SIAM J. Sci. Comput.
Vol. 42, No. 3, pp. A1686–A1713

LOW-RANK MATRIX ITERATION USING POLYNOMIAL-FILTERED SUBSPACE EXTRACTION[*]

YONGFENG LI[†], HAOYANG LIU[†], ZAIWEN WEN[†], AND YA-XIANG YUAN[‡]

Abstract. In this paper, we study fixed-point schemes with certain low-rank structures arising from matrix optimization problems. Traditional first-order methods depend on the eigenvalue decomposition at each iteration, which may take most of the computational time. In order to reduce the cost, we propose an inexact algorithmic framework based on a polynomial subspace extraction. The idea is to use an additional polynomial-filtered iteration to extract an approximated eigenspace and to project the iteration matrix on this subspace, followed by an optimization update. The accuracy of the extracted subspace can be controlled by the degree of the polynomial filters. This kind of subspace extraction also enjoys the warm-start property: the subspace of the current iteration is refined from the previous one. Then this framework is instantiated into two algorithms: the polynomial-filtered proximal gradient method and the polynomial-filtered alternating direction method of multipliers. The convergence of the proposed framework is guaranteed if the polynomial degree grows with an order $\mathcal{O}(\log k)$ at the kth iteration. If the warm-start property is considered, the degree can be reduced to a constant, independent of the iteration k. Preliminary numerical experiments on several matrix optimization problems show that the polynomial-filtered algorithms usually provide multifold speedups.

Key words. low-rank matrix iteration, eigenvalue decomposition, inexact optimization method, polynomial filter, subspace extraction

AMS subject classifications. 65F15, 90C06, 90C22, 90C25

DOI. 10.1137/19M1259444

1. Introduction.

Consider the fixed-point iteration scheme

$$x^{k+1} = Tx^k \overset{\triangle}{=} \mathcal{T}(x^k, \Psi(\mathcal{B}(x^k))), \quad x^k \in \mathcal{D}, \tag{1.1}$$

where $\mathcal{B} : \mathcal{D} \to \mathcal{S}^n$ is a bounded mapping from a given Euclidean space $\mathcal{D}$ to the n-dimensional symmetric matrix space $\mathcal{S}^n$, and $\mathcal{T} := \mathcal{T}(x, W)$ is a general mapping from $\mathcal{D} \times \mathcal{S}^n$ to $\mathcal{D}$. The operator $\Psi : \mathcal{S}^n \to \mathcal{S}^n$ is a *spectral operator*, whose detailed definition is explained in section 2.1. Generally speaking, for any symmetric matrix X, the spectral operator Ψ generates a new symmetric matrix using the eigenvalues and eigenvectors of X. In many applications, a certain low-rank structure of $\Psi(\mathcal{B}(x^k))$ is expected in (1.1), that is, evaluating $\Psi(\mathcal{B}(x^k))$ only requires a small number of eigenvalues and their corresponding eigenvectors of $\mathcal{B}(x^k)$. Therefore, scheme (1.1) can be formally interpreted as "the next iteration point is defined by the current point and a partial eigenvalue decomposition (EVD) of some matrix," since the spectral operator is determined by the EVD.

A variety of matrix optimization problems have low-rank structures as long as they can be solved by (1.1) involving an intermediate low-rank variable. For example,

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section May 1, 2019; accepted for publication (in revised form) March 24, 2020; published electronically June 8, 2020.
https://doi.org/10.1137/19M1259444
Funding: The work of the third author was supported by the NSFC through grants 11831002 and 11421101, and by the BAAI. The work of the fourth author was supported by the NSFC through grants 11331012 and 11688101.

[†] Beijing International Center for Mathematical Research, Peking University, Beijing, 100871 China (YongfengLi@pku.edu.cn, liuhaoyang@pku.edu.cn, wenzw@pku.edu.cn).
[‡] State Key Laboratory of Scientific and Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, 100190 China (yyx@lsec.cc.ac.cn).

in semidefinite programming (SDP), first-order methods such as the operator splitting method and the alternating direction method of multipliers (ADMM) need to compute all positive eigenvalues and their corresponding eigenvectors of a matrix at each iteration in order to preserve the semidefinite structure [18, 26]. In the robust principle component analysis (PCA), algorithms such as the accelerated proximal gradient (APG) approach, the augmented Lagrangian method (ALM), and the iterative thresholding (IT) method [27, 16] can also be regarded as an optimization problem on the rank-r matrix space, so manifold optimization methods can be applied [25]. All methods mentioned above can be transformed into scheme (1.1). In maximal eigenvalue problems, the objective function is the largest eigenvalue of a symmetric matrix variable. It is used in many real applications, such as the dual formulation of the max-cut problem [9], phase retrieval, blind deconvolution [8], the distance metric learning problem [29], and Crawford number computing [12]. Methods such as the (sub)gradient method can be regarded as a special case of (1.1).

In general, truncated EVDs in (1.1) are very time consuming when the dimension of the matrix $\mathcal{B}(x^k)$ and the number of the required eigenvalues are both large. First-order methods suffer from this issue greatly since they usually take thousands of iterations to converge. Therefore, many inexact methods are proposed to save computation time. One popular approach relies on approximated eigensolvers to solve the eigenvalue subproblem, such as the Lanczos method, locally optimal block preconditioned conjugate gradient (LOBPCG), and randomized EVD with early stopping rules; see [30, 2, 24] and the references therein. The performance of these methods is determined by the accuracy of the eigensolver, which is sometimes hard to control in practice. Another type of method is the so-called subspace method. By introducing a low-dimensional subspace, one can significantly reduce the dimension of the original problem, and then perform refinement on this subspace as the iterations proceed. This is widely used to solve, for instance, univariate maximal eigenvalue optimization problems [12, 23, 11]. In [31], the authors propose an inexact method which simplifies the eigenvalue computation using the Chebyshev filters in the self-consistent field (SCF) iterations. Although inexact methods are widely used, the convergence analysis is still very limited. Moreover, designing a practical strategy to control the inexactness of the algorithm is a big challenge.

