  $G$ . Notation  $G_{ij}$  and  $x_{ij}$  denote the element on  $j$ -th row and  $i$ -th column of  $G$  and  $X$  respectively. Then the Jacobian matrix is written as a  $p$ -by- $p$  block matrix,

$$Dg_1(X) = DG = \begin{pmatrix} J_{11} & \cdots & J_{1p} \\ \vdots & \ddots & \vdots \\ J_{p1} & \cdots & J_{pp} \end{pmatrix}, \quad (17)$$

with each block  $J_{ij}$  being,

$$J_{ij} = \begin{pmatrix} \frac{\partial G_{i1}}{\partial x_{j1}} & \cdots & \frac{\partial G_{in}}{\partial x_{j1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial G_{i1}}{\partial x_{jn}} & \cdots & \frac{\partial G_{in}}{\partial x_{jn}} \end{pmatrix}. \quad (18)$$

Notice that the  $i$ -th column of  $G$ ,  $G_i = Ax_i + X_i X_i^\top x_i$ , is independent of  $x_{i+1}, \dots, x_p$ , which means  $J_{ij} = 0$  for  $i < j$ . Hence  $Dg_1(X)$  is a block upper triangular matrix. The spectrum of  $Dg_1(X)$  is determined by the spectrum of  $J_{ii}$  for  $i = 1, 2, \dots, p$ . Through a multivariable calculus, we obtain the explicit expression for  $J_{ii}$ ,

$$J_{ii} = A + X_i X_i^\top + x_i^\top x_i I + x_i x_i^\top. \quad (19)$$

We first show the stability of the stationary points of form  $X = U_p \sqrt{-\Lambda_p} D$ . Substituting these points into (19), we have,

$$J_{ii} = A - U_i \Lambda_i U_i^\top - \lambda_i I - u_i \lambda_i u_i^\top. \quad (20)$$

Since  $\lambda_i$  is negative and strictly smaller than all eigenvalues of  $A - U_i \Lambda_i U_i^\top$ ,  $J_{ii}$  is strictly positive definite for all  $i = 1, 2, \dots, p$ . Therefore we have all eigenvalues of  $Dg_1(U_p \sqrt{-\Lambda_p} D)$  are strictly positive and  $X = U_p \sqrt{-\Lambda_p} D$  are stable stationary points.

Next, we show the instability of the rest stationary points. If  $X$  is a stationary points but not of the form  $U_p \sqrt{-\Lambda_p} D$ , then there exist indices  $s$  such that  $x_s^\top u_s = 0$ . Denote  $s$  as the first such index. Substituting this point into  $J_{ss}$  and computing the bilinear form of  $J_{ss}$  with respect to  $u_s$ , we have,

$$u_s^\top J_{ss} u_s = \lambda_s - x_s^\top x_s < 0, \quad (21)$$

where the inequality comes from the fact that  $x_s$  is zero or corresponds to eigenvalues greater than  $\lambda_s$ . Hence the Jacobian matrix has negative eigenvalues. Hence these points are unstable stationary points.  $\square$

Algorithm 1 illustrates the detailed pseudocode for (13). The choice of the stepsize is unspecified, which will be revealed in later sections.

There is another way to understand the iterative scheme. The column with smaller index is decoupled from the later columns. For example, the iterative scheme of  $x_1$  is independent of all later

---

**Algorithm 1** TriOFM-(Obj1)

---

**Input:** a symmetric matrix  $A$ , an initial point  $X^{(0)}$   
 $t = 0$   
**while** not converged **do**  
     $g_1^{(t)} = AX^{(t)} + X^{(t)}\text{triu}\left(\left(X^{(t)}\right)^\top X^{(t)}\right)$   
    Choose a stepsize  $\alpha^{(t)}$   
     $X^{(t+1)} = X^{(t)} - \alpha^{(t)}g_1^{(t)}$   
     $t = t + 1$

---

columns. For the second column,  $x_2$ , the iterative scheme on  $x_2$  is the same as the second column in the 2-column version of (Obj1). Recursively applying the idea, we also reach to Algorithm 1.

Similar idea can be applied to solve (Obj2) as well. We notice that there are two terms in (11) coupling columns together, *i.e.*,  $AX^{(t)}\left(X^{(t)}\right)^\top X^{(t)}$  and  $X^{(t)}\left(X^{(t)}\right)^\top AX^{(t)}$ . Using the decoupling idea, we can replace the  $\left(X^{(t)}\right)^\top X^{(t)}$  and  $\left(X^{(t)}\right)^\top AX^{(t)}$  by their upper triangular parts and result the following iterative scheme,

$$X^{(t+1)} = X^{(t)} - \alpha \left( 2AX^{(t)} - AX^{(t)}\text{triu}\left(\left(X^{(t)}\right)^\top X^{(t)}\right) - X^{(t)}\text{triu}\left(\left(X^{(t)}\right)^\top AX^{(t)}\right) \right). \quad (22)$$

Comparing to (11), the gradient is modified as,

$$g_2(X) = 2AX - AX\text{triu}(X^\top X) - X\text{triu}(X^\top AX). \quad (23)$$

The stationary points of (22) can be analyzed in a slightly modified way. We summarize the properties in Theorem 3.2 and leave the proof in Appendix A.

**Theorem 3.2.** *Let  $A$  be a negative definite matrix. All **stationary points** of (22) are of form  $X = UPSD$  and all **stable stationary points** are of form  $X = U_p D$ , where  $P \in \mathbb{R}^{n \times p}$  is the first  $p$  columns of an arbitrary  $n$ -by- $n$  permutation matrix,  $S \in \mathbb{R}^{p \times p}$  is a diagonal matrix with diagonal entries being 0 or 1, and  $D \in \mathbb{R}^{p \times p}$  is a diagonal matrix with diagonal entries being +1 or -1.*

Algorithm 2 illustrates the detailed pseudocode for (11) and the choice of the stepsize is also deferred to later sections.

---

**Algorithm 2** TriOFM-(Obj2)

---

**Input:** a symmetric matrix  $A$ , an initial point  $X^{(0)}$   
 $t = 0$   
**while** not converged **do**  
     $g_2^{(t)} = 2AX^{(t)} - AX^{(t)}\text{triu}\left(\left(X^{(t)}\right)^\top X^{(t)}\right) - X^{(t)}\text{triu}\left(\left(X^{(t)}\right)^\top AX^{(t)}\right)$   
    Choose a stepsize  $\alpha^{(t)}$   
     $X^{(t+1)} = X^{(t)} - \alpha^{(t)}g_2^{(t)}$   
     $t = t + 1$

---

We claim a few advantages of Algorithm 1 and Algorithm 2 over other related methods. First, both algorithms converge to the eigenvectors or their scaled ones without mixing them. Hence the sparsity of the eigenvectors is preserved. Although we do not benefit from the sparsity during the iteration in Algorithm 1 and Algorithm 2 directly, we expect that the coordinate descent versions

of them would benefit from the sparsity and achieve fast convergence and small memory cost for FCI problems. Second, the orthogonalization step is totally removed, which makes the algorithm friendly to parallel computing. Third, all cubic scaling operations can be processed through BLAS3-level routines. Algorithms, therefore, benefit from the memory hierarchy of modern computer architecture.

All gradient based optimization algorithms can be viewed as a discrete time dynamic systems on a conservative field. However, Algorithm 1 and Algorithm 2 are discrete time dynamic systems on a non-conservative field, since neither  $g_1(X)$  nor  $g_2(X)$  corresponds to a gradient of an energy functional. Hence the usual convergence analysis in optimization field can not be applied directly here. Through a careful analysis inspired by optimization analysis, in Section 4 and Section 5, we propose the global and local convergence analysis for both algorithms.

Although we only propose Algorithm 1 and Algorithm 2 and analyze their convergence in this paper, the idea of TriOFM can be applied to a wide range of algorithms to remove the redundancy introduced by gauge freedom or gauge invariance. The key point here is decoupling each column from later columns while ensuring the iterative scheme for later column is the same as solving the multicolumn version of the objective function. We conduct the convergence analysis for Algorithm 1 and Algorithm 2 case by case. Hence we expect the convergence analysis for other TriOFM algorithms also need to be done case by case. The unified convergence analysis for TriOFM is open.

## 4 Global Convergence Analysis

In this section, we prove the global convergence for Algorithm 1 and Algorithm 2. The global convergence analysis for both algorithms are intuitively simple but technically difficult. We first state the intuition for  $p = 2$ . The global convergence of a single column version of Algorithm 1 is almost surely guaranteed [19]. Intuitively, if we assume the first columns  $x_1$  has converged and is frozen to be a global minimum, then the second column in the iterative scheme (13) is associated with an optimization problem,

$$\|A + x_2x_2^\top\|_F^2 + 2\|x_1^\top x_2\|_F^2 = \|\tilde{A} + x_2x_2^\top\|_F^2, \quad (24)$$

where  $\tilde{A} = A + x_1x_1^\top = A - \lambda_1u_1u_1^\top$ . Notice the optimization problem for  $x_2$  is of the same form as the single column version of Algorithm 1. Hence the convergence of  $x_2$  is also guaranteed almost surely. However, there are two niches. First, the numerical error for the converged  $x_1$  should be taken into account in the convergence analysis of  $x_2$ , which would be similar to an sensitivity analysis. The second niche is more problematic. For a random initial  $x_1$  and  $x_2$ , the convergence of  $x_1$  should not make  $x_2$  fall into the problematic zero-measure initial set where  $x_2$  will converge to an unstable stationary point. A complete proof must fill these two niches.

In the following, we conduct a careful global convergence analysis for Algorithm 1 step-by-step. The analog proof procedure can be applied to show the global convergence for Algorithm 2. Hence, the global convergence theorem for Algorithm 2 is given without a detailed proof.

All lemmas and theorems for Algorithm 1 in this section are stated under Assumption A, where we assume the iteration starts within a big domain, where the Hessian matrix is bounded from above, and the stepsize is fixed and upper bounded by a small constant.

**Assumption A.** Let  $R_i = 2^{i-1}\sqrt{3\rho}$  for all  $1 \leq i \leq p$ . The initial point  $X^{(0)} = \begin{pmatrix} x_1^{(0)} & x_2^{(0)} & \cdots & x_p^{(0)} \end{pmatrix}$  satisfies  $\|x_i^{(0)}\| \leq R_i$  for all  $1 \leq i \leq p$ . Iteration follows Algorithm 1 with a constant stepsize satisfying  $\alpha \leq \frac{1}{5R_p^2}$ .

According to Theorem 3.1, stable stationary points are of form  $X = U_p \sqrt{-\Lambda_p} D$  and columns are scalar multiple of the  $p$  low-lying eigenvectors of  $A$ . Global convergence aims to show that Algorithm (1) converges to one of the stable stationary points. In order to simplify the notation, we define the set of stable stationary points as  $\mathcal{X}^* = \{U_p \sqrt{-\Lambda_p} D\}$ , where  $U_p$ ,  $\Lambda_p$ , and  $D$  are defined in Theorem 3.1. Further, the distance between a point  $X$  and the set  $\mathcal{X}^*$  is denoted as  $\|X - \mathcal{X}^*\|_F = \min_{Y \in \mathcal{X}^*} \|X - Y\|_F$ , which means the smallest F-norm between  $X$  and all points in  $\mathcal{X}^*$ . For the first  $k$  columns, we define the set of stable stationary points as  $\mathcal{X}_k^* = \{U_k \sqrt{-\Lambda_k} D_k\}$ , where  $U_k$  is the first  $k$  columns of  $U_p$ ,  $\Lambda_k$  and  $D_k$  are the first principle  $k$ -by- $k$  submatrices of  $\Lambda_p$  and  $D$  respectively.

Theorem 4.4 states that Algorithm 1 converges to a global minimum almost surely. While the proof depends on the following lemmas, including another few lemmas in Appendix B. Within these lemmas, the proof directly depends on the following ones: Lemma 4.1 guarantees that the iteration following Algorithm 1 stays within a big domain as long as the initial point is in there; Lemma 4.2 shows the global convergence of  $x_1$ ; Lemma 4.3 shows the global convergence of  $x_k$  for  $k > 1$ . We leave the proofs of these lemmas in Appendix B.

**Lemma 4.1.** *Assume Assumption A is satisfied. Then for any iteration  $t$ ,  $X^{(t)} = (x_1^{(t)}, \dots, x_n^{(t)})$  satisfies  $\|x_i^{(t)}\| \leq R_i$  for all  $1 \leq i \leq p$ .*

**Lemma 4.2.** *Assume Assumption A is satisfied and  $x_1^{(0)}$  is not perpendicular to  $u_1$ . Then  $x_1^{(t)}$  converges to  $\pm \sqrt{-\lambda_1} u_1$ .*

**Lemma 4.3.** *Assume Assumption A is satisfied and  $\lim_{t \rightarrow \infty} \|X_{k-1}^{(t)} - \mathcal{X}_{k-1}^*\|_F = 0$ . Then  $x_k^{(t)}$  converges to one of the points in  $\{0, \pm \sqrt{-\lambda_j} u_j \text{ for } j = k, \dots, p\}$ .*

**Theorem 4.4.** *If Assumption A is satisfied, then Algorithm 1 converges to  $\mathcal{X}^*$  for all initial points except those in  $W$ , where the set  $W$  has measure zero.*

*Proof.* For the iteration in TriOFM-(Obj1), as mentioned earlier, earlier columns are independent of later columns. Hence induction is used to prove this theorem. In the following proof, all initial points satisfy Assumption A without further notice and the set of these initial points is denoted as  $\mathcal{X}_k$  for the first  $k$  columns. Lemma 4.1 guarantees that  $\iota - \alpha g_1$  maps points in  $\mathcal{X}_k$  to  $\mathcal{X}_k$ , where  $\iota$  denotes the identity operator and  $g_1$  is the operator defined in (14). We further introduce a notation for unstable stationary points for  $\mathcal{X}_k$  as  $\mathcal{A}_k^*$ . Recall Theorem 3.1 for  $p = k$ , we can characterize  $\mathcal{A}_k^*$  as,

$$\mathcal{A}_k^* = \left\{ X \in \mathcal{X}_k \mid X = U_q \sqrt{-\Lambda_q} PSD \text{ and } X \neq U_k \sqrt{-\Lambda_k} D \right\}. \quad (25)$$

For the first column  $x_1^{(t)}$ , Lemma 4.2 shows that  $\lim_{t \rightarrow \infty} x_1^{(t)} = \pm \sqrt{-\lambda_1} u_1$  for all  $x_1^{(0)}$  not perpendicular to  $u_1$ . Alternatively, it can be restated as  $\lim_{t \rightarrow \infty} \|X_1^{(t)} - \mathcal{X}_1^*\|_F = 0$  for all initial points except those in  $W_1 = \left\{ X_1^{(0)} \mid u_1^\top X_1^{(0)} = 0 \right\} = \left\{ X_1^{(0)} \mid \lim_{t \rightarrow \infty} (\iota - \alpha g_1)^t (X_1^{(0)}) \in \mathcal{A}_1^* \right\}$ . Obviously the set  $W_1$  has measure zero.

Now we assume that the statement of Theorem 4.4 holds for the first  $k-1$  columns for  $k \in (1, p]$ , i.e.,  $\lim_{t \rightarrow \infty} \|X_{k-1}^{(t)} - \mathcal{X}_{k-1}^*\|_F = 0$  for all initial points except those in  $W_{k-1}$  and the set  $W_{k-1}$  has measure zero.

We first define the set  $W_k$  for  $k$  as,

$$W_k = \left\{ X_k^{(0)} \mid \lim_{t \rightarrow \infty} (\iota - \alpha g_1)^t (X_k^{(0)}) \in \mathcal{A}_k^* \right\} \cup V_k, \quad (26)$$

for  $V_k = \{X_k^{(0)} \mid X_{k-1}^{(0)} \in W_{k-1}\}$ . Since  $W_{k-1}$  has measure zero, we know that the set  $V_k$  also has measure zero. Next we focus on the points in  $\mathcal{X}_k \setminus V_k$  and  $W_k \setminus V_k$ .

Here we would like to apply Theorem 2 in Lee et al. [15] to show that  $W_k \setminus V_k$  has measure zero. All conditions therein must be checked first. Since, under the operator  $\iota - \alpha g_1$ , the first  $k-1$  columns are independent of the  $k$ -th one, the operator  $\iota - \alpha g_1$  is smooth and maps  $\mathcal{X}_k \setminus V_k$  to  $\mathcal{X}_k \setminus V_k$ . According to Theorem 3.1, points in  $\mathcal{A}_k^*$  are unstable stationary points, so are points in  $\mathcal{A}_k^* \setminus V_k$ . The last thing to check is the invertibility of  $D(\iota - \alpha g_1) = I - \alpha Dg_1$ . As has been discussed in the proof of Theorem 3.1, the spectrum of  $Dg_1$  is determined by the spectrum of  $J_{ii}$  for  $i = 1, \dots, k$ , where  $J_{ii}$  is of form (19). For points in  $\mathcal{X}_k$ , the spectrum norm of  $X_i X_i^\top$ ,  $x_i^\top x_i I$ , and  $x_i x_i^\top$  are upper bounded by  $2R_p^2$ ,  $R_p^2$ , and  $R_p^2$  respectively. Further we have  $\|A\| < \frac{R_p^2}{3}$ . Hence, combined with the assumption on  $\alpha$ , we have the following bound,

$$\lambda(D(\iota - \alpha g_1)) > \frac{1}{15} \quad (27)$$

for all  $X_k \in \mathcal{X}_k$ , which implies that  $\det(D(\iota - \alpha g_1)) \neq 0$  for all  $X_k \in \mathcal{X}_k \setminus V_k$ . Finally, by applying Theorem 2 in Lee et al. [15], the set  $W_k \setminus V_k$  has measure zero. Therefore, the set  $W_k$  has measure zero.

Then Lemma 4.3 shows that for  $X_k^{(0)} \in \mathcal{X}_k \setminus W_k$  there is

$$\lim_{t \rightarrow \infty} x_k^{(t)} = \pm \sqrt{-\lambda_k} u_k. \quad (28)$$

Hence we have  $\lim_{t \rightarrow \infty} \|X_k^{(t)} - \mathcal{X}_k^*\|_{\mathbb{F}} = 0$  for all initial points except those in  $W_k$  and the set  $W_k$  has measure zero. Using the induction argument, the theorem is proved.  $\square$

After proving the global convergence of TriOFM-(Obj1), here we informally state another theorem for TriOFM-(Obj2) without proof.