1.1. Our contribution.

Our contributions are briefly summarized as follows:

1. For low-rank matrix optimization problems involving eigenvalue computations, we propose a general inexact first-order framework with polynomial filters which can be applied to most existing solvers without much difficulty. It can be observed that for low-rank problems, the iterates always lie in a low-dimensional eigenspace. Hence our key motivation is to use one polynomial filter subspace extraction step to estimate the subspace, followed by a standard optimization update. The algorithm also benefits from the warm-start property, i.e., computing $\Psi(\mathcal{B}(x^{k+1}))$ by using the EVD of $\mathcal{B}(x^k)$. Then we apply this framework to the proximal gradient (PG) method and the ADMM to obtain a polynomial-filtered PG (PFPG) method and a polynomial-filtered ADMM (PFAM) method, respectively.
2. We analyze the convergence property of the proposed framework. It can be shown that the difference between one exact and one inexact iteration is bounded by the principle angle between the true and the extracted subspace, which is further controlled by the polynomial degree. The convergence relies on an assumption that the initial space should not be orthogonal to the

target space, which is essential but usually holds in many applications. Consequently, the polynomial degree barely increases during the iterations. We also establish stronger convergence results for the PFPG method. For example, the polynomial degree can even remain constant under the warm-start setting. This result provides us the opportunity to use low-degree polynomials throughout the iterations in practice.

3. We provide a portable implementation of the polynomial filtered framework. It can be plugged into many first-order methods with only a few lines of code added, which means that the computational cost reduction is almost a free lunch.

We mention that our work is different from randomized eigensolver-based methods [30, 2, 24]. The inexactness of our method is mainly dependent on the quality of the subspace, which can be highly controllable using polynomial filters and the warm-start strategy. On the other hand, the convergence of randomized eigensolver-based methods relies on the tail bound or the per-vector error bound of the solution returned by the eigensolver, which is usually stronger than subspace assumptions. Our work also differs from the subspace method proposed in [12, 11]. First, the authors mainly focus on minimizing the jth largest eigenvalue or the sum of the squares of the j largest eigenvalues for a sufficiently small j, which has special structures, while we consider general matrix optimization problems with low-rank variables. Second, in the subspace method, one has to compute the exact solution of the subproblem, which is induced by the orthogonal projection of the original one onto the subspace. However, after we obtain the subspace, the next step is to extract eigenvalues and eigenvectors from the projected matrix to generate inexact variable updates. There are no subproblems in general.

1.2. Source codes. The MATLAB codes for several low-rank matrix optimization examples are available at https://github.com/RyanBernX/PFOpt.

1.3. Organization. The paper is organized as follows: In section 2, we introduce the definition of the spectral operator, propose the general framework for the polynomial-filtered algorithm, and show the convergence result. Then we instantiate the framework with two polynomial-filtered optimization methods, namely, PFPG and PFAM in sections 3 and 4, respectively. More specific convergence results are also included in these two sections. The effectiveness of PFPG and PFAM is demonstrated in section 5 with a number of practical applications. Finally, we conclude the paper in section 6.

1.4. Notation. Let $\mathcal{S}^n$ be the collection of all n-by-n symmetric matrices. The set of n-dimensional positive semidefinite matrices is written as $\mathcal{S}^n_+$. For any $X \in \mathcal{S}^n$, we use $\lambda(X) \in \mathbb{R}^n$ to denote all eigenvalues of X, which are permuted in descending order, i.e., $\lambda(X) = (\lambda_1, \ldots, \lambda_n)^T$, where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. For any matrix $X \in \mathbb{R}^{n\times n}$, $\operatorname{diag}(X)$ denotes a vector in $\mathbb{R}^n$ consisting of all diagonal entries of X. For any vector $x \in \mathbb{R}^n$, $\operatorname{Diag}(x)$ denotes a diagonal matrix in $\mathbb{R}^{n\times n}$ whose ith diagonal entry is x_i. We use $\langle A, B\rangle$ to define the inner product in the matrix space $\mathbb{R}^{n\times n}$, i.e., $\langle A, B\rangle := \operatorname{Tr}(A^T B)$, where $\operatorname{Tr}(X) := \sum_{i=1}^n X_{ii}$ for $X \in \mathbb{R}^{n\times n}$. The Frobenius norm is defined as $\|A\|_F = \sqrt{\operatorname{Tr}(A^T A)}$. For any matrix $X \in \mathbb{R}^{n\times p}$, $\operatorname{Range}(X)$ denotes the subspace spanned by the columns of X. In our convergence analysis, we denote by c a constant whose specific definition is restricted in the corresponding context. We use $\log(\cdot)$ to denote the natural logarithm and $\|\cdot\|$ to denote the norm of a general Euclidean space, which can either be the ℓ_2-norm $\|\cdot\|_2$ in the vector space or the Frobenius norm $\|\cdot\|_F$ in the matrix space.

2. Polynomial-filtered matrix iteration.

2.1. Spectral operator and low-rank assumption.

Let $X = V\mathrm{Diag}(\lambda_1, \dots, \lambda_n)V^T$ be the EVD of X. In scheme (1.1), the spectral operator Ψ is given by

$$\Psi(X) = V\mathrm{Diag}(\psi(\lambda(X)))V^T, \tag{2.1}$$

where the operator $\psi : \mathbb{R}^n \to \mathbb{R}^n$ is a vector-valued *symmetric mapping*; i.e., for any permutation matrix P we have $\psi(P\lambda) = P\psi(\lambda)$. We also say Ψ *is induced by* ψ for simplicity. More details of spectral operators can be found in [13, 14, 5].

According to the definition, in order to compute a spectral operator Ψ, one should perform a full EVD first, followed by an evaluation of ψ. A full EVD is expensive for large n in practice. Fortunately, computing $\Psi(X)$ may only require the orthogonal projection of X onto a low-dimensional subspace in many applications. Let $X \in \mathcal{S}^n$ and $Q \in \mathbb{R}^{n\times p}$ be a matrix with orthonormal columns, i.e., $Q^TQ = I$. The *orthogonal projection* of X onto Range(Q) is defined as

$$\mathcal{P}_Q(X) := \operatorname*{argmin}_{Y\in\mathcal{S}^n,\ \mathrm{Range}(Y)=\mathrm{Range}(Q)} \|Y - X\|_F^2 = QQ^TXQQ^T. \tag{2.2}$$

The low-rank property of Ψ is described in Assumption 1.