**Theorem 4.5** (informal). *If stepsize is a small constant and the initial point satisfies  $\|X^{(0)}\| \leq \frac{3}{2}$ , then Algorithm 2 converges to  $\{U_p D\}$  as defined in Theorem 3.2 for all initial points except those in  $W$ , where the set  $W$  has measure zero.*

Both Theorem 4.4 and Theorem 4.5 show that algorithms with the proposed triangularization strategy converge to the global minima of the objective functions for almost all the initial points, if the stepsize is a small constant. However, since the objective functions we have here, (Obj1) and (Obj2), are non-convex and have strict saddle points, we do not expect any provable rate of convergence.

## 5 Local Convergence Analysis

In this section, we will prove the local linear convergence of TriOFM-(Obj1), the similar conclusion for TriOFM-(Obj2) can be proved.

In Section 4 when we are proving the global convergence, we have already showed the linear convergence of our TriOFM algorithm in the single column case, but the convergence rate hidden in the proof of Lemma 4.2 is too messy to become a part of a linear convergence proof. In this section we assume  $X^{(t)}$  is already close to global minima and obtain a linear convergence for TriOFM-(Obj1) in detail.

The set of global minima  $\mathcal{X}^* = \{U_p \sqrt{-\Lambda_p} D\}$  are used as the convergence target in previous section and the convergence is analyzed under the F-norm distance between the iterator and this set. However, in this section, we first focus on the convergence to a global minimum, denoted as  $X^* \in \mathcal{X}^*$ . Similar notations as in Table 1 are applied to  $X^*$  as well. More precisely,  $x_i^*$  denotes the  $i$ -th column of  $X^*$  and  $X_i^*$  denotes the first  $i$  columns of  $X^*$ . The analysis of the local convergence to  $X^*$  is carried out through Lemma 5.1, Lemma 5.2, and Theorem 5.3. Lemma 5.1 and Lemma 5.2 provide per-iteration bound on the residual of the first column and later columns respectively, and Theorem 5.3 put together these bounds and concludes a local linear convergence. Once the local convergence to a global minimum is analyzed, we show the linear convergence to the set of global minima in Corollary 5.4. Since all global minima are isolated from each other and the small stepsize restrict the iterator staying around a particular minimum, the linear convergence to the set of global minima is a straightforward result from Theorem 5.3.

**Lemma 5.1.** *Assume Assumption A is satisfied. Let  $\varepsilon_1^{(t)}$  be the residual of the first column in  $t$ -th iteration,  $\varepsilon_1^{(t)} = x_1^{(t)} - x_1^*$ . If  $\|\varepsilon_1^{(t)}\| \leq \frac{\lambda_2 - \lambda_1}{8\sqrt{-\lambda_1}}$ , then  $\|\varepsilon_1^{(t+1)}\| \leq \left(1 - \alpha \frac{\lambda_2 - \lambda_1}{2}\right) \|\varepsilon_1^{(t)}\|$ .*

*Proof.* Without loss of generality, we assume that  $A$  is a diagonal matrix. For simplicity, we drop the iteration index superscript and use  $x_1 = x_1^{(t)}$ ,  $\varepsilon_1 = \varepsilon_1^{(t)}$ ,  $\tilde{x}_1 = x_1^{(t+1)}$  and  $\tilde{\varepsilon}_1 = \varepsilon_1^{(t+1)}$  instead. Further we denote the first column of  $X^*$  as  $v_1 = x_1^*$ . Using the expression of  $X^*$ , we have  $v_1^\top v_1 = -\lambda_1$  and  $v_1 v_1^\top = -\lambda_1 u_1 u_1^\top = -\lambda_1 e_1 e_1^\top$ .

Based on the iterative scheme on the first column, *i.e.*,  $\tilde{x}_1 = x_1 - \alpha A x_1 - \alpha x_1^\top x_1 x_1$ , there is

$$\begin{aligned} \tilde{\varepsilon}_1 &= \tilde{x}_1 - v_1 = \varepsilon_1 - \alpha A(v_1 + \varepsilon_1) - \alpha(v_1 + \varepsilon_1)^\top(v_1 + \varepsilon_1)(v_1 + \varepsilon_1) \\ &= \left((1 + \alpha\lambda_1)I - \alpha A - 2\alpha v_1 v_1^\top\right) \varepsilon_1 - \alpha v_1 \|\varepsilon_1\|^2 - 2\alpha v_1^\top \varepsilon_1 \varepsilon_1 - \alpha \|\varepsilon_1\|^2 \varepsilon_1. \end{aligned} \quad (29)$$

The assumption on  $\alpha$  implies that  $1 + \alpha\lambda_1 \pm \alpha\lambda_i > 0$  holds for any  $i$ . Hence, the two norm of the diagonal matrix  $(1 + \alpha\lambda_1)I - \alpha A - 2\alpha v_1 v_1^\top$  is

$$\begin{aligned} \|(1 + \alpha\lambda_1)I - \alpha A - 2\alpha v_1 v_1^\top\| &= \left\| \begin{pmatrix} 1 + 2\alpha\lambda_1 & & & \\ & 1 + \alpha\lambda_1 - \alpha\lambda_2 & & \\ & & \ddots & \\ & & & 1 + \alpha\lambda_1 - \alpha\lambda_n \end{pmatrix} \right\| \\ &= 1 + \alpha\lambda_1 - \alpha\lambda_2. \end{aligned} \quad (30)$$

With this estimation, the two norm of  $\tilde{\varepsilon}_1$  can be bounded as,

$$\begin{aligned} \|\tilde{\varepsilon}_1\| &\leq (1 + \alpha\lambda_1 - \alpha\lambda_2) \|\varepsilon_1\| + 3\alpha\sqrt{-\lambda_1} \|\varepsilon_1\|^2 + \alpha \|\varepsilon_1\|^3 \\ &\leq \left(1 - \alpha \frac{\lambda_2 - \lambda_1}{2}\right) \|\varepsilon_1\|, \end{aligned} \quad (31)$$

where the last inequality adopts the fact  $\|\varepsilon_1\| \leq \frac{\lambda_2 - \lambda_1}{8\sqrt{-\lambda_1}}$ . □

Here we have proved, for the single column case, the residual of  $x_1$  has a linear convergence with speed at least  $1 - \alpha \frac{\lambda_2 - \lambda_1}{2}$ . While, before we move onto the multicolumn case, we would like to mention that, if we ignore the higher order terms,  $\|\varepsilon_1\|^2$ , and  $\|\varepsilon_1\|^3$ , we are able to get a better linear convergence rate. For  $u_1^\top \varepsilon_1$ , we have  $u_1^\top \varepsilon_1^{(t+1)} \approx (1 + 2\alpha\lambda_1) u_1^\top \varepsilon_1^{(t)}$ . For  $u_i^\top \varepsilon_1$  with  $i > 1$ , we have  $u_i^\top \varepsilon_1^{(t+1)} \approx (1 - \alpha\lambda_i + \alpha\lambda_1) u_1^\top \varepsilon_1^{(t)}$ .

Now we turn to the multicolumn case. For the  $i$ -th column  $x_i$ , we have the following lemma.

**Lemma 5.2.** Assume Assumption A is satisfied. Let  $\varepsilon_i^{(t)}$  be the residual of the  $i$ -th column in  $t$ -th iteration,  $\varepsilon_i^{(t)} = x_i^{(t)} - x_i^*$ . If  $\|\varepsilon_j^{(t)}\| \leq \frac{\lambda_{j+1} - \lambda_j}{8\sqrt{-\lambda_j}}$  for all  $j \leq i$ , then we have  $\|\varepsilon_i^{(t+1)}\| \leq \left(1 - \alpha \frac{\lambda_{i+1} - \lambda_i}{2}\right) \|\varepsilon_i^{(t)}\| + \alpha \sum_{j=1}^{i-1} \frac{2\|A\|^2}{\sqrt{\lambda_j \lambda_i}} \|\varepsilon_j^{(t)}\|$ .

*Proof.* Similar as in previous lemma, we drop the superscript in the proof. We denote the  $i$ -th column of  $X^*$  as  $v_i = x_i^*$ . Using the expression of  $X^*$ , we have  $v_i^\top v_i = -\lambda_i$  and  $v_i v_i^\top = -\lambda_i u_i u_i^\top$  for  $i = 1, \dots, p$ .

Based on the iterative scheme,  $\tilde{x}_i = x_i - \alpha A x_i - \alpha \sum_{j=1}^i x_j x_j^\top x_i$ , there is

$$\begin{aligned} \tilde{\varepsilon}_i &= \tilde{x}_i - v_i = \varepsilon_i - \alpha A(v_i + \varepsilon_i) - \alpha \sum_{j=1}^i (v_j v_j^\top + \varepsilon_j \varepsilon_j^\top + v_j \varepsilon_j^\top + \varepsilon_j v_j^\top)(v_i + \varepsilon_i) \\ &= ((1 + \alpha \lambda_i)I - \alpha A - \alpha v_i v_i^\top - \alpha \sum_{j=1}^i v_j v_j^\top) \varepsilon_i - \alpha \sum_{j=1}^{i-1} v_j v_j^\top \varepsilon_j \\ &\quad - \alpha \sum_{j=1}^i \varepsilon_j \varepsilon_j^\top v_i - \alpha \sum_{j=1}^i (v_j \varepsilon_j^\top + \varepsilon_j v_j^\top) \varepsilon_i - \alpha \sum_{j=1}^i \varepsilon_j \varepsilon_j^\top \varepsilon_i. \end{aligned} \quad (32)$$

Similar as the diagonal form in the proof of Lemma 5.1, the norm of the prefactor of  $\varepsilon_i$  can be bounded as,

$$\left\| (1 + \alpha \lambda_i)I - \alpha A - \alpha v_i v_i^\top - \alpha \sum_{j=1}^i v_j v_j^\top \right\| \leq 1 + \alpha \lambda_i - \alpha \lambda_{i+1}. \quad (33)$$

When we take norm on both sides of (32), there is

$$\begin{aligned} \|\tilde{\varepsilon}_i\| &\leq (1 + \alpha \lambda_i - \alpha \lambda_{i+1}) \|\varepsilon_i\| + \alpha \sum_{j=1}^{i-1} \sqrt{\lambda_i \lambda_j} \|\varepsilon_j\| + \alpha \sqrt{-\lambda_i} \sum_{j=1}^i \|\varepsilon_j\|^2 \\ &\quad + 2\alpha \sum_{j=1}^i \sqrt{-\lambda_j} \|\varepsilon_j\| \|\varepsilon_i\| + \alpha \sum_{j=1}^i \|\varepsilon_j\|^2 \|\varepsilon_i\| \\ &= (1 - \alpha \lambda_{i+1} + \alpha \lambda_i) \|\varepsilon_i\| + 3\alpha \sqrt{-\lambda_i} \|\varepsilon_i\|^2 + \alpha \|\varepsilon_i\|^3 \\ &\quad + \alpha \sum_{j=1}^{i-1} \left[ \sqrt{\lambda_i \lambda_j} \|\varepsilon_j\| + \sqrt{-\lambda_i} \|\varepsilon_j\|^2 + 2\sqrt{-\lambda_j} \|\varepsilon_i\| \|\varepsilon_j\| + \|\varepsilon_j\|^2 \|\varepsilon_i\| \right]. \end{aligned} \quad (34)$$

Denote  $\Delta_j = \lambda_{j+1} - \lambda_j$  as the  $j$ -th eigengap. Due to the assumption  $\|\varepsilon_j\| \leq \frac{\Delta_j}{8\sqrt{-\lambda_j}}$  for all  $1 \leq j \leq i$ , we have

$$\|\tilde{\varepsilon}_i\| \leq \left(1 - \alpha \frac{\Delta_i}{2}\right) \|\varepsilon_i\| + \alpha \sum_{j=1}^{i-1} \left( \sqrt{\lambda_i \lambda_j} + \sqrt{-\lambda_i} \frac{\Delta_j}{8\sqrt{-\lambda_j}} + \sqrt{\lambda_j} \frac{\Delta_i}{4\sqrt{-\lambda_i}} + \frac{\Delta_i \Delta_j}{64\sqrt{\lambda_i \lambda_j}} \right) \|\varepsilon_j\|, \quad (35)$$

where the first term  $\left(1 - \alpha \frac{\Delta_i}{2}\right) \|\varepsilon_i\|$  is bounded in the same way as in the proof of Lemma 5.1. The

summation term can be controlled as follows,

$$\begin{aligned}
\|\tilde{\varepsilon}_i\| &\leq \left(1 - \alpha \frac{\Delta_i}{2}\right) \|\varepsilon_i\| + \alpha \sum_{j=1}^{i-1} \frac{64\lambda_i\lambda_j - 8\lambda_i\Delta_j - 16\lambda_j\Delta_i + \Delta_i\Delta_j}{64\sqrt{\lambda_i\lambda_j}} \|\varepsilon_j\| \\
&\leq \left(1 - \alpha \frac{\Delta_i}{2}\right) \|\varepsilon_i\| + \alpha \sum_{j=1}^{i-1} \frac{116\|A\|^2}{64\sqrt{\lambda_i\lambda_j}} \|\varepsilon_j\| \\
&\leq \left(1 - \alpha \frac{\Delta_i}{2}\right) \|\varepsilon_i\| + \alpha \sum_{j=1}^{i-1} \frac{2\|A\|^2}{\sqrt{\lambda_i\lambda_j}} \|\varepsilon_j\|.
\end{aligned} \tag{36}$$

Here  $\|A\|$  is adopted for simplicity cause all terms in the  $\lambda$ s can be controlled by  $\|A\|$ , thus the lemma is proved.  $\square$

To make the linear convergence more clear, in Theorem 5.3, we investigate the additional term  $2\alpha \frac{\|A\|^2}{\sqrt{\lambda_j\lambda_i}} \|\varepsilon_j\|$  and find it does not harm the linear convergence.

**Theorem 5.3.** *Assume Assumption A is satisfied. Let  $\varepsilon_i^{(t)}$  be the residual of the  $i$ -th column in  $t$ -th iteration,  $\varepsilon_i^{(t)} = x_i^{(t)} - x_i^*$ . If  $\|\varepsilon_j^{(0)}\| \leq \frac{\lambda_{j+1} - \lambda_j}{8\sqrt{-\lambda_j}}$  for all  $j \leq i$ , then there exists a series of polynomials of degree  $i - 1$ ,  $C_i(t)$ , such that  $\|\varepsilon_i^{(t)}\| \leq C_i(t)r_i^t$  for all  $1 \leq i \leq p$ , where  $r_i = 1 - \frac{\alpha}{2} \min_{j \in [1, i]} \{\lambda_{j+1} - \lambda_j\}$ .*

*Proof.* The theorem is proved in an inductive way. First, for  $i = 1$ , Lemma 5.1 shows that  $\|\varepsilon_1^{(t)}\|$  has already satisfied the condition in the theorem and it is not difficult to figure out that  $C_1 = \|\varepsilon_1^{(0)}\|$ . Given  $i \leq p$ , assume that for all  $j < i$ , there exists a non-decreasing polynomial of degree  $j - 1$ ,  $C_j(t)$ , and  $\|\varepsilon_j^{(t)}\| \leq C_j(t)r_j^t$  holds. Here a non-decreasing polynomial means that the polynomial is non-decreasing for  $t \geq 0$ . Denoting  $a_j = \alpha \frac{2\|A\|^2}{\sqrt{\lambda_j\lambda_i}}$ , the inequality in Lemma 5.2 can be bounded as,

$$\|\varepsilon_i^{(t)}\| \leq r_i \|\varepsilon_i^{(t-1)}\| + \sum_{j=1}^{i-1} a_j \|\varepsilon_j^{(t-1)}\| \leq r_i \|\varepsilon_i^{(t-1)}\| + C_{max}(t)r_{i-1}^{t-1}, \tag{37}$$

where  $C_{max}(t) = \sum_{j=1}^{i-1} a_j C_j(t)$  and the relationship  $r_1 \leq \dots \leq r_{i-1}$  is used so that all  $r_j$ s are controlled by  $r_{i-1}$ . Notice that for each  $j$ ,  $a_j$  is positive and  $C_j(t)$  is a non-decreasing polynomial of degree  $j - 1$ .  $C_{max}(t)$  is then a non-decreasing polynomial of degree  $i - 2$ .

Since the inequality above holds for all  $t \geq 1$ , we apply it repeatedly and obtain,

$$\|\varepsilon_i^{(t)}\| \leq r_i^t \|\varepsilon_i^{(0)}\| + \sum_{k=0}^{t-1} r_i^{t-1-k} C_{max}(k) r_{i-1}^k \leq \left( \|\varepsilon_i^{(0)}\| + \frac{t}{r_i} C_{max}(t) \right) r_i^t = C_i(t) r_i^t, \tag{38}$$

where  $C_i(t) = \|\varepsilon_i^{(0)}\| + \frac{t}{r_i} C_{max}(t)$  is a non-decreasing polynomial of degree  $i - 1$ . Hence the theorem is proved by induction.  $\square$



**Corollary 5.4.** *Assume Assumption A is satisfied. Let  $\delta^{(t)}$  be the distance from the global minima in  $t$ -th iteration,  $\delta^{(t)} = \|X^{(t)} - \mathcal{X}^*\|_{\text{F}}$ . If  $\delta^{(0)} \leq \min_{j \in [1, p]} \frac{\lambda_{j+1} - \lambda_j}{8\sqrt{-\lambda_j}}$ , then there exists a polynomial of degree  $p - 1$ ,  $C(t)$ , such that  $\delta^{(t)} \leq C(t)r^t$ , where  $r = 1 - \frac{\alpha}{2} \min_{j \in [1, p]} \{\lambda_{j+1} - \lambda_j\}$ .*

*Proof.* We first notice that for any two distinct points in  $\mathcal{X}^*$ , the smallest distance in F-norm is  $-2\sqrt{\lambda_p}$ , which is larger than the condition of initial residual  $\delta^{(0)} \leq \min_{j \in [1, p]} \frac{\lambda_{j+1} - \lambda_j}{8\sqrt{-\lambda_j}}$ , so for one initial point, it can only be attracted by one stable fixed point. Then using the definition of  $\delta^{(t)}$ , there is

$$\delta^{(t)} = \sqrt{\sum_{i=1}^p \|\varepsilon_i^{(t)}\|^2} \leq \sqrt{\sum_{i=1}^p (C_i(t)r_i^t)^2}. \quad (39)$$

Here the conclusion of Theorem 5.3 is used. Due to the fact  $r_1 \leq \dots \leq r_p = r$  and define  $C(t) = \sum_{i=1}^p C_i(t)$  which is a polynomial of degree  $p - 1$ , there is

$$\delta^{(t)} \leq \sqrt{\sum_{i=1}^p C_i^2(t)r^t} \leq \sum_{i=1}^p C_i(t)r_p^t = C(t)r^t \quad (40)$$

The inequality holds because  $C_i(t)$  is always non-negative. Thus, the conclusion is proved.  $\square$

Before we move onto the next conclusion, we shall notice that the estimation  $\delta^{(t)} \leq C(t)r^t$  satisfies  $\lim_{t \rightarrow \infty} \frac{\delta^{(t+1)}}{\delta^{(t)}} = r$  which is the definition of linear convergence, because for certain polynomial  $C(t)$ , there is always  $\lim_{t \rightarrow \infty} \frac{C(t+1)}{C(t)} = 1$ .

## 6 Implementation Details

In previous sections, we introduce TriOFM algorithms based on the gradient descent method with a constant stepsize and prove their global and local convergence properties. TriOFM can be regarded as a modified iterative scheme of traditional gradient based optimization algorithms. In this section, we explore traditional techniques for accelerating iterations and adapt them to TriOFM. Such techniques include momentum acceleration, stepsize choices, column locking.

### 6.1 Momentum Acceleration

Momentum is a widely-used technique to accelerate gradient descent methods. In traditional gradient descent methods, with the help of momentum, the oscillatory trajectories are much smoothed and the convergence rates become depending on the square root of the condition number rather than the condition number.

Generally, momentum method, instead of using the gradient direction directly as the moving direction, uses an accumulated gradient direction with a discounting parameter  $\beta \in (0, 1]$ , *i.e.*,

$$V^{(t)} = \beta g(X^{(t)}) + (1 - \beta)V^{(t-1)}, \quad (41)$$

where  $V^{(t)}$  denotes the accumulated gradient directions and  $g$  is the gradient function. Then the iteration moves along  $V^{(t)}$  with a stepsize parameter  $\alpha$ , *i.e.*,

$$X^{(t+1)} = X^{(t)} - \alpha V^{(t)}. \quad (42)$$

Since  $V$  can also be regarded as a linear combination of previous moving directions, an explicit way to generalize it to the triangularized method is replacing the gradient function  $g$  by our triangularized direction function either  $g_1$  or  $g_2$ . Hence we get the modified algorithm for TriOFM-(Obj1) as in Algorithm 3. It can be modified for TriOFM-(Obj2) in a similar way.

---

**Algorithm 3** TriOFM-(Obj1)-momentum

---

**Input:** symmetric matrix  $A$ , initial point  $X^{(0)}$ , momentum coefficient  $\beta$  and stepsize  $\alpha$   
 $t = 0$

**while** not converged **do**

$$V^{(t)} = \begin{cases} g_1(X^{(t)}) & \text{if } t = 0 \\ \beta g_1(X^{(t)}) + (1 - \beta)V^{(t-1)} & \text{otherwise} \end{cases}$$

$$X^{(t+1)} = X^{(t)} - \alpha V^{(t)}$$

$$t = t + 1$$


---

Obviously, such a modification will not change the dependency between columns of  $X$ . Hence with this momentum enabled algorithm, Algorithm 3, the first  $i$  columns can be regarded as this algorithm applied on  $X = (x_1 \ x_2 \ \cdots \ x_i)$ . Hence any column of  $X$  still only depends on its previous columns throughout the iterations. However, choosing an efficient momentum parameter  $\beta$  is an art.

Similarly, we can adopt the idea of conjugate gradient (CG) [11] to triangularized algorithms as well. Since CG methods can be regarded as a momentum accelerated algorithm with adaptive momentum parameters, choosing  $\beta$  is avoided. For linear problems with symmetric positive definite matrices, CG method is efficient. The convergence rate depends on the square root of the condition number. In addition to linear problems, CG has also been widely used to solve nonlinear problems. Some success of nonlinear CG in solving eigenvalue problems have already been observed in OMM [4]. A typical non-linear CG method is the Fletcher-Reeves CG (FR-CG) [8], which adopts the following steps per iteration in a single-vector setting:

$$\begin{aligned} \beta^{(t)} &= \frac{g(x^{(t)})^\top g(x^{(t)})}{g(x^{(t-1)})^\top g(x^{(t-1)})}, \\ v^{(t)} &= -g(x^{(t)}) + \beta^{(t)}v^{(t-1)}, \\ x^{(t+1)} &= x^{(t)} + \alpha v^{(t)}. \end{aligned} \tag{43}$$

In a multi-vector setting, *i.e.*, the iterator is a matrix, the formula for  $\beta^{(t)}$  must be modified. One natural way is to extend the single-vector inner product to the multi-vector inner product, *i.e.*,

$$\beta^{(t)} = \frac{\text{tr}(g(X^{(t)})^\top g(X^{(t)}))}{\text{tr}(g(X^{(t-1)})^\top g(X^{(t-1)}))}, \tag{44}$$

where  $X^{(t)}$  denotes the multi-vector iterator. However, this choice of  $\beta^{(t)}$  is unfavorable in the triangularized algorithms. The  $\beta^{(t)}$  as in (44) mixes information of all columns of  $X^{(t)}$ . Hence the dependency of columns in triangularized algorithms will be changed if (44) is used. Applying the algorithm to the first  $i$  columns will not give you the identical iteration comparing to the first  $i$  columns of applying the algorithm to all columns. Numerically, on practical problems, algorithms using (44) show slower convergence than the following strategy in choosing  $\beta$ .

A favorable choice of  $\beta$  for triangularized algorithms is to use different  $\beta^{(t)}$ s for different columns, which is called a columnwise CG here. The parameter for the  $i$ -th column, denoted as  $\beta_i^{(t)}$ , is

calculated as the single-vector setting with  $x_i^{(t)}$  and applied to update  $x_i^{(t)}$  only. The corresponding algorithm for TriOFM-(Obj1) is summarized as Algorithm 4. In Algorithm 4,  $g_i^{(t)}$  and  $v_i^{(t)}$  denote the  $i$ -th column of  $G^{(t)}$  and  $V^{(t)}$  respectively. The algorithm for TriOFM-(Obj2) can be updated accordingly.

---

**Algorithm 4** Columnwise CG for TriOFM-(Obj1)

---

**Input:** symmetric matrix  $A$ , initial point  $X^{(0)}$ , stepsize  $\alpha$   
 $G^{(0)} = g_1(X^{(0)})$   
 $V^{(0)} = -G^{(0)}$   
 $X^{(1)} = X^{(0)} + \alpha V^{(0)}$   
 $t = 1$   
**while** not converged **do**  
     $G^{(t)} = g_1(X^{(t)})$   
    **for**  $i = 1, 2, \dots, p$  **do**  
         $\beta_i^{(t)} = \frac{(g_i^{(t)})^\top g_i^{(t)}}{(g_i^{(t-1)})^\top g_i^{(t-1)}}$   
         $v_i^{(t)} = -g_i^{(t)} + \beta_i^{(t)} v_i^{(t)}$   
     $X^{(t+1)} = X^{(t)} + \alpha V^{(t)}$   
     $t = t + 1$

---

Comparing the per-iteration computational cost, Algorithm 4 remains the same as using (44). While the column dependent  $\beta$  allows a column of  $X^{(t)}$  to be decoupled from all later columns. Hence we preserve the column dependency in triangularized algorithms. As a remark, there is another way in computing the column dependent parameter  $\beta_i^{(t)}$ s, *i.e.*,  $\beta_i^{(t)}$  is calculated via (44) using  $X_i^{(t)}$ . The column dependency is also preserved in this setting. However, in order to avoid the increase of computational cost, the trace operation must be carefully computed in a cumulative way.

## 6.2 Stepsize Choices

In previous sections, we describe algorithms with constant stepsize to simplify the presentation. However, we find that, comparing to the later described linesearch strategy, algorithms with constant stepsize are significantly slower in practice. In this section, we introduce an exact linesearch strategy as the practical choice of stepsize.

The idea of using an exact linesearch has been adopted by several previous work [16, 19, 29]. Since both (Obj1) and (Obj2) we considered in this paper are quartic polynomials of  $X$ , given a direction  $V$ , the exact linesearch can be calculated through minimizing a quartic polynomial. Noticing that we want to minimize a quartic polynomial with a positive leading coefficient, whose global minimum can also be found through solving a cubic polynomial. For (Obj1), the cubic polynomial is of the following form,

$$\begin{aligned}
 \frac{d}{d\alpha} f_1(X + \alpha V) &= \text{tr}(V^\top \nabla f_1(X + \alpha V)) \\
 &= \alpha^3 \text{tr}((V^\top V)^2) + 3\alpha^2 \text{tr}(V^\top V X^\top V) \\
 &\quad + \alpha \text{tr}(V^\top A V + (V^\top X)^2 + V^\top X X^\top V + V^\top V X^\top X) \\
 &\quad + V^\top A X + \text{tr}(V^\top X X^\top X).
 \end{aligned} \tag{45}$$

Solving the expression above would result possibly one, two, or three real roots. Through a basic analysis, we can show that 1) if there is only one real root, then the optimal stepsize is the root; 2) if there are two real roots, then the optimal stepsize is the root with multiplicity one; 3) if there are three real roots, then the optimal stepsize is the one further away from the middle one. Similar calculation and analysis can also be carried out for (Obj2). We omit the details here.

However the above stepsize does not work for TriOFM. Consider a simple case for example. If  $X$  is in the space spanned by the smallest eigenpairs but not the global minimum we want, which means  $X = U_p \sqrt{-\Lambda_p} Q$  with  $Q$  different from the identical matrix, then  $X$  is already a global minimum of (Obj1) and the stepsize  $\alpha$  is zero for the linesearch mentioned above. This simple example shows that the above linesearch strategy is not working properly for TriOFM and we need to find a different strategy for stepsize.

Notice that the exact linesearch computes an  $\alpha$  such that  $\text{tr}(V^\top \nabla f(X + \alpha V)) = 0$ . However, for TriOFM, optimization is based on  $g_1$  or  $g_2$  rather than  $\nabla f_1$  or  $\nabla f_2$ , which means the iteration is not consistency with the exact linesearch. One option to fix the inconsistency is to solve an equation for  $\alpha$

$$\text{tr}(V^\top g(X + \alpha V)) = 0, \quad (46)$$

where  $g$  is either  $g_1$  or  $g_2$ . Such a modification works in practice. However the stepsize calculated in this way also involves information from all columns of  $X^{(t)}$ . Hence the column dependency in TriOFM is destroyed again. Recall the columnwise strategy for CG parameters, which can be adapted to the stepsize as well.

The columnwise stepsize strategy based on exact linesearch can be described as follows. First, considering the stepsize for the first column  $x_1$ , two equations,  $v_1^\top g(x_1 + \alpha v_1) = 0$  and  $v_1^\top \nabla f(x_1 + \alpha v_1) = 0$  are identical. According to the convergence proof in [19], we know that  $x_1$  will converge to  $\pm \sqrt{-\lambda_1} u_1$ , which are desired global minima for  $x_1$ . Now we move on to the stepsize  $\alpha_i$  for the  $i$ -th column  $x_i$ . We can solve either  $\text{tr}(V_i^\top \nabla f(X_i + \alpha_i V_i)) = 0$  or  $\text{tr}(V_i^\top g(X_i + \alpha_i V_i)) = 0$  for  $\alpha_i$ . The former is the same as (45) with  $X$  and  $V$  replaced by  $X_i$  and  $V_i$  respectively. The later can be expressed as again a cubic polynomial of  $\alpha_i$ ,

$$\begin{aligned} p(\alpha_i) = & \alpha_i^3 \text{tr}(V_i^\top V_i \text{triu}(V_i^\top V_i)) \\ & + \alpha_i^2 \text{tr}(V_i^\top V_i \text{triu}(X_i^\top V_i) + V_i^\top V_i \text{triu}(V_i^\top X_i) + V_i^\top X_i \text{triu}(V_i^\top V_i)) \\ & + \alpha_i \text{tr}(V_i^\top A V_i + V_i^\top X_i \text{triu}(V_i^\top X_i) + V_i^\top X_i \text{triu}(X_i^\top V_i) + V_i^\top V_i \text{triu}(X_i^\top X_i)) \\ & + \text{tr}(V_i^\top A X_i + V_i^\top X_i \text{triu}(X_i^\top X_i)). \end{aligned} \quad (47)$$

Using either equation, we are able to avoid  $\alpha_i = 0$  if  $X_i$  stays in the space spanned by eigenvectors while  $X_i$  is not the desired global minima. The convergence using either equation can be shown in a similar inductive way as in previous sections. Regarding the computational cost, since all trace term can be computed in a accumulative way, the computational cost for getting coefficients in (47) for all  $i$  remains the same as getting the coefficients in (45).

According to our numerical experiments, the columnwise stepsize strategy based on exact linesearch significantly outperforms the fixed stepsize, while there is not much difference between solving  $\text{tr}(V_i^\top g(X_i + \alpha V_i)) = 0$  and  $\text{tr}(V_i^\top \nabla f(X_i + \alpha V_i)) = 0$ . Throughout the rest paper, we solve  $\text{tr}(V_i^\top g) = 0$  for stepsize.

### 6.3 Column Locking

In Section 5 we notice that each column has its own convergence rate and later columns converge slower than earlier ones in terms of the analysis. Numerically, we also observe that earlier columns

usually converge faster than later ones. It is waste of computational resources if we update all columns throughout iterations until the last column converges. Hence we introduce column locking technique here to allow early stopping for converged columns.

The idea of column locking has been widely adopted in many traditional eigensolvers, where orthogonalization is applied every a few iterations. However, in orthogonalization-free eigensolvers [4, 19, 29], usual locking technique is not applicable, since all columns are coupled together throughout iterations. TriOFM, very differently, can adopts a column locking technique in a particular ordering. Since the earlier columns in TriOFM are independent of later columns, as long as they converged, we can lock these columns and no further updating is needed for these columns.

In this paper, either of the following criteria can be adopted as the overall stopping criteria:

$$\left\|g(X^{(t)})\right\|_{\text{F}} < \epsilon \quad \text{or} \quad \frac{\|X^{(t)} - \mathcal{X}^*\|_{\text{F}}}{\|\mathcal{X}^*\|_{\text{F}}} < \epsilon. \quad (48)$$

Since any global minimum is of the same F-norm, we denote the value as  $\|\mathcal{X}^*\|_{\text{F}}$  here.

Correspondingly, the  $i$ -th column is locked if all previous 1st to  $(i - 1)$ -th columns are locked and the criteria (corresponding to the overall stopping criteria) for  $i$ -th column is satisfied,

$$\left\|g_i^{(t)}\right\|_{\text{F}} < \frac{\epsilon}{m} \quad \text{or} \quad \frac{\|x_i^{(t)} - x_i^*\|}{\sqrt{-\lambda_i}} < \frac{\epsilon}{m}, \quad (49)$$

where  $g_i^{(t)}$  is the  $i$ -th column of  $g(X^{(t)})$ ,  $x_i^*$  denote the  $i$ -th column of the nearest global minimum, and  $m$  is a given constant. We emphasize again that column locking must be done in order. While, locking multiple contiguous columns in one iteration is allowed. Usually  $m$  should be chosen slightly larger than  $p$ . If  $m$  is too small, the earlier columns do not converge with high accuracy enough, then the convergence of later columns may not be achievable or the convergence rate would be much slower.

## 7 Numerical Results

In this section, we show the efficiency of TriOFM applying to three different groups of matrices, *i.e.*, random matrices with different eigenvalue distributions, a synthetic matrix from density functional theory, and a matrix of Hubbard model under full configuration interaction framework.

In Section 7.1, we first show that TriOFM with a constant stepsize has linear local convergence rate on different random matrices with different eigenvalue distributions, which agrees with our analysis in Section 5. Further in Section 7.1, numerical techniques introduced in Section 6 are adopted and the comparisons against vanilla TriOFM with a constant stepsize show the advantages of using these techniques. Then we apply TriOFM with these techniques enabled to two numerical examples from chemistry in Section 7.2 and Section 7.3. In both examples, TriOFM converges to sparse eigenvectors whereas traditional orthogonalization-free methods fail to recover the sparsity pattern. Regarding the computational cost, TriOFM is, in general, comparable to its non-triangularized counterpart.

Throughout this section, we adopt the following expressions as the stopping criterion and accuracy measurements. For TriOFM, the F-norm of the triangularized gradient is used as the overall stopping criterion together with the column locking status, *i.e.*, the overall algorithm stops if either  $\|g(X^{(t)})\|_{\text{F}}$  is smaller than a tolerance  $\epsilon$  or all columns have been locked. For traditional orthogonalization-free methods, the iteration stops if the F-norm of the gradient is smaller than a tolerance, *i.e.*,  $\|\nabla f(X^{(t)})\|_{\text{F}} < \epsilon$ .

Since we want to calculate the accuracy of both TriOFM and traditional methods under the same measurement, the expression must be valid for both methods. Two measurements are used with focus on the accuracy of eigenvectors and eigenvalues respectively. The first measure of the accuracy is as follows,

$$e_{vec} = \min_{X^* \in \mathcal{X}^*} \frac{\|X - X^*\|_F}{\|X^*\|_F}, \quad (50)$$

where  $\mathcal{X}^*$  denotes the set of all possible global minima of the used algorithm. The second measure more focuses on the eigenvalues,

$$e_{val} = \frac{\left| \text{tr} \left( (X^\top X)^{-1} X^\top A X \right) - \sum_{i=1}^p \lambda_i \right|}{\left| \sum_{i=1}^p \lambda_i \right|}. \quad (51)$$

In addition to these criteria and accuracy measurements defined above, we also define two measurements for computational costs. Since all of our codes are implemented in MATLAB, which favors matrix operations over vector operations, the runtime comparison is not fair especially considering the matrix sizes we used to explore the efficiency in Section 7.1 and the matrix sizes of practical examples in Section 7.2 and Section 7.3 are not extremely large. Hence we introduce two other measurements of computational costs, *i.e.*, *total number of iterations* and *total number of column accesses*. The total number of iterations is self-explanatory. The total number of column accesses aims to provide measurement of two equal quantities, *i.e.*, the number of writing columns of  $X^{(t)}$  to its data structure and the number of multiplying matrix  $A$  to vectors in  $X^{(t)}$ . When the scale of the matrix is huge and  $X^{(t)}$  has sparse structure as in Section 7.3, the columns of  $X^{(t)}$  are stored in special data structures such as hash table, black-red tree, etc, writing to any of these special data structures is one of the computational bottleneck in FCI computations. Another computational bottleneck is multiplying the matrix  $A$  to columns of  $X^{(t)}$ . Without column locking technique, the total number of column accesses is simply the number of iterations multiplying the number of columns in  $X_{(t)}$ . When column locking is enabled, it is the summation of the number of unlocked columns throughout iterations.