Assumption 1. *The operator Ψ satisfies the low-rank property at X if there exists a matrix with orthonormal columns $V_{\mathcal{I}} \in \mathbb{R}^{n\times p}$ $(p \ll n)$ that span a p-dimensional eigenspace corresponding to $\lambda_i(X)$, $i \in \mathcal{I}$, such that*

$$\Psi(X) = \Phi(\mathcal{P}_{V_{\mathcal{I}}}(X)), \tag{2.3}$$

where Φ is a spectral operator induced by ϕ, and $\mathcal{I}$ is an index set depending on X. Note that Ψ and Φ can be different.

A direct interpretation of Assumption 1 is that evaluating Ψ is equivalent to feeding a low-rank matrix to another spectral operator Φ. Therefore, all information needed for Ψ can be provided by $\mathcal{P}_{V_{\mathcal{I}}}(X)$, whose EVD is easy to compute if p is small. Alternatively, under this assumption it makes no difference to use another matrix $\tilde{X}$ as the input of Ψ, as long as it shares the same orthogonal projection onto Range$(V_{\mathcal{I}})$ with X. Since Range$(V_{\mathcal{I}})$ is a p-dimensional eigenspace of X, the EVD of $\mathcal{P}_{V_{\mathcal{I}}}(X)$ can be written as

$$\mathcal{P}_{V_{\mathcal{I}}}(X) = \begin{bmatrix}\tilde{V} & V_\perp\end{bmatrix}\begin{bmatrix}\Lambda_p & O\\ O & O\end{bmatrix}\begin{bmatrix}\tilde{V}^T\\ V_\perp^T\end{bmatrix}, \tag{2.4}$$

where $(\tilde{V}, \Lambda_p)$ is the corresponding eigenpairs and we have Range$(V_{\mathcal{I}})$ = Range$(\tilde{V})$. Note that $(\tilde{V}, \Lambda_p)$ has a closed-form expression with $\tilde{V} = V_{\mathcal{I}}Y$, where $Y\Lambda_pY^T$ is the full EVD of $H := V_{\mathcal{I}}^TXV_{\mathcal{I}} \in \mathbb{R}^{p\times p}$. Moreover, the choice of $V_{\mathcal{I}}$ is not unique, since Assumption 1 also holds for all $V_{\mathcal{I}'}$ such that $\mathcal{I}' \supseteq \mathcal{I}$ under a different definition of Φ.

We point out an important fact that computing $V_\perp$, the null space of $\mathcal{P}_{V_{\mathcal{I}}}(X)$, can be avoided because Φ is a spectral operator. Suppose Φ is induced by $\phi = (\phi_1, \dots, \phi_n)^T$, where ϕ_i is the ith component of ϕ. We claim that if $\lambda_i = \lambda_j$, then $\phi_i(\lambda) = \phi_j(\lambda)$. This observation can be verified using the property of a *symmetric mapping*. Consequently, for all i such that $\lambda_i = 0$, we have $\phi_i(\lambda) = c$, where c is a constant. Based on this fact, we have

$$\begin{aligned}\Psi(X) = \Phi(\mathcal{P}_{V_{\mathcal{I}}}(X)) &= \begin{bmatrix}\tilde{V} & V_\perp\end{bmatrix}\mathrm{Diag}(\phi_1(\lambda), \dots, \phi_p(\lambda), c, \dots, c)\begin{bmatrix}\tilde{V}^T\\ V_\perp^T\end{bmatrix}\\ &= \tilde{V}\mathrm{Diag}(\phi_1(\lambda), \dots, \phi_p(\lambda))\tilde{V}^T + cV_\perp V_\perp^T\\ &= \tilde{V}\mathrm{Diag}(\phi_1(\lambda), \dots, \phi_p(\lambda))\tilde{V}^T + c(I - \tilde{V}\tilde{V}^T).\end{aligned}$$

We give a concrete example to further describe Assumption 1.

Example 2.1. Let $\Psi(X) := \mathcal{P}_+(X)$ be the orthogonal projection onto $\mathcal{S}^n_+$, which is a spectral operator induced by ψ with the form

$$(\psi(\lambda))_i = \max\{\lambda_i, 0\} \quad \forall i.$$

It can be verified that $\Psi(X) = \Psi(\mathcal{P}_{V_\mathcal{I}}(X))$, where $\mathcal{I}$ contains all i such that $\lambda_i(X) > 0$. We say that $\Psi(X)$ satisfies the low-rank assumption *at* X if X only has a few positive eigenvalues. Note that $\mathcal{I}$ may also contain a few indices of negative eigenvalues without changing the definition of $\Phi = \Psi$.

2.2. Chebyshev-filtered subspace iteration. The idea of polynomial filtering is originated from a well-known fact that polynomials are able to manipulate the eigenvalues of any symmetric matrix A while keeping its eigenvectors unchanged. Suppose A has the eigenvalue decomposition $A = U\text{Diag}(\lambda_1, \dots, \lambda_n)U^T$; then the matrix $\rho(A)$ has the eigenvalue decomposition $\rho(A) = U\text{Diag}(\rho(\lambda_1), \dots, \rho(\lambda_n))U^T$ for any polynomial $\rho(\cdot)$.