## 7.1 Random Matrices

In this section we apply different TriOFM algorithms to random matrices and compare the performance against their non-triangularized counterparts. For testing purpose, we limit the size of matrices being  $n = 500$  and the number desired eigenpairs being  $p = 5$  in Section 7.1.1 and  $p = 10$  in Section 7.1.2. Random matrices are of the form

$$A = U^\top \Lambda U, \quad (52)$$

where  $U$  is a random orthogonal matrix generated by QR factorization of a random matrix with entries sampled from a standard normal distribution independently. Here  $\Lambda$  denotes a diagonal matrix with its elements  $\{\lambda_i\}_{i=1}^n$  generated from three different ways, which are generated as follows,

1. (Uniform)  $\lambda_i = \frac{i-1}{500} - 1$  for  $1 \leq i \leq n$ ;
2. (Logarithm)  $\lambda_i = -\frac{2^{10}}{500} \frac{1}{2^i}$  for  $1 \leq i \leq n$ ;
3. (U-Shape)  $\lambda_1 = -\frac{14}{16}, \lambda_2 = -\frac{10}{16}, \lambda_3 = -\frac{8}{16}, \lambda_4 = -\frac{7}{16}, \lambda_5 = -\frac{5}{16}, \lambda_i = -\frac{1}{16}$  for all  $6 \leq i \leq n$ .

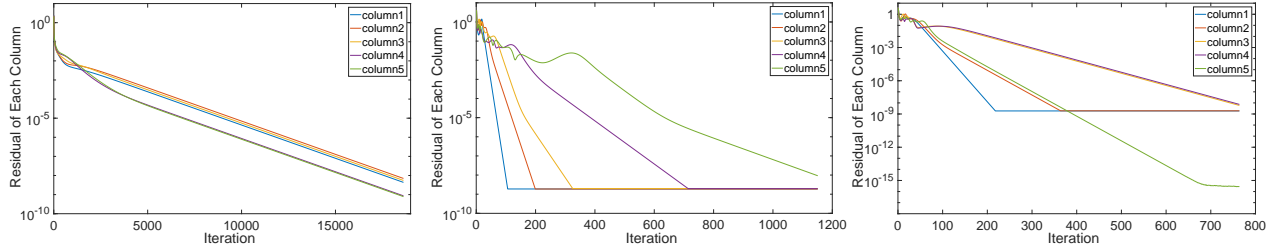


Figure 1: Convergence behavior of TriOFM-(Obj1) applying to  $A_{uni}$  (left),  $A_{log}$  (middle), and  $A_{ushape}$  (right). Fixed step size  $\alpha = 0.4$  is used and column locking is applied.

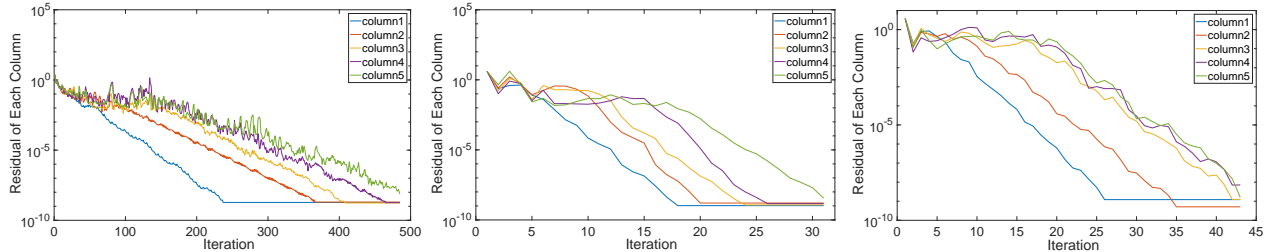


Figure 2: Convergence behavior of TriOFM-(Obj1) applying to  $A_{uni}$  (left),  $A_{log}$  (middle), and  $A_{ushape}$  (right). CG, exact linesearch, and column locking are applied together.

Here, in the U-shape gap case, the first 5 eigengaps are set to be  $\frac{4}{16}, \frac{2}{16}, \frac{1}{16}, \frac{2}{16}, \frac{4}{16}$ , which decays exponentially first and then grows exponentially. We denote these three random matrices as  $A_{uni}$ ,  $A_{log}$ , and  $A_{ushape}$  respectively. Since as shown in Section 5, the convergence rate depends on the eigengap,  $A_{uni}$  and  $A_{log}$  serves as two typical cases of eigengap distribution, where the former has equal eigengap and the later has exponentially decaying eigengap. The last matrix,  $A_{ushape}$  is adopted to reveal the difference between TriOFM and traditional orthogonalization-free methods. The convergence rate of TriOFM depends on the smallest eigengap of all desired eigenpairs, *e.g.*,  $\Delta_3 = \lambda_4 - \lambda_3$  for  $A_{ushape}$ , whereas the rate of traditional methods depends on the eigengap of desired ones and undesired ones, *e.g.*,  $\Delta_5 = \lambda_6 - \lambda_5$  for  $A_{ushape}$ . Since  $\Delta_3$  is exponentially smaller than  $\Delta_5$ , we expect that TriOFM in this case converges slower than traditional methods.

### 7.1.1 Local Convergence Rate

We first numerically validate the local linear convergence as proved in Section 5 and its rate for all three matrices. Momentum or CG is disabled. Fixed stepsize  $\alpha = 0.4$  is used. Each time we sample  $p$  unit vectors independently to form  $X^{(0)} \in \mathbb{R}^{n \times p}$  as the initial value. Column locking technique is applied here.

Figure 1 shows the convergence behaviors of TriOFM-(Obj1) applied to all three matrices. Nonlinear convergence is observed in all three figures for the first few iterations. Linear convergence is observed for every single curve towards convergence. Hence we provide numerical support to the statement that TriOFM-(Obj1) converges linearly in neighborhoods of global minima.

In Figure 1 left, we notice that all curves towards convergence are parallel to each other. Hence numerically the local convergence rates are the same. Since  $A_{uni}$  has all equal eigengaps, as shown in Section 5, the local convergence rates are the same, which agrees with our numerical observation. In Figure 1 middle, curves towards convergence have very different slopes, hence very different convergence rates. Since  $A_{log}$  has exponentially decaying eigengaps, the local convergence rates

Matrix		Convergence rate				
		$\lambda_1$	$\lambda_2$	$\lambda_3$	$\lambda_4$	$\lambda_5$
$A_{log}$	Reference rate	0.7952	0.8976	0.9489	0.9143	0.9872
	Numerical rate	0.7952	0.8976	0.9489	0.9143	0.9872

Table 2: Local convergence rate of TriOFM-(Obj1) applying to  $A_{log}$ .

are indeed different. Here we provide a quantitative comparison of the convergence rates for  $A_{log}$  in Table 2. We fit the slope of each curve after certain number of iterations and use it as the numerical rate. The reference rates (or theoretical rate) are computed according Theorem 5.3,  $r_i = 1 - \alpha \min_{j=1}^i \{\lambda_{j+1} - \lambda_j\}$ . According to Table 2, the numerical rates agree perfectly with reference rate for  $A_{log}$ . Hence we claim that Theorem 5.3 provides a tight upper bound in terms of the convergence rate. In Figure 1 right, first three curves associated with 1st, 2nd, 3rd eigenpairs towards convergence have different convergence rate, and two curves associated with 3rd, 4th eigenpairs are parallel to each other since the convergence behavior of the 4th eigenpair is affected by the smallest eigengap among first 4 gaps which is the same as 3rd eigenpair. Meanwhile, we find that the convergence of the 5th eigenpair is faster than that of the 3rd or 4th one. Hence its convergence rate is theoretically upper bounded by the previous smallest eigengaps but numerically faster. Therefore, through numerical results for all three matrices, we claim that our theoretical analysis of the local convergence rate provides a tight upper bound for practice.

A more interesting numerical observation is that, for all three matrices, the local linear convergences of all columns have some degree of overlapping iterations, which means they converge simultaneously during part of the iterations. Especially for  $A_{uni}$ , there are a lot of iterations that all curves converge linearly simultaneously. However we should notice that if the eigengaps shrink too fast, there is a delay for the later eigenvectors to converge linearly. This delay can be observed in Figure 1 middle. For  $A_{log}$ , neighbour curves have some iterations of simultaneous linear convergence and the delay of convergence is observed. The performance on  $A_{ushape}$  is a combination of that on  $A_{uni}$  and  $A_{log}$ . In Section 5, we prove the local convergence in a column by column fashion, *i.e.*, the convergence of later column is proved if all earlier columns are close to their global minima. Since only an  $\epsilon$  accuracy is needed for the convergence of previous columns, such a delay of convergence is also reasonably agree with our analysis.

In addition to the fixed stepsize, we also investigate the convergence behaviors of TriOFM-(Obj1) with all techniques enabled applying to three matrices, whose convergence behaviors are included in Figure 2 left, middle, and right for  $A_{uni}$ ,  $A_{log}$ , and  $A_{ushape}$  respectively. We emphasize that the scales of iteration number ( $x$ -axis) are totally different in Figure 1 and Figure 2. Overall, the convergence of TriOFM-(Obj1) with all techniques enabled are much faster than that of vanilla TriOFM-(Obj1). Nonlinear convergence is observed in all three figures throughout iterations. Nearly linear convergence is observed for every single curve towards convergence. The approximate linear convergence rates, however, are very different (in general, faster) from that in Figure 1. We leave the convergence analysis of TriOFM with techniques as future work.

### 7.1.2 Accelerating Techniques

Comparing numerical results in Figure 1 and Figure 2, we already notice significant acceleration due to all techniques we introduced, *i.e.*, CG, exact linesearch, and column locking. In this section, we aim to provide more quantitative comparisons for column locking and momentum acceleration for both TriOFM-(Obj1) and TriOFM-(Obj2).



Specifically in this section, the tolerance  $\epsilon$  used for stopping criteria and column locking is  $10^{-8}$ . And each experiment is repeated 500 times, with different random  $Us$  and initial values. For both the iteration number and the total number of column accesses, we report the mean, max, and min among 500 random tests.

Similar to the observation in previous work [4, 19, 29], we also observe that an algorithm with exact linesearch, in all cases, outperforms the same algorithm with fixed stepsize. Hence we omit the comparison results for exact linesearch technique from the paper and focus on the other two techniques, momentum and column locking.

First we show the advantage of using column locking. This experiment is done on (Obj1) with CG and exact linesearch enabled. Table 3 list the iteration number and the total number of column accesses for TriOFM-(Obj1) applied to  $A_{uni}$  with and without column locking.

Method	Iter Num			Total Col Access		
	Mean	Max	Min	Mean	Max	Min
TriOFM-(Obj1) + CG +locking	673.2	773	579	5139.2	6749	4372
TriOFM-(Obj1) + CG	676.4	931	580	6764.4	9310	5800

Table 3: Performance comparison of TriOFM-(Obj1) applied to  $A_{uni}$  with and without column locking. CG and exact linesearch are enabled.

From Table 3, we observe that the number of iteration is not much affected by the column locking. Especially, the mean of iteration number only reduced by about 3 iterations out of nearly 700 iterations. However, the number of column accesses is significantly reduced. All three numbers, *i.e.*, mean, max and min of the number of column access, are reduced by nearly 25% after applying the column locking. Similar results can be obtained for other matrices and TriOFM-(Obj2) as well. Considering the negligible increase of computational cost, we conclude that the column locking technique significantly reduced the number of column accesses and hence the computational cost of TriOFM.

Then we move on to explore the advantage of momentum and CG techniques. In this part, all algorithms are using their own exact linesearch as stepsize choices. For algorithms with vanilla momentum acceleration, the coefficient are chosen as  $\beta = 0.9$  for (Obj1) and  $\beta = 0.95$  for (Obj2). Although not extensive search for  $\beta$  is done, several different values of  $\beta$  are tested for both objective functions. Among those tested  $\beta$ s, we pick the  $\beta$  for each objective function leading to the fastest convergence. For each objective function, there is a comparison between TriOFM and its non-triangularized version (denoted as OFM) with and without momentum acceleration enabled. Numerical results are summarized in Table 4 and Table 5 for  $A_{uni}$  and  $A_{log}$  respectively.

For matrices with equal eigengaps,  $A_{uni}$ , we find that from Table 4 all TriOFMs are slightly slower than their non-triangularized counterparts. Such a slow down is acceptable since TriOFM converges to eigenvectors directly, which include a finite number of global minima, while its non-triangularized counterpart converges to the eigenspace, which includes infinity number of global minima. For matrices with logarithmic decay eigengaps,  $A_{log}$ , as in Table 5, the comparison between TriOFMs and their non-triangularized counterparts gives opposite results. For all cases in Table 5, TriOFMs converge in less number of iterations and less number of column accesses. When algorithms are applied to  $A_{log}$ , convergence rates are determined by the last eigengap. Hence we anticipate that TriOFM performs no worse than its non-triangularized counterpart. While the results in Table 5 are beyond our theoretical analysis. Our explanation for such results has two-folds. First, since first several columns converge much faster in TriOFM, it makes later columns

Objective Function	Method	Iter Num			Total Col Access		
		Mean	Max	Min	Mean	Max	Min
(Obj1)	TriOFM+CG	673.2	773	579	5139.2	6749	4372
	OFM+CG	325.6	681	217	3255.8	6810	2170
	TriOFM+Momentum	1196.3	1494	1001	8528.9	10877	7328
	OFM+Momentum	428.5	706	390	4285.0	7060	4900
	TriOFM+GD	7157.2	9880	4863	65829.6	89045	46400
	OFM+GD	6910.1	9736	4400	69101.2	97360	44000
(Obj2)	TriOFM+CG	968.8	1144	862	7141.1	9915	6158
	OFM+CG	540.0	1135	298	5400.5	11350	2980
	TriOFM+Momentum	1783.2	2116	1588	13336.1	15837	12227
	OFM+Momentum	757.3	1281	739	7573.6	12810	7390
	TriOFM+GD	14217.7	19444	9392	129519.9	171456	90160
	OFM+GD	13141.9	19071	8021	131419.3	190710	80210

Table 4: Performance comparison of TriOFM and OFM with and without momentum accelerating techniques for  $A_{uni}$ . Exact linesearch is enabled for all algorithms and column locking is enabled for TriOFM.

move into linear convergence at very early iterations. Second, in TriOFM, different stepsizes from exact linesearches are applied to different columns, whereas OFM uses the stepsize is uniform for all columns, which is impacted by the smallest eigengap. Overall, the computational cost comparison between TriOFMs and their non-triangularized counterparts depends on the eigengap distribution, although TriOFMs aim to solve a more tough problem.

From both Table 4 and Table 5, in everything single comparison, algorithms with momentum accelerations outperform their non-accelerated versions. Further, comparing CG with momentum accelerate, algorithms with CG are at least no slower than their vanilla momentum acceleration with carefully chosen parameter  $\beta$ . If the parameter  $\beta$  is not carefully chosen, algorithms with CG definitely win. Hence we prefer to use CG as momentum acceleration since it does not have an extra hyper-parameter and is faster.

Summarizing all these tests, the comparison of computational costs for TriOFM and its non-triangularized counterpart is not definitive and depends on the eigengap distribution of the matrix. Regarding the accelerating techniques, CG is the best momentum acceleration, and both exact linesearch and column locking accelerate TriOFM. Hence, our best choice is to use TriOFM with CG, exact linesearch, and column locking enabled. As in the later sections, Section 7.2 and Section 7.3, this will be our representative algorithm of TriOFM and we will move our focus onto the sparsity of eigenvectors.

## 7.2 Synthetic Density Functional Theory

In this section we show a synthetic example from DFT computation. We propose a second order differential operator on the domain  $[0, 1]$  with periodic boundary condition,

$$H(x) = -\Delta + V(x), \tag{53}$$

Objective Function	Method	Iter Num			Total Col Access		
		Mean	Max	Min	Mean	Max	Min
(Obj1)	TriOFM+CG	54.8	67	44	415.0	495	340
	OFM+CG	967.7	2716	461	9676.8	27160	4610
	TriOFM+Momentum	51.3	59	44	414.3	463	377
	OFM+Momentum	1556.5	2423	968	15565.6	24230	9670
	TriOFM+GD	68.0	80	58	610.1	700	528
	OFM+GD	20593.4	26445	14864	205934.7	264450	148640
(Obj2)	TriOFM+CG	293.0	637	195	1094.0	1628	899
	OFM+CG	1035.2	2239	544	10352.1	22390	5440
	TriOFM+Momentum	934.1	1398	706	2905.0	2985	1264
	OFM+Momentum	1988.0	2986	1265	19879.3	29860	12650
	TriOFM+GD	8384.3	10990	4497	188263.6	22400	10706
	OFM+GD	36126.9	45066	24280	361268.9	450660	242800

Table 5: Performance comparison of TriOFM and OFM with and without momentum accelerating techniques for  $A_{log}$ . Exact linesearch is enabled for all algorithms and column locking is enabled for TriOFM.

where  $-\Delta$  is the Laplace operator denoting the kinetic term and  $V(x)$  is a local potential with four Gaussian potential wells,

$$V(x) = - \sum_{i=1}^4 \alpha_i e^{-\frac{(x-\ell_i)^2}{2\sigma^2}}. \quad (54)$$

In (54), the centers of the wells are located at  $\ell_i = \frac{2i-1}{8}$ , the depths of the wells are  $\alpha_i = 850 + 50 \times \text{mod}(i, 4)$ , and the constant width of the wells is  $\sigma = 0.1$ . This second order differential operator, (53), can be viewed as a synthetic linear operator in each self-consistent field iteration in DFT computation, simulating four different atoms located periodically on a line. In this example, we are interested in computing the low-lying four eigenpairs. The matrix form of (53) is obtained via discretizing the problem on a uniform grid with  $n = 500$  points. The Laplace operator is discretized with second-order central difference scheme. In Figure 3 left, we plot the four eigenvectors corresponding to smallest four eigenvalues. Due to the localized potential and periodicity, the eigenvectors associated with low-lying eigenvalues have localized property as well, which means that these vectors are sparse.

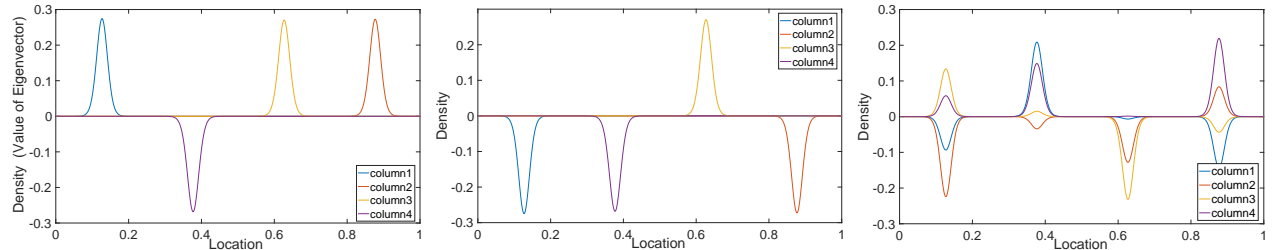


Figure 3: Left figure plots the ground truth of eigenvectors associated with four low-lying eigenvalues of problem (53); middle figure plots scaled four convergent columns from TriOFM-(Obj1); and right figure plots scaled four convergent columns from the non-triangularized counterpart.

Method	Iter Num	Total Column Access	NNZ	$e_{vec}$	$e_{val}$
TriOFM+CG	665.1	2531.5	178	$1.30 \times 10^{-11}$	$1.17 \times 10^{-15}$
OFM+CG	401.3	1601.2	660.0	–	$1.35 \times 10^{-11}$

Table 6: Performance comparison of TriOFM-(Obj1) and its non-triangularized counterpart applied to (53).

Numerical results are demonstrated in Figure 3 and Table 6 over 100 runs. Figure 3 middle plots scaled four convergent columns from TriOFM-(Obj1) and the right figure plots scaled four convergent columns from the non-triangularized counterpart. Table 6 further includes the iteration numbers, total numbers of column accesses, sparsity counts, and accuracy measurements for both algorithms. In Table 6, the NNZ (number of non-zeros) is measured as the number of entries with absolute values greater than  $10^{-5}$ . For TriOFM case, NNZ is 178 which is exactly equal to that of reference eigenvectors. The non-triangularized algorithm does not provide eigenvectors without extra orthogonalization step. Hence the accuracy measurement of eigenvectors is not available.

According to Figure 3, the convergent columns of TriOFM-(Obj1) recover the eigenvectors as in the left plot up to a sign difference. While the convergent columns of non-triangularized algorithm mixes all four eigenvectors and have non-zero peaks near all four Gaussian centers. Hence the sparsity pattern of eigenvectors are destroyed. The sparsity column in Table 6 further provides quantitative comparison. TriOFM-(Obj1) has about 75% less non-zero entries after thresholding. We claim this saving by a factor of 4 is very important from the memory cost point of view. As the problem size getting larger and larger, the memory cost is the key bottleneck in many computations. In our opinion, saving the memory cost by a factor of 4 is more important than saving the computational cost by the same factor, since the former leads to infeasibility and the later leads to a longer runtime. Meanwhile, the number of iterations and total number of column accesses stay similar (within a factor of 1.5) to TriOFM-(Obj1) and its non-triangularized counterpart. Hence, TriOFM is potentially a valuable replacement of non-triangularized counterpart in DFT computations. If some metallic systems are considered in DFT, where eigenvalues are explicitly needed for Fermi-Dirac function, TriOFM with its orthogonalization free property would be more valuable.

### 7.3 Full Configuration Interaction

This section solves the low-lying eigenpairs for two dimensional Hubbard model under FCI framework. The fermion Hubbard model is widely used approximate model in solid-state physics, which ignores long range interaction and only includes near site hopping on lattice. Since the FCI framework is applied, the matrix size scales exponentially with respect to the lattice size and the number of electrons in the Hubbard model. The eigenvectors associated with low-lying eigenvalues are in general very sparse. This is one of the most important target applications of TriOFM.

The Hamiltonian operator in the second quantization notation is,

$$\hat{H} = -t \sum_{\langle r,r' \rangle, \sigma} \hat{a}_{r,\sigma}^\dagger \hat{a}_{r',\sigma} + U \sum_r \hat{a}_{r,\uparrow}^\dagger \hat{a}_{r,\uparrow} \hat{a}_{r,\downarrow}^\dagger \hat{a}_{r,\downarrow} \quad (55)$$

where  $t$  is the hopping strength,  $U$  is the interaction strength,  $r, r'$  are lattice index,  $\langle r, r' \rangle$  means that  $r$  and  $r'$  are neighbor on lattice. Further,  $\hat{a}_{r,\sigma}^\dagger$  and  $\hat{a}_{r,\sigma}$  denotes the creation and annihilation operator of an electron with spin  $\sigma$  on site  $r$ .

In stead of working on the real space, our matrix in this section is generated from the Hubbard model in momentum space, where the hopping term is then a diagonal matrix. The Fourier transform of the creation and annihilation operator is  $\hat{a}_{k,\sigma} = \frac{1}{\sqrt{N^{orb}}} \sum_r e^{ik \cdot r} \hat{a}_{r,\sigma}$ , where  $k$  is the wave number and  $N^{orb}$  is the number of orbitals (sites). The Hamiltonian operator in momentum space is then,

$$\hat{H} = t \sum_{k,\sigma} -2(\cos k_1 + \cos k_2) \hat{a}_{k,\sigma}^\dagger \hat{a}_{k,\sigma} + \frac{U}{N^{orb}} \sum_{k,p,q} \hat{a}_{p-q,\uparrow}^\dagger \hat{a}_{k+q,\downarrow} \hat{a}_{k,\downarrow} \hat{a}_{p,\uparrow} \quad (56)$$

where  $k = (k_1, k_2)$ .

The 2D Hubbard model we used in this section is on a lattice of size  $4 \times 4$  with 6 electrons (3 spin up and 3 spin down). The strength of hopping and interaction are  $t = 1$  and  $U = 0.25N^{orb}$ . Hence the FCI matrix has diagonal entries between  $-20$  and  $20$  and off-diagonal entries being  $\pm 0.25$ . We compute the smallest  $p = 10$  eigenpairs. TriOFM-(Obj1) and its non-triangularized counterpart are applied to address this problem. The tolerance of convergence is set as  $10^{-10}$ . Both CG and exact linesearch are enabled for both algorithms. And column locking is enabled for TriOFM-(Obj1). For each algorithm, we test 100 random initializations and report the mean of the iteration number, the total number of column accesses, the sparsity, and accuracies. The sparsity counts the number of entries with magnitude greater than  $10^{-5}$ . All numerical results are reported in Table 7.

Method	Iter Num	Total Col Access	NNZ	$e_{vec}$	$e_{val}$
TriOFM+CG	718.2	4305.0	57901.4	$2.07 \times 10^{-8}$	$3.44 \times 10^{-15}$
OFM+CG	466.5	4655.3	75041.2	–	$3.43 \times 10^{-15}$

Table 7: Performance comparison of TriOFM-(Obj1) and its non-triangularized counterpart applied to (56)

According to Table 7, although TriOFM-(Obj1) requires larger number of iterations than its non-triangularized counterpart, the total number of column accesses remains similar for both algorithms, which is proportional to the actual runtime. For the matrix in this experiment, there are several eigenvalues of multiplicity larger than 1, and the stable fixed points form a subspace. Hence, NNZ for TriOFM is not a constant and many change due to different rotations within each subspace. Also, the sparsity of TriOFM-(Obj1) is lower than that of its non-triangularized counterpart. Hence, TriOFM-(Obj1) outperforms its non-triangularized counterpart on this FCI problem again. We add a few more discussion for applying TriOFM to FCI problems below.

In FCI problems, eigenvalues and eigenvectors are needed for the ground state and low-lying excited states. Hence non-triangularized algorithms need an explicit post orthogonalization step. However, since FCI problems are usually of extremely large scale, the post orthogonalization step is often computationally too expensive to be practical. Therefore, TriOFM is, to authors' best knowledge, only orthogonalization free family of eigensolvers for eigenpairs. Direct comparison of TriOFM-(Obj1) with its non-triangularized counterpart is not fair in this setting, since the later does not provide needed results. Another important thing is the sparsity. Again, due to the extremely large scale of the problems, the memory cost is the key bottleneck in practice. In order to save memory, we would like to recover the sparsity pattern of eigenvectors as much as possible throughout iterations. Although, in this paper, we only investigate the sparsity of convergent point, adding thresholding together with coordinate-wise descent algorithm would potentially give sparse pattern throughout iterations. Hence this is the first step towards practical algorithms for excited states problem under FCI framework.

## 8 Conclusion and Discussion

In this work, we introduce the novel TriOFM framework for solving extreme eigenvalue problems. Under TriOFM framework, the eigenpairs are directly solved via orthogonalization-free iterative methods, where the orthogonalization-free feature is crucial for extremely large scale eigenvalue problems with sparse eigenvectors. Two algorithms, namely TriOFM-(Obj1) and TriOFM-(Obj2), under the framework are proposed for (Obj1) and (Obj2). Global convergence for both algorithms are guaranteed for almost all initial values in a big domain. Locally, we prove that, in neighbors of global minima, TriOFM-(Obj1) converges linearly. The local convergence proof can also be adapted to TriOFM-(Obj2). Although, under the framework, our proposed algorithms are different from gradient-based optimization algorithms, acceleration techniques, including momentum, linesearch, and column locking, still work effectively. According to a sequence of numerical experiments on both synthetic examples and practical examples, our algorithms converge efficiently and reveals the sparsity of eigenvectors without any orthogonalization step.

There are many future directions. As has been mentioned before, TriOFM framework is not only applicable to (Obj1) and (Obj2). It can be adopted by other algorithms to remove the gauge freedom. We would like to explore more such algorithms and potential applications in the future. Moreover, we claim the advantage of TriOFM in keeping sparsity towards convergent. It is a interesting future direction to explore truncation techniques as well as coordinate-wise algorithms so that the sparse property can be kept throughout iterations. Then the application to FCI low-lying excited states problems would be potentially of great interest to many other communities including computational physics, computational chemistry, and material science, etc. In addition to above two directions, orthogonalization-free algorithms are friendly to massive parallel computing. Hence the parallelization of these proposed algorithms is another future direction.

**Acknowledgments.** The authors thank Jianfeng Lu and Zhe Wang for helpful discussions. YL is supported in part by the US Department of Energy via grant de-sc0019449. WG is supported in part by National Science Foundation of China under Grant No. 11690013, U1811461.

## References

- [1] Banerjee, A. S., Lin, L., Hu, W., Yang, C., and Pask, J. E. (2016). Chebyshev polynomial filtered subspace iteration in the discontinuous Galerkin method for large-scale electronic structure calculations. *J. Chem. Phys.*, 145(15):154101.
- [2] Bottou, L., Curtis, F. E., and Nocedal, J. (2018). Optimization methods for large-scale machine learning. *SIAM Rev.*, 60(2):223–311.
- [3] Brouder, C., Panati, G., Calandra, M., Mourougane, C., and Marzari, N. (2007). Exponential localization of Wannier functions in insulators. *Phys. Rev. Lett.*, 98(4):046402.
- [4] Corsetti, F. (2014). The orbital minimization method for electronic structure calculations with finite-range atomic basis sets. *Comput. Phys. Commun.*, 185(3):873–883.
- [5] Dai, X., Wang, Q., and Zhou, A. (2019a). Gradient flow based discretized Kohn-Sham density functional theory. <http://arxiv.org/abs/1907.06321>.
- [6] Dai, X., Zhang, L., and Zhou, A. (2019b). Adaptive step size strategy for orthogonality constrained line search methods. <http://arxiv.org/abs/1906.02883>.

- [7] Davidson, E. R. (1975). The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices. *J. Comput. Phys.*, 17(1):87–94.
- [8] Fletcher, R. and Reeves, C. M. (1964). Function minimization by conjugate gradients. *Comput. J.*, 7(2):149–154.
- [9] Gao, B., Liu, X., Chen, X., and Yuan, Y. X. (2018). A new first-order algorithmic framework for optimization problems with orthogonality constraints. *SIAM J. Optim.*, 28(1):302–332.
- [10] Gao, B., Liu, X., and Yuan, Y.-x. (2019). Parallelizable algorithms for optimization problems with orthogonality constraints. *SIAM J. Sci. Comput.*, 41(3):A1949–A1983.
- [11] Golub, G. H. and Van Loan, C. F. (2013). *Matrix Computations*. The Johns Hopkins University Press, 4th edition.
- [12] Huang, W., Gallivan, K. A., and Absil, P.-A. (2015). A Broyden class of quasi-Newton methods for Riemannian optimization. *SIAM J. Optim.*, 25(3):1660–1685.
- [13] Knowles, P. J. and Handy, N. C. (1984). A new determinant-based full configuration interaction method. *Chem. Phys. Lett.*, 111(4-5):315–321.
- [14] Knyazev, A. V. (2001). Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM J. Sci. Comput.*, 23(2):517–541.
- [15] Lee, J. D., Panageas, I., Piliouras, G., Simchowitz, M., Jordan, M. I., and Recht, B. (2019). First-order methods almost always avoid strict saddle points. *Math. Program.*, 176(1-2):311–337.
- [16] Lei, Q., Zhong, K., and Dhillon, I. S. (2016). Coordinate-wise power method. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Adv. Neural Inf. Process. Syst.* 29, pages 2064–2072. Curran Associates, Inc.
- [17] Li, Y. and Lu, J. (2019). Bold diagrammatic Monte Carlo in the lens of stochastic iterative methods. *Trans. Math. Its Appl.*, 3(1):1–17.
- [18] Li, Y. and Lu, J. (2020). Optimal orbital selection for full configuration interaction (OptOrbFCI): Pursuing basis set limit under budget. <http://arxiv.org/abs/2004.04205>.
- [19] Li, Y., Lu, J., and Wang, Z. (2019). Coordinatewise descent methods for leading eigenvalue problem. *SIAM J. Sci. Comput.*, 41(4):A2681–A2716.
- [20] Li, Y. and Yang, H. (2017). Spectrum slicing for sparse Hermitian definite matrices based on Zolotarev’s functions. <http://arxiv.org/abs/1701.08935>.
- [21] Liu, X., Wen, Z., and Zhang, Y. (2015). An efficient Gauss-Newton algorithm for symmetric low-rank product matrix approximations. *SIAM J. Optim.*, 25(3):1571–1608.
- [22] Lu, J. and Thicke, K. (2017). Orbital minimization method with l1 regularization. *J. Comput. Phys.*, 336:87–103.
- [23] Lu, J. and Yang, H. (2017). Preconditioning orbital minimization method for planewave discretization. *Multiscale Model. Simul.*, 15(1):254–273.
- [24] Mauri, F., Galli, G., and Car, R. (1993). Orbital formulation for electronic-structure calculations with linear system-size scaling. *Phys. Rev. B*, 47(15):9973–9976.

- [25] Ordejón, P., Drabold, D. A., Grumbach, M. P., and Martin, R. M. (1993). Unconstrained minimization approach for electronic computations that scales linearly with system size. *Phys. Rev. B*, 48(19):14646–14649.
- [26] Peter Tang, P. T. and Polizzi, E. (2014). FEAST as a subspace iteration eigensolver accelerated by approximate spectral projection. *SIAM J. Matrix Anal. Appl.*, 35(2):354–390.
- [27] Stubbs, K. D., Watson, A. B., and Lu, J. (2020). Existence and computation of generalized Wannier functions for non-periodic systems in two dimensions and higher. <http://arxiv.org/abs/2003.06676>.
- [28] Vecharynski, E., Yang, C., and Pask, J. E. (2015). A projected preconditioned conjugate gradient algorithm for computing many extreme eigenpairs of a Hermitian matrix. *J. Comput. Phys.*, 290:73–89.
- [29] Wang, Z., Li, Y., and Lu, J. (2019). Coordinate descent full configuration interaction. *J. Chem. Theory Comput.*, 15(6):3558–3569.
- [30] Wen, Z., Yang, C., Liu, X., and Zhang, Y. (2016). Trace-penalty minimization for large-scale eigenspace computation. *J. Sci. Comput.*, 66(3):1175–1203.
- [31] Wen, Z. and Yin, W. (2013). A feasible method for optimization with orthogonality constraints. *Math. Program.*, 142(1-2):397–434.
- [32] Yu, V. W.-z., Campos, C., Dawson, W., García, A., Havu, V., Hourahine, B., Huhn, W. P., Jacquelin, M., Jia, W., Keçeli, M., Laasner, R., Li, Y., Lin, L., Lu, J., Moussa, J., Roman, J. E., Vázquez-Mayagoitia, Á., Yang, C., and Blum, V. (2019). ELSI – an open infrastructure for electronic structure solvers. <http://arxiv.org/abs/1912.13403>.
- [33] Yu, V. W.-z., Corsetti, F., García, A., Huhn, W. P., Jacquelin, M., Jia, W., Lange, B., Lin, L., Lu, J., Mi, W., Seifitokaldani, A., Vázquez-Mayagoitia, Á., Yang, C., Yang, H., and Blum, V. (2018). ELSI: A unified software interface for Kohn–Sham electronic structure solvers. *Comput. Phys. Commun.*, 222:267–285.
- [34] Zhang, X., Zhu, J., Wen, Z., and Zhou, A. (2014). Gradient type optimization methods for electronic structure calculations. *SIAM J. Sci. Comput.*, 36(3):C265–C289.
- [35] Zhou, Y., Saad, Y., Tiago, M. L., and Chelikowsky, J. R. (2006). Self-consistent-field calculations using Chebyshev-filtered subspace iteration. *J. Comput. Phys.*, 219(1):172–184.