Consider the traditional subspace iteration $U \leftarrow AU$. The convergence of the desired eigenspace is determined by the gap of the eigenvalues, which can be very slow if the gap is nearly zero. In order to obtain a better convergence, polynomial filters are widely used to manipulate the gap. For any polynomial $\rho(t)$ and initial matrix U, the polynomial-filtered subspace iteration is given by $U \leftarrow \rho(A)U$. In general, there are many choices of $\rho(t)$. One popular choice is to use Chebyshev polynomials of the first kind, whose explicit expression can be written as

$$T_d(t) = \begin{cases} \cos(d \arccos t), & |t| \le 1, \\ \frac{1}{2}((t - \sqrt{t^2-1})^d + (t + \sqrt{t^2-1})^d), & |t| > 1, \end{cases} \tag{2.5}$$

where d is the degree of the polynomial. One important property of Chebyshev polynomials is that they grow pretty fast outside the interval $[-1, 1]$, which can be helpful to suppress all unwanted eigenvalues in this interval effectively. Figure 1 shows some examples of Chebyshev polynomials $T_d(t)$ with $d = 1, \dots, 6$ and power of Chebyshev polynomials $T_3^q(t)$ with $q = 1, \dots, 5$. That is exactly why Chebyshev polynomials are taken into our consideration.

In order to suppress all eigenvalues in a general interval $[a, b]$, a linear mapping from $[a, b]$ to $[-1, 1]$ is constructed. To be precise, the polynomial can be chosen as

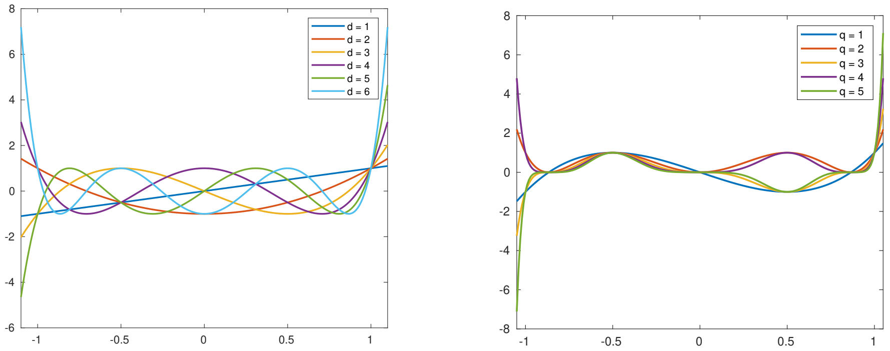
$$\rho(t) = T_d\left(\frac{t - (b+a)/2}{(b-a)/2}\right). \tag{2.6}$$

After the polynomial filters are applied, an orthogonalization step usually follows in order to prevent the subspace from losing rank. This step is often performed by a single QR decomposition. Consequently, given an arbitrary matrix $U \in \mathbb{R}^{n \times p}$ and a polynomial $\rho(t)$, the polynomial-filtered subspace iteration can be simply written as

$$U^+ = \text{orth}(\rho(A)U), \tag{2.7}$$

where "orth" stands for the orthogonalization operation. In most cases, the updated matrix U^+ spans a subspace which contains some approximated eigenvectors of A.

The concept of Chebyshev-filtered subspace iteration is very common in eigenvalue approximations; see [7, 21, 31, 12]. However, combining it with optimization algorithms seems novel and very appealing.



(a) Chebyshev polynomials of the first kind $T_d(t)$ (b) Power of Chebyshev polynomial $T_3^q(t)$

Fig. 1. *Chebyshev polynomials and their variants.*

2.3. A general framework. In this subsection, we present a general algorithm framework for scheme (1.1) under Assumption 1 and with the help of Chebyshev polynomials. Let x^* be the solution of the fixed-point problem. We now assume that Assumption 1 holds for every $\mathcal{B}(x^k)$ in (1.1). Consequently, scheme (1.1) is equivalent to

$$x^{k+1} = \mathcal{T}(x^k, \Phi(\mathcal{P}_{V^k_{\mathcal{I}_k}}(\mathcal{B}(x^k)))), \tag{2.8}$$

where $V^k_{\mathcal{I}_k}$ is determined by $\mathcal{B}(x^k)$. According to the argument in section 2.1, if $V^k_{\mathcal{I}_k}$ is known somehow, the computational cost of Ψ can be greatly reduced. In practice, the subspace $V^k_{\mathcal{I}_k}$ is never known, yet it can be approximated by a polynomial-filtered subspace U^k. Combining (2.7) and (2.8) yields

$$x^{k+1} = \mathcal{T}(x^k, \Phi(\mathcal{P}_{U^k}(\mathcal{B}(x^k)))), \tag{2.9}$$

$$U^{k+1} = \mathrm{orth}(\rho_{k+1}^{q_{k+1}}(\mathcal{B}(x^{k+1}))U^k). \tag{2.10}$$

Compared to (2.8), the exact subspace $V^k_{\mathcal{I}_k}$ is substituted by its approximation U^k in (2.9). After x^{k+1} is obtained, a polynomial filter step (2.10) follows in order to extract a new subspace U^{k+1} based on U^k, aiming to find a good approximation of $V^{k+1}_{\mathcal{I}_{k+1}}$. Here q_k is the number of times the filter $\rho_k(\cdot)$ is applied to U^k, which is usually a small integer (e.g., 1 to 3). The initial U^0 can be set to the exact subspace $V^0_{\mathcal{I}_0}$. Note that the warm-start property follows directly from iteration (2.10) since U^{k+1} is refined from U^k.

We remark that (2.9) and (2.10) are actually two different updates that couple together. Given a subspace spanned by U^k, step (2.9) performs the fixed-point iteration with the approximated Ψ in terms of an orthogonal projection into the subspace spanned by the columns of U^k. The next step, (2.10), makes refinement on the subspace to obtain U^{k+1} via polynomial filters. The subspace can become more and more accurate as the iterations proceed. In practice, it is usually observed that U^k is very "close" to the true eigenspace $V^k_{\mathcal{I}_k}$ during the last few iterations of the algorithm framework.

The choice of $\rho_k(\cdot)$ depends on the eigenvalue index set $\mathcal{I}_k$. Chebyshev polynomials are suitable if most λ_i, $i \notin \mathcal{I}_k$, are located within an interval. For instance, λ_i,

$i \in \mathcal{I}_k$, can be the l largest/smallest eigenvalues in magnitude, or all positive/negative eigenvalues. If $\mathcal{I}_k$ does not satisfy this property, then other polynomial filters are preferred, which is not covered in this paper.

2.4. Convergence analysis.

2.4.1. Preliminary. In this section we introduce some basic notation and tools that are used in our analysis. First, we introduce the definition of principal angles to measure the distance between two subspaces.

Definition 1. *Let $X, Y \in \mathbb{R}^{n\times p}$ be two matrices with orthonormal columns. The singular values of X^TY are $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p$, which lie in $[0,1]$. The principal angle between $\mathrm{Range}(X)$ and $\mathrm{Range}(Y)$ is defined by*

$$\Theta(X,Y) := \mathrm{Diag}(\arccos\sigma_1, \dots, \arccos\sigma_p). \tag{2.11}$$

In particular, we define $\sin\Theta$ by taking the sine of Θ componentwisely.

The following lemma describes a very important property of Chebyshev polynomials.

Lemma 2. *The Chebyshev polynomials increase quickly outside $[-1,1]$, i.e.,*

$$|T_d(\pm(1+\epsilon))| \geq 2^{d\min\{\sqrt{\epsilon},1\}-1} \quad \forall \epsilon > 0. \tag{2.12}$$

Proof. It follows from the expression (2.5) that

$$\begin{aligned}|T_d(\pm(1+\epsilon))| &\geq \frac{1}{2}(1+\epsilon+\sqrt{2\epsilon+\epsilon^2})^d \geq \frac{1}{2}(1+\sqrt{\epsilon})^d\\ &= \frac{1}{2}e^{d\log(1+\sqrt{\epsilon})} \geq 2^{d\min\{\sqrt{\epsilon},1\}-1},\end{aligned}$$

where the last inequality follows from that $\log(1+x) \geq \log 2 \cdot \min\{x,1\}$. □

Given a Euclidean space $\mathcal{D}$, an operator $F : \mathcal{D} \to \mathcal{D}$ is called nonexpansive if it satisfies

$$\|F(x) - F(y)\| \leq \|x-y\| \quad \forall x, y \in \mathcal{D}.$$

A mapping F is called α-averaged ($0 < \alpha < 1$) if there exists a nonexpansive mapping T such that $F = \alpha I + (1-\alpha)T$, where I is the identity mapping. The following lemma is frequently used in our analysis.

Lemma 3 ([1, Proposition 4.25]). *If $F : \mathcal{D} \to \mathcal{D}$ is α-averaged, then*

$$\|Fx - Fy\|^2 \leq \|x-y\|^2 - \frac{1-\alpha}{\alpha}\|(I-F)x + (I-F)y\|^2 \quad \forall\ x, y \in \mathcal{D},$$

where I is the identity mapping.

A differential function $f : \mathcal{D} \to \mathcal{D}$ is called μ-strongly convex if it satisfies

$$f(y) \geq f(x) + \langle \nabla f(x), y-x\rangle + \frac{\mu}{2}\|y-x\|^2.$$

2.4.2. Convergence analysis for general framework. We next analyze the convergence of the general framework (2.9)–(2.10). Generally speaking, if the initial space U^k is not orthogonal to $V^{k+1}_{\mathcal{I}_{k+1}}$ and both $\mathcal{T}$ and Ψ have some continuity property (see Assumption 2), then the general framework converges sublinearly in the sense of the residual norm $\|x^k - Tx^k\|$ when $d_k \approx c\log(k)$ (logarithm growth).

Let λ_i be an eigenvalue of $\mathcal{B}(x^k)$, and let $\mathcal{I}_k$ be the index set of the eigenvalues beyond $[a_k, b_k]$, i.e., the interval suppressed by Chebyshev polynomials at the kth iteration. The relative gap is defined by

$$\mathcal{G}_k = \min_{i \in \mathcal{I}_k} \left(\frac{|\lambda_i - (a_k + b_k)/2|}{(b_k - a_k)/2} - 1 \right). \tag{2.13}$$

Throughout our analysis, we make the following assumptions.

Assumption 2.

(i) $\|\sin\Theta(V^{k+1}_{\mathcal{I}_{k+1}}, U^k)\|_2 < \gamma \; \forall k$ *with* $\gamma < 1$, *where* U^k *is defined in* (2.10) *at iteration* k.

(ii) *The sequence generated by* (2.9)–(2.10) *is bounded, i.e.,* $\|x^k\| \le C \; \forall k$. *The mapping* $\mathcal{B}$ *is bounded, i.e.,* $\|\mathcal{B}(x)\|_F \le B\|x\| \; \forall x \in \mathcal{D}$.

(iii) *The relative gap* (2.13) *has a lower bound, i.e.,* $\mathcal{G}_k \ge l > 0 \; \forall k$.

(iv) *The spectral operator in* (2.8) *is Lipschitz continuous, i.e.,*

$$\|\Phi(X) - \Phi(Y)\|_F \le L_\Phi \|X - Y\|_F \quad \forall X, Y \in \mathcal{S}^n. \tag{2.14}$$

(v) *The operator* $\mathcal{T}(x, W)$ *in* (1.1) *satisfies that*

$$\|\mathcal{T}(x, W_1) - \mathcal{T}(x, W_2)\| \le L_{\mathcal{T}} \|W_1 - W_2\|_F \quad \forall W_1, W_2 \in \mathcal{S}^n. \tag{2.15}$$

Assumption 2(i) implies that the initial eigenspace is not orthogonal to the truly wanted eigenspace so that we can obtain $V^{k+1}_{\mathcal{I}_{k+1}}$ by performing subspace iterations on U^k. Essentially, this property is required by almost all iterative eigensolvers for finding the correct eigenspace at each iteration. Assumption 2(ii) is commonly used in the optimization literature. Assumption 2(iii) is a key assumption which guarantees that the relative gap $\mathcal{G}_k$ exists in the asymptotic sense. Therefore, the polynomial filters are able to separate the wanted eigenvalues from the unwanted ones. In eigenvalue computation, this property is often enforced by adding a sufficient number of guard vectors so that the corresponding eigenvalue gap is large enough. Assumption 2(iv) is necessary for a stable computation of the spectral operator. Assumption 2(v) is a continuity property which is usually satisfied in many methods, such as the proximal gradient method and the ADMM.