## A Proof of Theorem 3.2

**Proof of Theorem 3.2.** All stationary points of (22) satisfy  $g_2(X) = 0$ . We first analyze the stationary points for a single column case, and then complete the proof by induction. Notations used in this proof are the same as that in the proof of Theorem 3.1.

We denote the single column  $X$  as  $x$ . Obviously, when  $x = 0$ , we have  $g_2(x) = 0$ . Now, consider the nontrivial case  $x \neq 0$ . The equality  $g_2(x) = 0$  can be expanded as,

$$((2 - x^\top x)A - x^\top AxI)x = 0. \tag{57}$$



According to (57), for nonzero  $x$ , the matrix  $B = (2 - x^\top x)A - x^\top AxI$  must have a zero eigenvalue and  $x$  lies in its corresponding eigenspace. When  $x^\top x = 2$ , the matrix  $B = x^\top AxI$  does not have zero eigenvalue due to the negativity assumption on  $A$ . Hence  $x$  is parallel to one of  $A$ 's eigenvector, *i.e.*,  $Ax = \lambda x$ . Substituting this into (57), we obtain,

$$2(1 - x^\top x)\lambda x = 0. \quad (58)$$

Since  $\lambda < 0$  and  $x \neq 0$ , we have  $x^\top x = 1$ . Hence we conclude that for  $g_2(x) = 0$ ,  $x$  is either a zero vector or an eigenvector of  $A$ .

Now we consider multicolumn case. The first column of  $g_2(X) = 0$  is the same as (57). Hence  $X_1 = UP_1S_1D_1$ .

Assume the first  $i$  columns of  $X$  obey  $X_i = UP_iS_iD_i$ . Then the  $(i+1)$ -th column of  $g_2(X) = 0$  is

$$2Ax_{i+1} - Ax_{i+1}x_{i+1}^\top x_{i+1} - x_{i+1}x_{i+1}^\top Ax_{i+1} - AX_iX_i^\top x_{i+1} - X_iX_i^\top Ax_{i+1} = 0. \quad (59)$$

Obviously, if  $x_{i+1} = 0$ , then (59) holds. When  $x_{i+1} \neq 0$ , we left multiply (59) with  $X_i^\top$ , adopt the commuting property of diagonal matrices, and obtain,

$$\begin{aligned} 0 &= D_iS_iP_i^\top (2\Lambda - x_{i+1}^\top x_{i+1}\Lambda - x_{i+1}^\top Ax_{i+1}I - \Lambda P_iP_i^\top - \Lambda) U^\top x_{i+1} \\ &= -D_iS_iP_i^\top (x_{i+1}^\top x_{i+1}\Lambda + x_{i+1}^\top Ax_{i+1}I) U^\top x_{i+1} \end{aligned} \quad (60)$$

where the second equality adopts the fact that  $P_i^\top \Lambda P_iP_i^\top = P_i^\top \Lambda$ . Due to the negativity of  $A$ , we notice that  $x_{i+1}^\top x_{i+1}\Lambda + x_{i+1}^\top Ax_{i+1}I$  is a diagonal matrix with strictly negative diagonal entries. Hence the equality (60) is equivalent to

$$S_iP_i^\top U^\top x_{i+1} = 0. \quad (61)$$

As long as (61) holds, we have  $X_i^\top x_{i+1} = 0$  and  $X_i^\top Ax_{i+1} = 0$ . Therefore, solving (59) can be addressed via solving

$$2Ax_{i+1} - Ax_{i+1}x_{i+1}^\top x_{i+1} - x_{i+1}x_{i+1}^\top Ax_{i+1} = 0 \quad (62)$$

such that  $x_{i+1}$  satisfies (61). Combining the solution of the single column case (57) and the constraint (61), we conclude that  $X_{i+1}$  is of the form  $UP_{i+1}S_{i+1}D_{i+1}$ .

The stabilities of stationary points should also be analyzed through the spectrum properties of their Jacobian matrices. The Jacobian matrix  $Dg_2(X)$ , again, can be written as a  $p$ -by- $p$  block matrix. And using the similar argument as in the proof of Theorem 3.1,  $Dg_2(X) = DG$  is a block upper triangular matrix, whose spectrum is determined by the spectrum of its diagonal blocks. Through a multivariable calculus, we obtain the expression for  $J_{ii}$  as,

$$J_{ii} = 2A - AX_iX_i^\top - X_iX_i^\top A - Ax_i x_i^\top - x_i^\top x_i A - x_i^\top Ax_i I - x_i x_i^\top A. \quad (63)$$

We first show the stability of the stationary points of form  $X = U_p D$ . Substituting these points into (63), we have,

$$J_{ii} = A - 2U_i \Lambda_i U_i^\top - 2\lambda_i u_i u_i^\top - \lambda_i I. \quad (64)$$

Since  $\lambda_i$  is smaller than all eigenvalues of  $A - U_i \Lambda_i U_i^\top$ ,  $A - U_i \Lambda_i U_i^\top - \lambda_i I$  is strictly positive definite. The rest part of (63) is, obviously, positive definite. Hence  $J_{ii}$  is strictly positive definite for all  $i = 1, 2, \dots, p$  and stationary points of the form  $X = U_p D$  are stable stationary points.

Next we show the rest stationary points are not stable. For a stationary point  $X$ , we denote the first index  $s$  such that  $x_s^\top u_s = 0$ . Then we estimate  $u_s^\top J_{ss} u_s$  as,

$$u_s^\top J_{ii} u_s = 2\lambda_s - x_s^\top x_s \lambda_s - x_s^\top A x_s < 0, \quad (65)$$

since  $x_s^\top x_s \leq 1$  and  $A$  is negative definite. Therefore, the rest stationary points are not stable.  $\square$

## B TriOFM Global Convergence Analysis

This section aims to facilitate the proof of Theorem 4.4.

**Proof of Lemma 4.1.** It is sufficient to show that the condition holds for one iteration. In order to simplify the notations, we denote  $x_i^{(t)}$  and  $x_i^{(t+1)}$  as  $x_i$  and  $\tilde{x}_i$  respectively. The norms of  $x_i$  and  $\tilde{x}_i$  are denoted as  $r_i$  and  $\tilde{r}_i$  respectively.

The iteration in Algorithm 1 can be written as,

$$\tilde{x}_i = x_i - \alpha A x_i - \alpha \left( \sum_{j=1}^i x_j x_j^\top \right) x_i = x_i - \alpha \tilde{A} x_i - \alpha x_i x_i^\top x_i. \quad (66)$$

where  $\tilde{A} = A + \sum_{j=1}^{i-1} x_j x_j^\top$ . The norm square of  $\tilde{x}_i$  can be calculated as,

$$\begin{aligned} \tilde{r}_i^2 &= \tilde{x}_i^\top \tilde{x}_i = x_i^\top x_i - 2\alpha \left( x_i^\top \tilde{A} x_i + (x_i^\top x_i)^2 \right) + \alpha^2 \left( x_i^\top \tilde{A}^\top \tilde{A} x_i + (x_i^\top x_i)^3 + 2x_i^\top \tilde{A} x_i (x_i^\top x_i) \right) \\ &= r_i^2 - 2\alpha \left( x_i^\top \tilde{A} x_i + r_i^4 \right) + \alpha^2 \left( x_i^\top \tilde{A}^\top \tilde{A} x_i + r_i^6 + 2x_i^\top \tilde{A} x_i r_i^2 \right). \end{aligned} \quad (67)$$

Given that all  $x_i$  satisfy the conditions  $\|x_i\| \leq R_i$ , we have inequality for any vector  $x$  and power  $k$ ,

$$- \left( \rho + \sum_{j=1}^{i-1} R_j^2 \right)^k \|x\|^2 \leq x^\top \tilde{A}^k x \leq \left( \rho + \sum_{j=1}^{i-1} R_j^2 \right)^k \|x\|^2, \quad (68)$$

where we adopt the definition of eigenvalues for the first part in  $\tilde{A}$  and Cauchy-Schwartz inequality for the second part in  $\tilde{A}$ . Due to the assumption on  $R_j$ , we can bound the factor in the right part of (68) as,

$$\rho + \sum_{j=1}^{i-1} R_j^2 \leq \rho \left( 1 + 3 \sum_{j=1}^{i-1} 4^{j-1} \right) \leq 4^{i-1} \rho = \frac{R_i^2}{3}. \quad (69)$$

With these inequalities, the first order term of  $\alpha$  can be estimated as,

$$\begin{aligned} -2\alpha \left( x_i^\top \tilde{A} x_i + r_i^4 \right) &\leq 2\alpha \left( \left( \rho + \sum_{j=1}^{i-1} R_j^2 \right) r_i^2 - r_i^4 \right) \\ &\leq 2\alpha r_i^2 \left( \frac{R_i^2}{3} - r_i^2 \right), \end{aligned} \quad (70)$$

which is due to (68) and (69). And the second order term of  $\alpha$  can be estimated as,

$$\begin{aligned} \alpha^2 \left( x_i^\top \tilde{A}^2 x_i + r_i^6 + 2x_i^\top \tilde{A} x_i r_i^2 \right) &\leq \alpha^2 \left( \left( \rho + \sum_{j=1}^{i-1} R_j^2 \right)^2 r_i^2 + r_i^6 + 2 \left( \rho + \sum_{j=1}^{i-1} R_j^2 \right) r_i^4 \right) \\ &\leq \alpha^2 r_i^2 \left( \frac{R_i^4}{9} + r_i^4 + \frac{2R_i^2}{3} r_i^2 \right), \end{aligned} \quad (71)$$

which is again due to (68) and (69). The rest of the proof is divided into two scenarios, *i.e.*,  $r_i \in [\frac{\sqrt{2}R_i}{\sqrt{3}}, R_i]$  and  $r_i \in [0, \frac{\sqrt{2}R_i}{\sqrt{3}})$ .

In the first scenario,  $r_i \in [\frac{\sqrt{2}R_i}{\sqrt{3}}, R_i]$ , we have  $-2\alpha \left( x_i^\top \tilde{A}x_i + r_i^4 \right) \leq -\alpha r_i^2 \frac{2R_i^2}{3}$  for the first order term. Applying  $\alpha \leq \frac{1}{5R_p^2}$ , we have,

$$\tilde{r}_i^2 \leq r_i^2 + \alpha r_i^2 \left( -\frac{2R_i^2}{3} + \frac{R_i^4}{45R_p^2} + \frac{R_i^4}{5R_p^2} + \frac{2R_i^4}{15R_p^2} \right) \leq r_i^2 - \alpha r_i^2 \frac{14R_i^2}{45} < r_i^2 \leq R_i^2. \quad (72)$$

In the second scenario,  $r_i \leq \sqrt{\frac{2}{3}}R_i$ , with bounds on the first and second order term, we have

$$\begin{aligned} \tilde{r}_i^2 &\leq r_i^2 + 2\alpha r_i^2 \left( \frac{R_i^2}{3} - r_i^2 \right) + \alpha^2 r_i^2 \left( \frac{R_i^4}{9} + r_i^4 + \frac{2R_i^2}{3} r_i^2 \right) \\ &\leq \frac{2R_i^2}{3} + \frac{4R_i^4}{45R_p^2} + \frac{2R_i^6}{675R_p^4} + \frac{8R_i^6}{675R_p^4} + \frac{8R_i^6}{675R_p^4} < R_i^2. \end{aligned} \quad (73)$$

This proves the lemma.  $\square$

Now we proved that each column  $x_i$  is trapped by a ball throughout iterations. Next we pave the path to prove Lemma 4.2, which show that the first column always converge to the scaled eigenvector corresponding to the first eigenvalue, to be more specific,  $\pm\sqrt{-\lambda_1}u_1$ .

Let  $\theta^{(t)}$  denotes the acute angle between  $x_1^{(t)}$  and  $\pm u_1$ , we will show that  $\theta^{(t)}$  converges to zero almost surely.

**Lemma B.1.** *Assume Assumption A is satisfied and  $x_1^{(0)}$  is not perpendicular to  $u_1$ . Then the tangent of  $\theta^{(t)} = \angle(x_1^{(t)}, u_1)$  linearly converges to 0, i.e.,  $\tan\theta^{(t+1)} \leq \frac{1-\alpha\lambda_2}{1-\alpha\lambda_1} \tan\theta^{(t)}$ .*

*Proof.* Since the eigenvectors of symmetric matrix  $A$  are orthonormal vectors, both vector 2-norm and the angle  $\theta^{(t)} = \angle(x_1^{(t)}, u_1) = \angle(U^\top x_1^{(t)}, e_1)$  are invariant to orthonormal transform, without loss of generality, we assume that  $A$  is a diagonal matrix with its diagonal being  $\lambda_1, \lambda_2, \dots, \lambda_n$  and the corresponding eigenvectors are  $e_i, 1 \leq i \leq n$ . Following the notation in the proof of Lemma 4.1, we drop the iteration index in the superscript and denote the following iteration variables with  $\tilde{\cdot}$ . The first column of  $X^{(t)}$  iterates as follows,

$$\tilde{x}_1 = (I - \alpha A - \alpha x_1^\top x_1 I) x_1. \quad (74)$$

Let  $x_{1i}$  and  $\tilde{x}_{1i}$  denotes the  $i$ -th element of  $x_1$  and  $\tilde{x}_1$  respectively. The  $i$ -th element of  $\tilde{x}_1$  then satisfies,

$$\tilde{x}_{1i} = (1 - \alpha\lambda_i - \alpha r_1^2) x_{1i}, \quad (75)$$

where  $r_1$  denotes the norm of  $x_1$ .

Since  $\tilde{\theta}$  is the angle between  $\tilde{x}_1$  and  $e_1$ , the tangent of  $\tilde{\theta}$  can be written in terms of elements of  $\tilde{x}_1$  as,

$$\tan \tilde{\theta} = \frac{\sqrt{\tilde{x}_{12}^2 + \tilde{x}_{13}^2 + \dots + \tilde{x}_{1n}^2}}{|\tilde{x}_{11}|}. \quad (76)$$

Substituting (75) into (76), we obtain,

$$\begin{aligned} \tan \tilde{\theta} &= \frac{\sqrt{(1 - \alpha\lambda_2 - \alpha r_1^2)^2 x_{12}^2 + \dots + (1 - \alpha\lambda_n - \alpha r_1^2)^2 x_{1n}^2}}{(1 - \alpha\lambda_1 - \alpha r_1^2) |x_{11}|} \\ &\leq \frac{1 - \alpha\lambda_2 - \alpha r_1^2}{1 - \alpha\lambda_1 - \alpha r_1^2} \left( \frac{\sqrt{x_{12}^2 + \dots + x_{1n}^2}}{|x_{11}|} \right) \\ &= \frac{1 - \alpha\lambda_2 - \alpha r_1^2}{1 - \alpha\lambda_1 - \alpha r_1^2} \tan \theta \leq \frac{1 - \alpha\lambda_2}{1 - \alpha\lambda_1} \tan \theta, \end{aligned} \quad (77)$$

where we adopt the assumption on  $\alpha$ , which guarantees the positivity of  $1 - \alpha\lambda_1 - \alpha r_1^2$  and  $1 - \alpha\lambda_2 - \alpha r_2^2$ .

Recursively applying (77), we have  $\tan \theta^{(t)} \leq \left(\frac{1-\alpha\lambda_2}{1-\alpha\lambda_1}\right)^t \tan \theta^{(0)}$ , which implies  $\theta^{(t)} \rightarrow 0$  as  $t \rightarrow \infty$ . □

Although we have linear convergence for the tangent of the angle between  $x_1^{(t)}$  and  $u_1$ ,  $\tan \theta^{(0)}$  can be huge if  $x_1^{(0)}$  is nearly orthogonal to  $u_1$ . Next, we would focus on the convergence of the vector length.

**Lemma B.2.** *Assume Assumption A is satisfied and  $x_1^{(0)}$  is not perpendicular to  $u_1$ . Then there exists an integer  $N$  such that  $\|x_1^{(t)}\| \geq \frac{\sqrt{-2\lambda_q}}{4}$  holds for all  $t \geq N$ .*

*Proof.* Without loss of generality, we assume that  $A$  is a diagonal matrix with its diagonal being  $\lambda_1, \lambda_2, \dots, \lambda_n$  and the corresponding eigenvectors are  $e_i, 1 \leq i \leq n$ . Following the notation in the proof of Lemma 4.1, we drop the iteration index in the superscript and denote the following iteration variables with  $\tilde{\cdot}$ . Let  $x_{1i}$  and  $\tilde{x}_{1i}$  denote the  $i$ -th element of  $x_1$  and  $\tilde{x}_1$  respectively. The  $i$ -th element of  $\tilde{x}_1$  then satisfies,

$$\tilde{x}_{1i} = (1 - \alpha\lambda_i - \alpha r_1^2) x_{1i}, \quad (78)$$

where  $r_1$  denotes the norm of  $x_1$ . Further we split  $x_1$  into two vectors as  $x_1 = (y_1^\top \ y_2^\top)^\top$  where  $y_1 = (x_{11} \ \dots \ x_{1q})^\top$  and  $y_2 = (x_{1(q+1)} \ \dots \ x_{1n})^\top$ .

The proof consists two parts. In the first part, we show that there exist an iteration  $N_1$ , such that  $\|y_1^{(N_1)}\| \geq \frac{\sqrt{-2\lambda_q}}{2}$  and  $\|y_2^{(N_1)}\| \leq \frac{\sqrt{-2\lambda_q}}{4}$ . In the second part, we show that as long as the condition in the first part is satisfied, the length of  $x_1$  never reduces below  $\frac{\sqrt{-2\lambda_q}}{4}$ .

Notice that for any  $r_1$  we have,

$$|\tilde{x}_{1i}| \leq (1 - \alpha\lambda_i - \alpha(x_{1i})^2) |x_{1i}| \quad (79)$$

for  $i = q+1, \dots, n$ .  $|x_{1i}|$  decays monotonically to zero. Hence there exists an integer  $M$  such that for any  $t \geq M$  we have  $\|y_2^{(t)}\| \leq \frac{\sqrt{-2\lambda_q}}{4}$ . Further, for  $t \geq M$ , if  $\|y_1\| \leq \frac{\sqrt{-2\lambda_q}}{2}$ , we have,

$$|\tilde{x}_{1i}| \geq \left(1 - \alpha\lambda_i + \alpha \frac{5\lambda_q}{8}\right) |x_{1i}|, \quad (80)$$

for  $i = 1, \dots, q$ , where the increasing factors are strictly greater than one. Also we have  $x_{11}^{(0)} = x_1^\top u_1 \neq 0$  in the assumption. Hence there exists a integer  $N \geq M$  such that  $\|y_1^{(N)}\| \geq \frac{\sqrt{-2\lambda_q}}{2}$  and  $\|y_2^{(N)}\| \leq \frac{\sqrt{-2\lambda_q}}{4}$ .