The following lemma gives an error estimate when using the approximated eigenspace extracted by Chebyshev polynomials to compute the spectral operator. It states that the error is exponentially decreasing with respect to the polynomial degree d_k.

Lemma 4. *Suppose that Assumption* 2 *holds. Define the error*

$$E^k = \Phi(\mathcal{P}_{U^k}(\mathcal{B}(x^k))) - \Phi(\mathcal{P}_{V^k_{\mathcal{I}_k}}(\mathcal{B}(x^k))). \tag{2.16}$$

Let the polynomial filter $\rho_k(t)$ *be a Chebyshev polynomial with degree* d_k. *Then we have*

$$\|E^k\|_F \le \frac{c_1 \cdot 2^{q_k}}{2^{q_k d_k \min\{l,1\}}}, \tag{2.17}$$

where c_1 *is a constant and* l *is the lower bound in Assumption* 2(iii).

Proof. Let $\rho_k^{q_k}(\lambda_i)$ denote the eigenvalue of $\rho_k^{q_k}(\mathcal{B}(x^k))$, where λ_i is the ith eigenvalue of $\mathcal{B}(x^k)$. According to Lemma 2, the eigenvalues have the following distribution after the Chebyshev polynomial is applied:

$$\min_{i \in \mathcal{I}_k} \rho_k^{q_k}(\lambda_i) \ge 2^{q_k d_k \min\{\sqrt{l},1\} - q_k} \text{ and } \max_{i \notin \mathcal{I}_k} \rho_k^{q_k}(\lambda_i) \le 1.$$

The update (2.10) can be regarded as one orthogonal iteration. Then it follows from [10, Theorem 8.2.2] that

$$\|\sin\Theta(V_{\mathcal{I}_k}^k, U^k)\|_2 \le \frac{2^{q_k}}{2^{q_k d_k \min\{l,1\}}} \frac{\gamma}{\sqrt{1-\gamma^2}}, \tag{2.18}$$

where γ is defined in Assumption 2(i). The following argument gives an upper bound on E^k:

$$\begin{aligned}
\|E^k\|_F &\le L_\Phi \|\mathcal{P}_{U^k}(\mathcal{B}(x^k)) - \mathcal{P}_{V_{\mathcal{I}_k}^k}(\mathcal{B}(x^k))\|_F \qquad \text{(2.19a)}\\
&\le L_\Phi \big(\|\mathcal{P}_{U^k}(\mathcal{B}(x^k)) - U^k (U^k)^T \mathcal{B}(x^k) V_{\mathcal{I}_k}^k (V_{\mathcal{I}_k}^k)^T\|_F \\
&\qquad + \|U^k (U^k)^T \mathcal{B}(x^k) V_{\mathcal{I}_k}^k (V_{\mathcal{I}_k}^k)^T - \mathcal{P}_{V_{\mathcal{I}_k}^k}(\mathcal{B}(x^k))\|_F \big) \\
&\le L_\Phi \big(\|U^k (U^k)^T\|_2 \|\mathcal{B}(x^k)\|_F \|U^k (U^k)^T - V_{\mathcal{I}_k}^k (V_{\mathcal{I}_k}^k)^T\|_2 \qquad \text{(2.19b)}\\
&\qquad + \|U^k (U^k)^T - V_{\mathcal{I}_k}^k (V_{\mathcal{I}_k}^k)^T\|_2 \|\mathcal{B}(x^k)\|_F \|V_{\mathcal{I}_k}^k (V_{\mathcal{I}_k}^k)^T\|_2 \big) \\
&\le 2 L_\Phi B C \|U^k (U^k)^T - V_{\mathcal{I}_k}^k (V_{\mathcal{I}_k}^k)^T\|_2, \qquad \text{(2.19c)}
\end{aligned}$$

where inequality (2.19a) holds due to the Lipschitz continuity of Φ, (2.19b) holds due the fact that $\|XY\|_F \le \|X\|_2 \|Y\|_F$, and (2.19c) follows from the column orthogonality between U^k and $V_{\mathcal{I}_k}^k$ and the boundedness of $\mathcal{B}$ and x^k. Moreover, it is shown in [10, Theorem 2.5.1] that

$$\|\sin\Theta(X, Y)\|_2 = \|XX^T - YY^T\|_2. \tag{2.20}$$

Combining (2.18), (2.19), and (2.20) yields

$$\|E^k\|_F \le 2 L_\Phi B C \|\sin\Theta(V_{\mathcal{I}_k}^k, U^k)\|_2 \le \frac{c_1 \cdot 2^{q_k}}{2^{q_k d_k \min\{l,1\}}}, \tag{2.21}$$

where c_1 is a constant depending on L_Φ, B, C, γ, which completes the proof. □

The next theorem is established by following the proof in [4, Lemma 3, Theorem 1].

THEOREM 5. *Suppose that Assumption* 2 *holds. If T defined by* (1.1) *is α-averaged, then the convergence of the residual $\|x^k - Tx^k\| = o(1/\sqrt{k})$ is achieved when*

$$d_k \ge \frac{c \log k}{\min\{l, 1\}}, \quad k > 1, \tag{2.22}$$

where c is a constant independent of k and l.

Proof. Let $e^k = x^{k+1} - Tx^k$ denote the error between one original update Tx^k and one polynomial-filtered iteration x^{k+1}. According to Lemma 4 and Assumption 2(v), we have

$$\|e^k\| \le L_{\mathcal{T}} \|\Phi(\mathcal{P}_{U^k}(\mathcal{B}(x^k))) - \Phi(\mathcal{P}_{V_{\mathcal{I}_k}^k}(\mathcal{B}(x^k)))\|_F \le \frac{c_1 L_{\mathcal{T}} \cdot 2^{q_k}}{2^{q_k d_k \min\{l,1\}}}.$$

Note that by choosing $d_k \ge \frac{c \log k}{\min(l,1)}$, the error norm $\|e^k\|$ can be bounded by

$$\|e^k\| \le 1/k^2. \tag{2.23}$$

We denote the residual vector by $w^k = Tx^k - x^k$ and prove the main result in three steps.

Step (a): $\|w^k\|$ *is bounded.* Define $s^k = x^{k+1} - x^k$ and $t^k = Tx^{k+1} - Tx^k$. Then we have

$$\begin{aligned} \|w^{k+1}\|^2 &= \|w^k\|^2 + \|w^{k+1} - w^k\|^2 + 2\langle w^k, w^{k+1} - w^k\rangle \\ &= \|w^k\|^2 + \|t^k - s^k\|^2 + 2\langle s^k - e^k, t^k - s^k\rangle. \end{aligned} \tag{2.24}$$

Since T is α-averaged, it follows from Lemma 3 that

$$\begin{aligned} 2\langle s^k, t^k - s^k\rangle &= \|t^k\|^2 - \|s^k\|^2 - \|t^k - s^k\|^2 \\ &\leq -\frac{1-\alpha}{\alpha}\|t^k - s^k\|^2 - \|t^k - s^k\|^2 = -\frac{1}{\alpha}\|t^k - s^k\|^2. \end{aligned} \tag{2.25}$$

Plugging (2.25) into (2.24) yields

$$\begin{aligned} &\|w^{k+1}\|^2 \leq \|w^k\|^2 + \left(1 - \frac{1}{\alpha}\right)\|t^k - s^k\|^2 - 2\langle e^k, t^k - s^k\rangle \\ &\quad = \|w^k\|^2 - \frac{1-\alpha}{\alpha}\|t^k - s^k + \frac{\alpha}{1-\alpha}e^k\|^2 + \frac{\alpha}{1-\alpha}\|e^k\|^2 \leq \|w^k\|^2 + \frac{\alpha}{1-\alpha}\|e^k\|^2. \end{aligned} \tag{2.26}$$

Applying (2.26) recursively gives

$$\|w^k\|^2 \leq \|w^0\|^2 + \frac{\alpha}{1-\alpha}\sum_{i=0}^{k-1}\|e^k\|^2 < \infty,$$

where the last inequality follows from the choice of d_k and (2.23).

Step (b): $\|w^k\| \to 0$. Let x^* be a fixed point of T. We have

$$\|Tx^k - x^*\| = \|Tx^k - Tx^*\| \leq \|x^k - x^*\| \leq \|Tx^{k-1} - x^*\| + \|e^{k-1}\|. \tag{2.27}$$

Applying (2.27) recursively gives

$$\|Tx^k - x^*\| \leq \|Tx^0 - x^*\| + \sum_{i=0}^{k-1}\|e^k\| < \infty,$$

where the last inequality follows from (2.23). Hence $\|Tx^k - x^*\|$ is bounded for any k. According to Lemma 3 and the Cauchy inequality, we have

$$\begin{aligned} \|x^{k+1} - x^*\|^2 &= \|Tx^k - x^*\|^2 + \|e^k\|^2 + 2\langle Tx^k - x^*, e^k\rangle \\ &\leq \|x^k - x^*\|^2 - \frac{1-\alpha}{\alpha}\|w^k\|^2 + \|e^k\|^2 + 2\|Tx^k - x^*\|\|e^k\|. \end{aligned}$$

By taking the sum from $k = 0$ to m, we obtain

$$\begin{aligned} \frac{1-\alpha}{\alpha}\sum_{i=0}^{m}\|w^i\|^2 &\leq \|x^0 - x^*\|^2 - \|x^{m+1} - x^*\|^2 + \sum_{i=0}^{m}\|e^i\|^2 + 2\sum_{i=0}^{m}\|Tx^i - x^*\|\|e^i\| \\ &\leq \|x^0 - x^*\|^2 + \sum_{i=0}^{m}\|e^i\|^2 + 2\sum_{i=0}^{m}\|Tx^i - x^*\|\|e^i\|. \end{aligned}$$

Therefore, it holds that

$$\frac{1-\alpha}{\alpha}\sum_{i=0}^{\infty}\|w^i\|^2 \le \|x^0-x^*\|^2+\sum_{i=0}^{\infty}\|e^i\|^2+2\sum_{i=0}^{\infty}\|Tx^i-x^*\|\|e^i\|<\infty, \tag{2.28}$$

where the last inequality follows from (2.23) and $\|Tx^i-x^*\|$ is bounded, which further implies $\|w^k\|\to 0$.

Step (c): $\|w^k\| = o(1/\sqrt{k})$. It suffices to show $\lceil k/2\rceil \|w^k\|^2 \to 0$. In fact we have

$$\lceil k/2\rceil\,\|w^k\|^2 = \sum_{i=\lceil k/2\rceil}^{k}\|w^k\|^2 \le \sum_{i=\lceil k/2\rceil}^{k}(\|w^i\|^2+\frac{\alpha}{1-\alpha}\sum_{j=i}^{k}\|e^j\|^2) \tag{2.29a}$$

$$= \sum_{i=\lceil k/2\rceil}^{k}\|w^i\|^2+\frac{\alpha}{1-\alpha}\sum_{i=\lceil k/2\rceil}^{k}(i+1-\lceil k/2\rceil)\|e^i\|^2$$

$$\le \sum_{i=\lceil k/2\rceil}^{k}\|w^i\|^2+\frac{\alpha}{1-\alpha}\sum_{i=\lceil k/2\rceil}^{k}1/i^3 \longrightarrow 0, \tag{2.29b}$$

where (2.29a) follows by recursively applying (2.26) from i to k, and (2.29b) is due to $(i+1-\lceil k/2\rceil)\|e^i\|^2 < i\cdot(1/i^4)=1/i^3$. The summability of $\|w^i\|^2$ given by (2.28) and $1/i^3$ implies that the right-hand side of (2.29b) converges to zero, which completes the proof. ∎