Next, we show that for any  $t \geq N$ , if  $\|y_1\| \geq \frac{\sqrt{-2\lambda_q}}{2}$ , then we have,

$$\|\tilde{y}_1\| \geq (1 - \alpha r_1^2) \|y_1\| \geq \left(1 - \frac{R_1^2}{5R_p^2}\right) \|y_1\| \geq \frac{\sqrt{-2\lambda_q}}{4}, \quad (81)$$

where we adopt Lemma 4.1 and the assumption on  $\alpha$  in the second inequality. Such a relation means that as long as  $\|y_1\| \geq \frac{\sqrt{-2\lambda_q}}{2}$ , the length cannot reduce below  $\frac{\sqrt{-2\lambda_q}}{4}$  in one iteration.

When  $\frac{\sqrt{-2\lambda_q}}{4} \leq \|y_1\| \leq \frac{\sqrt{-2\lambda_q}}{2}$  and  $\|y_2\| \leq \frac{\sqrt{-2\lambda_q}}{4}$ , we have,

$$\|\tilde{y}_1\| \geq (1 - \alpha\lambda_q - \alpha r_1^2)\|y_1\| \geq \left(1 - \frac{3}{8}\alpha\lambda_q\right)\|y_1\| \geq \frac{\sqrt{-2\lambda_q}}{4}. \quad (82)$$

Hence, as long as  $t \geq N$ , we have  $\|x_1^{(t)}\| \geq \|y_1^{(t)}\| \geq \frac{\sqrt{-2\lambda_q}}{4}$ .

□

**Proof of Lemma 4.2.** Similar as in the proofs of previous lemmas, without loss of generality, we assume that  $A$  is a diagonal matrix with its diagonal being  $\lambda_1, \lambda_2, \dots, \lambda_n$  and the corresponding eigenvectors are  $e_i, 1 \leq i \leq n$ . All notations follow that in the proofs of previous lemmas.

According to Lemma B.1, the tangent of  $\theta$  converges to zero, *i.e.*,

$$\tan \theta = \frac{\sqrt{x_{12}^2 + x_{13}^2 + \dots + x_{1n}^2}}{|x_{11}|} \rightarrow 0. \quad (83)$$

Lemma 4.1 implies the boundedness of  $x_1$ , which implies the boundedness of  $x_{11}$ . Hence we have,

$$\sqrt{x_{12}^2 + x_{13}^2 + \dots + x_{1n}^2} \rightarrow 0. \quad (84)$$

To simplify the notation, we denote  $\eta$  as  $\eta = \sqrt{x_{12}^2 + x_{13}^2 + \dots + x_{1n}^2}$ . The convergence of  $\eta$  can be stated as follows. For any  $\varepsilon \leq \min\left(\frac{\sqrt{-\lambda_q}}{4}, \frac{\sqrt{\lambda_1\lambda_q}}{8R_1}\right)$ , there exists an integer  $N_1$  such that for any  $t \geq N_1$ , we have  $\eta^2 \leq \varepsilon^2$ . Also recall Lemma B.2, there exists an integer  $N_2$ , such that for any  $t \geq N_2$ , we have  $\|x_1^{(t)}\| \geq \frac{\sqrt{-2\lambda_q}}{4}$ . Combining the bounds on  $\eta$  and  $\|x_1^{(t)}\|$ , we obtain, when  $t \geq M = \max(N_1, N_2)$ ,  $(x_{11}^{(t)})^2 = \|x_1^{(t)}\|^2 - \eta^2 \geq -\frac{\lambda_q}{8} - \varepsilon^2 \geq -\frac{\lambda_q}{16}$ .

Since the stepsize  $\alpha$  guarantees the positivity of  $(1 - \alpha\lambda_1 - \alpha r_1^2)$ ,  $\tilde{x}_{11} = (1 - \alpha\lambda_1 - \alpha r_1^2)x_{11}$  remains the same sign as  $x_{11}$  and the same as  $x_{11}^{(0)}$ . We first discuss the scenario  $x_{11}^{(0)} > 0$ .

Let  $\delta^{(t)} = x_{11}^{(t)} - \sqrt{-\lambda_1}$ . We have the recurrent relationship,

$$\begin{aligned} \delta^{(t+1)} &= x_{11}^{(t+1)} - \sqrt{-\lambda_1} = \left(1 - \alpha \left(\lambda_1 + (x_{11}^{(t)})^2 + (\eta^{(t)})^2\right)\right) x_{11}^{(t)} - \sqrt{-\lambda_1} \\ &= \left(1 - \alpha \left(\sqrt{-\lambda_1} + x_{11}^{(t)}\right) x_{11}^{(t)}\right) \delta^{(t)} - \alpha (\eta^{(t)})^2 x_{11}^{(t)}. \end{aligned} \quad (85)$$

Taking the absolute value of both side, we obtain the recurrent inequality relationship,

$$\begin{aligned} |\delta^{(t+1)}| &\leq \left(1 - \alpha \left(\sqrt{-\lambda_1} + x_{11}^{(t)}\right) x_{11}^{(t)}\right) |\delta^{(t)}| + \alpha (\eta^{(t)})^2 x_{11}^{(t)} \\ &\leq \left(1 - \alpha \frac{\sqrt{\lambda_1\lambda_q}}{4}\right) |\delta^{(t)}| + \alpha \varepsilon^2 R_1 \\ &\leq \dots \\ &\leq \left(1 - \alpha \frac{\sqrt{\lambda_1\lambda_q}}{4}\right)^{t+1-M} |\delta^{(M)}| + \varepsilon^2 \frac{4R_1}{\sqrt{\lambda_1\lambda_q}} \\ &\leq \left(1 - \alpha \frac{\sqrt{\lambda_1\lambda_q}}{4}\right)^{t+1-M} |\delta^{(M)}| + \frac{\varepsilon}{2}. \end{aligned} \quad (86)$$

Hence there exists an integer  $N \geq M$  such that for any  $t \geq N$ ,  $|\delta^{(t)}| \leq \varepsilon$ .

If  $x_{11}^{(0)} < 0$ , the iteration converges to  $-\sqrt{-\lambda_1}$ . The analysis is analogy to the above one. The lemma is proved.  $\square$

After proving the single column case, we turn to the multicolumn case. When we are proving the multicolumn case, we first assume the fact that all previous columns have converged to global minima, *i.e.*,

$$\lim_{t \rightarrow \infty} \left\| X_{k-1}^{(t)} - \mathcal{X}_{k-1}^* \right\|_{\text{F}} = 0, \quad (87)$$

then Theorem 4.4 proves the overall global convergence by induction. In the following, we first prove a few lemmas to support the proof of Lemma 4.3.

**Lemma B.3.** *Assume Assumption A is satisfied and  $\lim_{t \rightarrow \infty} \left\| X_{k-1}^{(t)} - \mathcal{X}_{k-1}^* \right\|_{\text{F}} = 0$ . Then  $\lim_{t \rightarrow \infty} u_i^\top x_k^{(t)} = 0$  for all integer  $i \in [1, k] \cup (q, n]$ .*

*Proof.* First we will introduce some notations. Let  $E$  be the symmetric residual of the first  $k-1$  columns, *i.e.*,  $E^{(t)} = \sum_{i=1}^{k-1} x_i^{(t)} \left( x_i^{(t)} \right)^\top + \lambda_i u_i u_i^\top$ , and  $E_i^{(t)}$  denote the  $i$ -th column of  $E^{(t)}$ . The convergence of  $X_{k-1}^{(t)}$  implies that  $\lim_{t \rightarrow \infty} \|E^{(t)}\|_{\text{F}} = 0$  and hence  $\lim_{t \rightarrow \infty} \|E_i^{(t)}\| = 0$  for any  $i = 1, 2, \dots, n$ . Using the notation  $E^{(t)}$  and (13), the iteration for the  $k$ -th column of  $X^{(t)}$  can be written as,

$$x_k^{(t+1)} = \left( I - \alpha \tilde{A} - \alpha \left( x_k^{(t)} \right)^\top x_k^{(t)} I \right) x_k^{(t)} - \alpha E^{(t)} x_k^{(t)}, \quad (88)$$

where  $\tilde{A} = A - \sum_{i=1}^{k-1} \lambda_i u_i u_i^\top$ .

Similar as in previous proofs, without loss of generality, we assume  $A$  is a diagonal matrix. Then showing the convergence of  $u_i^\top x_k^{(t)}$  is equivalent to showing the convergence of  $e_i^\top x_k^{(t)} = x_{ki}^{(t)}$ . In the following, we consider the convergence of  $x_{ki}^{(t)}$  for  $i \in [1, k] \cup (q, n]$ .

Since we have  $\lim_{t \rightarrow \infty} \|E_i^{(t)}\| = 0$ , for any  $\varepsilon < \sqrt{\frac{2}{\alpha}}$ , there exists  $N$  such that  $\|E_i^{(t)}\| < \frac{\varepsilon^3}{2R_k}$  holds for all  $t \geq N$ . Multiplying  $e_i^\top$  on both sides of (88) for  $i \in [1, k] \cup (q, n]$ , we have the following estimation,

$$\begin{aligned} \left| x_{ki}^{(t+1)} \right| &= \left| \left( 1 - \alpha \left( x_k^{(t)} \right)^\top x_k^{(t)} \right) x_{ki}^{(t)} - \alpha e_i^\top \tilde{A} x_k^{(t)} - \alpha \left( E_i^{(t)} \right)^\top x_k^{(t)} \right| \\ &\leq \left( 1 - \alpha \left( x_{ki}^{(t)} \right)^2 \right) \left| x_{ki}^{(t)} \right| + \alpha \left\| E_i^{(t)} \right\| \left\| x_k^{(t)} \right\| \\ &\leq \left( 1 - \alpha \left( x_{ki}^{(t)} \right)^2 + \frac{\alpha \varepsilon^3}{2 \left| x_{ki}^{(t)} \right|} \right) \left| x_{ki}^{(t)} \right|, \end{aligned} \quad (89)$$

where we adopt the assumption on  $\alpha$  and Cauchy-Schwartz inequality. Notice that when  $\left| x_{ki}^{(t)} \right| > \varepsilon$ , the inequality reduces to  $\left| x_{ki}^{(t+1)} \right| \leq \left( 1 - \frac{\alpha \varepsilon^2}{2} \right) \left| x_{ki}^{(t)} \right|$ , which means  $\left| x_{ki}^{(t)} \right|$  decays exponentially with the factor  $1 - \frac{\alpha \varepsilon^2}{2}$ . On the other hand, if there is a  $t$  such that  $\left| x_{ki}^{(t)} \right| \leq \varepsilon$ , then the quantity in the following iteration is upper bounded by

$$\left| x_{ki}^{(t+1)} \right| \leq (1 + \alpha R_k^2) \varepsilon + \frac{\alpha \varepsilon^3}{2} \leq (2 + \alpha R_k^2) \varepsilon, \quad (90)$$

where the second inequality holds due to  $\varepsilon < \sqrt{\frac{2}{\alpha}}$ .

Hence we conclude that, for any  $\varepsilon$ , there exist a constant  $N' > N$ , such that  $\left| x_{ki}^{(t)} \right| \leq (2 + \alpha R_k^2) \varepsilon$  holds for all  $t \geq N'$ . Thus we have  $\lim_{t \rightarrow \infty} u_i^\top x_k^{(t)} = 0$  for all  $i \in [1, k] \cup (q, n]$ .  $\square$

Lemma B.4 is the multicolumn version of Lemma B.2.

**Lemma B.4.** *Assume Assumption A is satisfied and  $\lim_{t \rightarrow \infty} \left\| X_{k-1}^{(t)} - \mathcal{X}_{k-1}^* \right\|_F = 0$ . If there exists an integer  $i \in [k, q]$  such that  $u_i^\top x_k^{(t)}$  does not converge to zero, then there exists an integer  $N$  such that  $\left\| x_k^{(t)} \right\| \geq \frac{\sqrt{-2\lambda_q}}{4}$  holds for all  $t > N$ .*

*Proof.* Without loss of generality, we assume  $A$  is diagonal. Hence we have  $u_i^\top x_k^{(t)} = e_i^\top x_k^{(t)} = x_{ki}^{(t)}$ . The notation of  $E^{(t)}$  and the same iteration (88) as in the proof of Lemma B.3 are used here.

We split the vector  $x_k^{(t)}$  into three parts:  $y_1^{(t)} = \left( x_{k1}^{(t)}, \dots, x_{k(k-1)}^{(t)} \right)^\top$ ,  $y_2^{(t)} = \left( x_{kk}^{(t)}, \dots, x_{kq}^{(t)} \right)^\top$ , and  $y_3^{(t)} = \left( x_{k(q+1)}^{(t)}, \dots, x_{kn}^{(t)} \right)^\top$ .

From the assumption there exists an integer  $i \in [k, q]$  such that  $u_i^\top x_k^{(t)}$  does not converge to zero. Hence, there exists a positive  $\varepsilon_0 < \frac{\sqrt{-2\lambda_q}}{8}$ , such that for any  $N$  there exists a  $t > N$  and  $\left\| y_2^{(t)} \right\| > \varepsilon_0$  holds. Further, we assume the convergence of  $E^{(t)}$  and Lemma B.3 guarantees the convergence of  $y_1^{(t)}$  and  $y_3^{(t)}$ . Thus for such  $\varepsilon_0$ , there exists an  $N_1$  such that  $\left\| E^{(t)} \right\| \leq \frac{\sqrt{-2\lambda_q} \varepsilon_0}{4}$ ,  $\left\| y_1^{(t)} \right\| < \frac{\sqrt{-2\lambda_q}}{8}$ , and  $\left\| y_3^{(t)} \right\| < \frac{\sqrt{-2\lambda_q}}{8}$  hold for all  $t \geq N_1$ , and  $\left\| y_2^{(N_1)} \right\| > \varepsilon_0$ . Considering the iteration of the  $j$ -th entry of  $x_k^{(t)}$  for  $j \in [k, q]$ , we have,

$$x_{kj}^{(t+1)} = \left( 1 - \alpha \lambda_j - \alpha \left\| x_k^{(t)} \right\|^2 \right) x_{kj}^{(t)} - \alpha e_j^\top E^{(t)} x_k^{(t)}. \quad (91)$$

If  $\left\| x_k^{(t)} \right\| \leq \frac{\sqrt{-2\lambda_q}}{2}$  for  $t = N_1$ , then we can bound the norm of  $y_2^{(t)}$  as,

$$\begin{aligned} \left\| y_2^{(t+1)} \right\| &\geq \left( 1 - \alpha \lambda_q - \alpha \left\| x_k^{(t)} \right\|^2 \right) \left\| y_2^{(t)} \right\| - \alpha \left\| E^{(t)} \right\| \left\| x_k^{(t)} \right\| \\ &\geq \left( 1 - \frac{\alpha \lambda_q}{2} - \alpha \frac{\left\| E^{(t)} \right\| \left\| x_k^{(t)} \right\|}{\left\| y_2^{(t)} \right\|} \right) \left\| y_2^{(t)} \right\| \\ &\geq \left( 1 - \frac{\alpha \lambda_q}{4} \right) \left\| y_2^{(t)} \right\|. \end{aligned} \quad (92)$$

The increasing factor is strictly greater than one. Hence  $\left\| y_2^{(t+1)} \right\| > \varepsilon_0$  holds for  $t + 1$  as well. And  $\left\| y_2^{(t)} \right\|$  increase monotonically until  $\left\| x_k^{(t)} \right\| > \frac{\sqrt{-2\lambda_q}}{2}$ .

When  $\|x_k^{(t)}\| > \frac{\sqrt{-2\lambda_q}}{2}$ , the norm of the following iteration is lower bounded as,

$$\begin{aligned} \|x_k^{(t+1)}\| &\geq (1 - \alpha\lambda_n - \alpha R_k^2) \|x_k^{(t)}\| - \alpha \|E^{(t)}\| R_k \\ &\geq (1 - \alpha\lambda_n - \alpha R_k^2) \frac{\sqrt{-2\lambda_q}}{2} - \alpha \frac{\sqrt{-2\lambda_q}\varepsilon_0}{4} \\ &\geq \frac{\sqrt{-2\lambda_q}}{4}, \end{aligned} \quad (93)$$

where the last inequality is due to the assumption on  $\alpha$ . Further, the norm of  $y_2^{(t+1)}$  can be lower bounded as,

$$\|y_2^{(t+1)}\| \geq \sqrt{\|x_k^{(t+1)}\|^2 - \|y_1^{(t+1)}\|^2 - \|y_3^{(t+1)}\|^2} \geq \sqrt{\frac{-2\lambda_q}{16} + 2\frac{2\lambda_q}{64}} > \varepsilon_0. \quad (94)$$

Therefore, the norm of  $x_k^{(t)}$  is lower bounded by  $\frac{\sqrt{-2\lambda_q}}{4}$  after the first iteration later than  $N_1$  such that  $\|x_k^{(t)}\| > \sqrt{\frac{-2\lambda_q}{2}}$ .  $\square$

Lemma B.5 and Lemma B.6 serve as the multicolumn version of Lemma B.1. More precisely, under the assumption that  $x_k^{(t)}$  does not converge to zero, Lemma B.5 and Lemma B.6 prove that there exists a tangent of  $\theta_k^{(t)} = \angle(x_k^{(t)}, u_k)$  or  $\theta_j^{(t)} = \angle(x_k^{(t)}, u_j)$  for  $j \in (k, q]$  converging linearly to zero, where as before  $\theta_j^{(t)}$  denotes the acute angle between  $x_k^{(t)}$  and  $\pm u_j$  for  $j \in [k, q]$ .

**Lemma B.5.** *Assume Assumption A is satisfied and  $\lim_{t \rightarrow \infty} \|X_{k-1}^{(t)} - \mathcal{X}_{k-1}^*\|_{\mathbb{F}} = 0$ . If  $u_k^\top x_k^{(t)}$  does not converge to zero, then the tangent of  $\theta_k^{(t)} = \angle(x_k^{(t)}, u_k)$  converges to 0.*

*Proof.* Without loss of generality, we assume  $A$  is diagonal. Hence we have  $u_i^\top x_k^{(t)} = e_i^\top x_k^{(t)} = x_{ki}^{(t)}$ . The notation of  $E^{(t)}$  and the same iteration (88) as in the proof of Lemma B.3 are used here.

Based on the assumptions that  $e_k^\top x_k^{(t)} = x_{kk}^{(t)}$  does not converge to zero, there exists a positive number  $\delta > 0$  such that for any  $N$ , there exists a  $t > N$  and  $|x_{kk}^{(t)}| > \delta$ . We also know that Lemma B.4 holds and  $\|E^{(t)}\|$  converges to zero. Hence, for any  $\varepsilon$  sufficiently small, there exists an integer  $N_1$  such that  $\|E^{(t)}\| \leq \varepsilon^2$  and  $\|x_k^{(t)}\| \geq \frac{\sqrt{-2\lambda_q}}{4}$  hold for all  $t > N_1$ . Since  $e_k^\top x_k^{(t)} = x_{kk}^{(t)}$  does not converge to zero, there exists an integer  $N_2 > N_1$ , such that,

$$\frac{|x_{kk}^{(N_2)}|}{\|x_k^{(N_2)}\|} \geq \frac{\delta}{R_k} \geq \frac{2\varepsilon^2}{\lambda_{k+1} - \lambda_k}, \quad (95)$$

where we notice that  $\cos \theta_k^{(N_2)} = \frac{|x_{kk}^{(N_2)}|}{\|x_k^{(N_2)}\|}$ . Since  $\theta_k^{(t)}$  is defined as the acute angle, we have,

$$\tan \theta_k^{(t+1)} = \frac{\sqrt{\sum_{j \neq k} (x_{kj}^{(t+1)})^2}}{|x_{kk}^{(t+1)}|}. \quad (96)$$



Next, we derive lower bound and upper bound for the denominator and numerator respectively when  $t = N_2$ .

Using the iterative relationship (88), we have the lower bound on the denominator,

$$\begin{aligned}
|x_{kk}^{(N_2+1)}| &= \left| \left( 1 - \alpha\lambda_k - \alpha \|x_k^{(N_2)}\|^2 \right) x_{kk}^{(N_2)} - \alpha e_k^\top E^{(N_2)} x_k^{(N_2)} \right| \\
&\geq \left( 1 - \alpha\lambda_k - \alpha \|x_k^{(N_2)}\|^2 - \alpha \frac{\|E^{(N_2)}\| \|x_k^{(N_2)}\|}{|x_{kk}^{(N_2)}|} \right) |x_{kk}^{(N_2)}| \\
&\geq \left( 1 - \alpha \frac{\lambda_k + \lambda_{k+1}}{2} - \alpha \|x_k^{(N_2)}\|^2 \right) |x_{kk}^{(N_2)}|,
\end{aligned} \tag{97}$$

where the second inequality is due to (95).

Regarding the numerator in (96), again using the iterative relationship (88), we have,

$$\begin{aligned}
\sqrt{\sum_{j \neq k} (x_{kj}^{(N_2+1)})^2} &\leq \left\{ \sum_{j \neq k} \left[ \left( 1 - \alpha\lambda_{k+1} - \alpha \|x_k^{(N_2)}\|^2 \right)^2 (x_{kj}^{(N_2)})^2 + \alpha^2 \|E^{(N_2)}\|^2 \|x_k^{(N_2)}\|^2 \right. \right. \\
&\quad \left. \left. + 2\alpha \left( 1 - \alpha\lambda_{k+1} - \alpha \|x_k^{(N_2)}\|^2 \right) |x_{kj}^{(N_2)}| \|E^{(N_2)}\| \|x_k^{(N_2)}\| \right] \right\}^{\frac{1}{2}} \\
&\leq \left( 1 - \alpha\lambda_{k+1} - \alpha \|x_k^{(N_2)}\|^2 \right) \sqrt{\sum_{j \neq k} (x_{kj}^{(N_2)})^2} + \alpha\sqrt{n}R_k\varepsilon^2 + 2\sqrt{\alpha n}R_k\varepsilon \\
&\leq \left( 1 - \alpha\lambda_{k+1} - \alpha \|x_k^{(N_2)}\|^2 \right) \sqrt{\sum_{j \neq k} (x_{kj}^{(N_2)})^2} + 3\sqrt{\alpha n}R_k\varepsilon.
\end{aligned} \tag{98}$$

The first inequality adopts the fact that, without  $k$ -th entry,  $\lambda_{k+1}$  is the smallest eigenvalue of  $\tilde{A}$ ; the second inequality mainly uses the inequality of square-root function; and the last inequality holds for sufficiently small  $\varepsilon$ .

Substituting (97) and (98) into (96), we obtain,

$$\begin{aligned}
\tan \theta_k^{(N_2+1)} &\leq \frac{1 - \alpha\lambda_{k+1} - \alpha \|x_k^{(N_2)}\|^2}{1 - \alpha \frac{\lambda_k + \lambda_{k+1}}{2} - \alpha \|x_k^{(N_2)}\|^2} \tan \theta_k^{(N_2)} + \frac{3\sqrt{\alpha n}R_k^2\varepsilon}{\frac{1}{2} \|x_k^{(N_2)}\| \delta} \\
&\leq \frac{1 - \alpha\lambda_{k+1}}{1 - \alpha \frac{\lambda_k + \lambda_{k+1}}{2}} \tan \theta_k^{(N_2)} + \frac{24\sqrt{\alpha n}R_k^2\varepsilon}{\sqrt{-2\lambda_q}\delta} \\
&\leq (1 - \beta) \tan \theta_k^{(N_2)} - \beta \tan \theta_k^{(N_2)} + C\varepsilon,
\end{aligned} \tag{99}$$

where  $\beta = \frac{1}{2} \left( 1 - \frac{1 - \alpha\lambda_{k+1}}{1 - \alpha \frac{\lambda_k + \lambda_{k+1}}{2}} \right) = \frac{\alpha(\lambda_{k+1} - \lambda_k)}{4 - 2\alpha(\lambda_k + \lambda_{k+1})} \in (0, 1)$  and  $C = \frac{24\sqrt{\alpha n}R_k^2}{\sqrt{-2\lambda_q}\delta}$ .

Based on the last inequality in (99), if  $\tan \theta_k^{(N_2)} > \frac{C\varepsilon}{\beta}$ , then we have  $\tan \theta_k^{(N_2+1)} < (1 - \beta) \tan \theta_k^{(N_2)}$ , which implies  $\cos \theta_k^{(N_2+1)} > \cos \theta_k^{(N_2)}$  due to the fact that all angles acute angle. Therefore, (95) holds for  $t = N_2 + 1$  and  $\tan \theta_k^{(t)}$  decay monotonically until  $\tan \theta_k^{(t)} \leq \frac{C\varepsilon}{\beta}$ . When  $\tan \theta_k^{(t)} \leq \frac{C\varepsilon}{\beta}$ , we obviously have  $\tan \theta_k^{(t+1)} \leq \frac{C\varepsilon}{\beta}$ . The inequality condition (99) still holds as long as  $\varepsilon$  is sufficiently small. Hence there exists a  $N$  such that for all  $t > N$ , we have  $\tan \theta_k^{(t+1)} \leq \frac{C\varepsilon}{\beta}$ , which can be arbitrarily small.

□

Lemma B.6 is fairly similar to Lemma B.5 with only subtle difference in the proof. Hence we provide the proof without detail derivation for inequalities.

**Lemma B.6.** *Assume Assumption A is satisfied and  $\lim_{t \rightarrow \infty} \|X_{k-1}^{(t)} - \mathcal{X}_{k-1}^*\|_{\text{F}} = 0$ . If  $u_k^\top x_k^{(t)}$  converges to zero and there exists an integer  $j \in (k, q]$  such that  $u_j^\top x_k^{(t)}$  does not converge to zero, then there exists an integer  $i \in (k, q]$  such that the tangent of  $\theta_i^{(t)} = \angle(x_k^{(t)}, u_i)$  converges to 0.*

*Proof.* Without loss of generality, we assume  $A$  is diagonal. Hence we have  $u_i^\top x_k^{(t)} = e_i^\top x_k^{(t)} = x_{ki}^{(t)}$ . The notation of  $E^{(t)}$  and the same iteration (88) as in the proof of Lemma B.3 are used here.

Based on the assumptions, we denote  $i \in (k, q]$  as the smallest integer such that  $e_i^\top x_k^{(t)} = x_{ki}^{(t)}$  does not converge to zero. Hence, there exists a positive number  $\delta > 0$  such that for any  $N$ , there exists a  $t > N$  and  $|x_{ki}^{(t)}| > \delta$ . We also know that Lemma B.4 holds,  $\|E^{(t)}\|$  converges to zero, and  $e_j^\top x_k^{(t)} = x_{kj}^{(t)}$  converges to zero for all  $j \in [k, i)$ . Hence, for any  $\varepsilon$  sufficiently small, there exists an integer  $N_1$  such that  $\|E^{(t)}\| \leq \varepsilon^2$ ,  $\|x_k^{(t)}\| \geq \frac{\sqrt{-2\lambda_q}}{4}$ , and  $|x_{kj}^{(t)}| \leq \varepsilon$  hold for all  $t > N_1$  and  $j \in [k, i)$ . Since  $e_j^\top x_k^{(t)} = x_{kj}^{(t)}$  does not converge to zero, there exists an integer  $N_2 > N_1$ , such that,

$$\frac{|x_{kj}^{(N_2)}|}{\|x_k^{(N_2)}\|} \geq \frac{\delta}{R_k} \geq \frac{2\varepsilon^2}{\lambda_{j+1} - \lambda_j}. \quad (100)$$

Using the iterative relationship (88), we have the lower bound on  $|x_{kj}^{(N_2+1)}|$ ,

$$|x_{kk}^{(N_2+1)}| \geq \left(1 - \alpha \frac{\lambda_j + \lambda_{j+1}}{2} - \alpha \|x_k^{(N_2)}\|^2\right) |x_{kj}^{(N_2)}|. \quad (101)$$

Again using the iterative relationship (88), we provide the upper bound for  $\sqrt{\sum_{j \neq i} (x_{kj}^{(N_2+1)})^2}$ , and the derivation is slightly different from that in (98),

$$\begin{aligned} \sqrt{\sum_{j \neq i} (x_{kj}^{(N_2+1)})^2} &\leq \left\{ \sum_{j < k} \left(1 - \alpha \|x_k^{(N_2)}\|^2\right)^2 (x_{kj}^{(N_2)})^2 + \sum_{k \leq j < i} \left(1 - \alpha \lambda_j - \alpha \|x_k^{(N_2)}\|^2\right)^2 \varepsilon^2 \right. \\ &\quad \left. + \sum_{j > i} \left(1 - \alpha \lambda_{i+1} - \alpha \|x_k^{(N_2)}\|^2\right)^2 (x_{kj}^{(N_2)})^2 \right\}^{\frac{1}{2}} + 3\sqrt{\alpha n} R_k \varepsilon. \quad (102) \\ &\leq \left(1 - \alpha \lambda_{i+1} - \alpha \|x_k^{(N_2)}\|^2\right) \sqrt{\sum_{j \neq i} (x_{kj}^{(N_2)})^2} + \sqrt{n} \varepsilon + 3\sqrt{\alpha n} R_k \varepsilon. \end{aligned}$$

The first inequality adopts similar derivation in (98) while keeps the first term unchanged; and the second inequality mainly uses the inequality of square-root function.

Substituting (101) and (102) into the expression of  $\tan \theta_i^{(N_2+1)}$ , we obtain,

$$\tan \theta_i^{(N_2+1)} \leq (1 - \beta) \tan \theta_i^{(N_2)} - \beta \tan \theta_i^{(N_2)} + C\varepsilon, \quad (103)$$

where  $\beta = \frac{\alpha(\lambda_{i+1} - \lambda_i)}{4 - 2\alpha(\lambda_i + \lambda_{i+1})} \in (0, 1)$  and  $C = \frac{8\sqrt{n}R_k + 24\sqrt{\alpha n}R_k^2}{\sqrt{-2\lambda_q}\delta}$ .

Based on the last inequality in (103), if  $\tan \theta_i^{(N_2)} > \frac{C\varepsilon}{\beta}$ , then we have  $\tan \theta_i^{(N_2+1)} < (1 - \beta) \tan \theta_i^{(N_2)}$ , which implies  $\cos \theta_i^{(N_2+1)} > \cos \theta_i^{(N_2)}$  due to the fact that all angles acute angle. Therefore, (100) holds for  $t = N_2 + 1$  and  $\tan \theta_i^{(t)}$  decay monotonically until  $\tan \theta_i^{(t)} \leq \frac{C\varepsilon}{\beta}$ . When  $\tan \theta_i^{(t)} \leq \frac{C\varepsilon}{\beta}$ , we obviously have  $\tan \theta_i^{(t+1)} \leq \frac{C\varepsilon}{\beta}$ . The inequality condition (99) still holds as long as  $\varepsilon$  is sufficiently small. Hence there exists a  $N$  such that for all  $t > N$ , we have  $\tan \theta_i^{(t+1)} \leq \frac{C\varepsilon}{\beta}$ , which can be arbitrarily small.  $\square$

**Proof of Lemma 4.3.** Without loss of generality, we assume that  $A$  is a diagonal matrix. All notations follow that in the proofs of previous lemmas.

Lemma B.3 implies that under the given assumptions, we have  $\lim_{t \rightarrow \infty} u_i^\top x_k^{(t)} = 0$  for all integer  $i \in [1, k] \cup (q, n]$ .

If  $u_i^\top x_k^{(t)}$  converges to zero for all  $i \in [k, q]$ , then  $x_k^{(t)}$  converges to zero vector, which is included in the statement of Lemma 4.3.

Otherwise, there exists an integer  $j \in [k, q]$  such that  $u_j^\top x_k^{(t)}$  does not converge to zero. Hence the condition in Lemma B.4 is satisfied. At the same time, either of Lemma B.5 or Lemma B.6 holds, which means that there exists an integer  $i \in [k, q]$  such that the tangent of  $\theta_i^{(t)}$  converges to zero. Now we focus on the convergence of  $x_{ki}^{(t)}$  and the proof follows steps in that of Lemma 4.2.

To simplify the notation, we denote  $\eta$  as  $\eta = \sqrt{x_{k1}^2 + \cdots + x_{k(i-1)}^2 + x_{k(i+1)}^2 + \cdots + x_{kn}^2}$ . Notice that  $\sin \theta_i^{(t)} = \frac{\eta^{(t)}}{\|x_k^{(t)}\|}$  converges to zero since the convergence of tangent and boundedness of cosine function. Lemma B.4 also shows the  $\|x_k^{(t)}\|$  is lower bounded by a constant for  $t$  large. Hence we conclude that  $\eta^{(t)}$  converges to zero.

According to the convergence of  $\eta^{(t)}$  and Lemma B.4, for any  $\varepsilon \leq \min\left(\frac{\sqrt{-\lambda_q}}{4}, \frac{\sqrt{\lambda_i \lambda_q}}{8R_k}\right)$ , there exists an integer  $M$  such that for any  $t \geq M$ , we have  $\eta^{(t)} \leq \varepsilon$ ,  $\|x_k^{(t)}\| \geq \frac{\sqrt{-2\lambda_q}}{4}$ , and  $\|E^{(t)}\| \leq \varepsilon^2$ . Combining these bounds, we obtain,  $(x_{ki}^{(t)})^2 = \|x_k^{(t)}\|^2 - \eta^2 \geq -\frac{\lambda_q}{8} - \varepsilon^2 \geq -\frac{\lambda_q}{16}$ .

Since the stepsize  $\alpha$  is small, the signs of  $x_{ki}^{(t)}$  and  $x_{ki}^{(0)}$  remain the same. We first discuss the scenario  $x_{ki}^{(0)} > 0$ .

Let  $\delta^{(t)} = x_{ki}^{(t)} - \sqrt{-\lambda_i}$ . We have the recurrent relationship for  $t > M$ ,

$$\begin{aligned} \delta^{(t+1)} &= x_{ki}^{(t+1)} - \sqrt{-\lambda_i} \\ &= \left(1 - \alpha \left(\lambda_i + (x_{ki}^{(t)})^2 + (\eta^{(t)})^2\right)\right) x_{ki}^{(t)} - \alpha \left(E_i^{(t)}\right)^\top x_k^{(t)} - \sqrt{-\lambda_i} \\ &= \left(1 - \alpha \left(\sqrt{-\lambda_i} + x_{ki}^{(t)}\right) x_{ki}^{(t)}\right) \delta^{(t)} - \alpha \left(\eta^{(t)}\right)^2 x_{ki}^{(t)} - \alpha \left(E_i^{(t)}\right)^\top x_k^{(t)}. \end{aligned} \quad (104)$$

Taking the absolute value of both side, we obtain the recurrent inequality relationship,

$$\begin{aligned}
|\delta^{(t+1)}| &\leq \left(1 - \alpha \left(\sqrt{-\lambda_i} + x_{ki}^{(t)}\right) x_{ki}^{(t)}\right) |\delta^{(t)}| + \alpha \left(\eta^{(t)}\right)^2 x_{ki}^{(t)} + \alpha \|E^{(t)}\| \|x_k^{(t)}\| \\
&\leq \left(1 - \alpha \frac{\sqrt{\lambda_i \lambda_q}}{4}\right) |\delta^{(t)}| + 2\alpha \varepsilon^2 R_k \\
&\leq \dots \\
&\leq \left(1 - \alpha \frac{\sqrt{\lambda_i \lambda_q}}{4}\right)^{t+1-M} |\delta^{(M)}| + \varepsilon^2 \frac{8R_k}{\sqrt{\lambda_i \lambda_q}} \\
&\leq \left(1 - \alpha \frac{\sqrt{\lambda_i \lambda_q}}{4}\right)^{t+1-M} |\delta^{(M)}| + \frac{\varepsilon}{2}.
\end{aligned} \tag{105}$$

Hence there exists an integer  $N \geq M$  such that for any  $t \geq N$ ,  $|\delta^{(t)}| \leq \varepsilon$ .

If  $x_{ki}^{(0)} < 0$ , the iteration converges to  $-\sqrt{-\lambda_i}$ . The analysis is analogy to the above one. The lemma is proved. □