3. Application on the proximal gradient method.

3.1. The PFPG method. For given symmetric matrices $A_i \in \mathcal{S}^n$, we define a linear operator $\mathcal{A}$ and its adjoint operator of $\mathcal{A}^*$ as

$$\mathcal{A}(X) = [\langle A_1, X\rangle,\ldots,\langle A_m, X\rangle]^T, \quad \mathcal{A}^*(x)=\sum_{i=1}^{m}x_iA_i. \tag{3.1}$$

Consider a function $f:\mathbb{R}^n\to\mathbb{R}$ that is smooth and *absolutely symmetric*, i.e., $f(x)=f(Px)$ $\forall x\in\mathbb{R}^n$ and any permutation matrix $P\in\mathbb{R}^{n\times n}$. Let $F(x)$ be a composite function $F(x)=f\circ\lambda(\mathcal{B}(x))$ with $\mathcal{B}(x)=G+\mathcal{A}^*(x)$ for a given matrix $G\in\mathcal{S}^n$, and let $R(x)$ be a (probably nonsmooth) convex function with simple structures. It can be easily verified that $F(x)$ is well defined. We next show the application of the subspace update (2.9) and (2.10) to the proximal gradient method on the composite optimization problem

$$\min_x \quad h(x) := F(x)+R(x). \tag{3.2}$$

According to [19, section 6.7.2], the gradient of F in (3.2) is

$$\nabla F(x) = \mathcal{A}(\Psi(\mathcal{B}(x))), \tag{3.3}$$

and Ψ is the spectral operator induced by $\psi=\nabla f$. Then the proximal gradient method to solve (3.2) can be written as

$$x^{k+1} = \operatorname{prox}_{\tau_k R}(x^k - \tau_k\mathcal{A}(\Psi(\mathcal{B}(x^k)))), \tag{3.4}$$

where τ_k is the step size, and the proximal operator is defined by

$$\operatorname{prox}_{tR}(x) = \operatorname*{argmin}_u R(u)+\frac{1}{2t}\|u-x\|_2^2. \tag{3.5}$$

Scheme (3.4) is a special case of (1.1) with

$$\begin{aligned} \mathcal{T}(x, W) &= \operatorname{prox}_{\tau_k R}(x - \tau_k \mathcal{A}(W)), \\ \Psi(X) &= V \operatorname{Diag}(\nabla f(\lambda(X))) V^T, \\ \mathcal{B}(x) &= G + \mathcal{A}^*(x). \end{aligned} \tag{3.6}$$

Suppose that Assumption 1 holds for each iteration; then the PFPG method can be written as

$$x^{k+1} = \operatorname{prox}_{\tau_k R}(x^k - \tau_k \mathcal{A}(\Phi(\mathcal{P}_{U^k}(\mathcal{B}(x^k))))), \tag{3.7}$$

$$U^{k+1} = \operatorname{orth}(\rho_{k+1}^{q_{k+1}}(\mathcal{B}(x^{k+1}))U^k). \tag{3.8}$$

We note that problem (3.2) is considered to have low-rank structures in the sense that computing $\nabla F(x)$ involves a truncated EVD of the matrix $\mathcal{B}(x)$.

3.2. Convergence analysis.

3.2.1. General convergence results. We can easily establish the convergence of the PFPG method (3.7)–(3.8) by Theorem 5. If Assumptions 2(i)–(iv) hold, we claim that $\nabla F(x)$ is Lipschitz continuous due to the boundedness of $\mathcal{A}$ and x^k and the Lipschitz continuity of Φ. In the following part, we denote the Lipschitz constant of $\nabla F(x)$ by L_F.

COROLLARY 6. *Suppose that Assumptions* 2(i)–(iv) *hold and* $\{x^k\}$ *is generated by the PFPG method* (3.7)–(3.8)*. If* F *and* R *are convex and the step size* $\tau_k = \tau < \frac{2}{L_F}$*, then we have*

$$\|x^k - \operatorname{prox}_{\tau R}(x^k - \tau \nabla F(x^k))\|_2 = o\left(1/\sqrt{k}\right)$$

when d_k *satisfies* (2.22).

Proof. The proof follows from Theorem 5 by verifying the Assumption 2(v) for the operator $\mathcal{T}$ defined in (3.6) and the averageness of the operator $\operatorname{prox}_{\tau R}(I - \tau\nabla F)$. Due to [28, Proposition 2.3], the operator $\operatorname{prox}_{\tau R}(I - \tau\nabla F)$ is $\frac{2}{4-\tau L_F}$-averaged. By the nonexpansiveness of proximal operators [4, Proposition 1] and the boundedness of $\mathcal{A}$, we have

$$\begin{aligned} \|\mathcal{T}(x, W_1) - \mathcal{T}(x, W_2)\|_2 &= \|\operatorname{prox}_{\tau R}(x - \tau\mathcal{A}(W_1)) - \operatorname{prox}_{\tau R}(x - \tau\mathcal{A}(W_2))\|_2 \\ &\leq \|\tau\mathcal{A}(W_1) - \tau\mathcal{A}(W_2)\|_2 \leq c_3\|W_1 - W_2\|_F, \end{aligned} \tag{3.9}$$

where c_3 is a constant, which completes the proof. □

3.2.2. Specific results on convex problems. In this subsection, we give more specific results for PFPG on convex problems. If F and R are convex and Assumptions 2(i)–(iv) hold, the PFPG method converges in the terms of the function value as long as $d_k \approx c\log(k)$. Moreover, if F becomes strongly convex, the PFPG method converges linearly when $d_k \approx c \cdot k$. In the following analysis, the optimal solution of (3.2) is denoted by x^*.

THEOREM 7. *Suppose that Assumptions* 2 (i)–(iv) *hold and* F *and* R *are convex. Let* $\tau_k = \tau \leq \frac{1}{L_F}$ *and* $\bar{x}_K = \frac{1}{K}\sum_{k=1}^K x^k$*. Then* $\lim_{K\to\infty} h(\bar{x}^K) = h(x^*)$ *holds if* d_k *satisfies* (2.22).

Proof. Define the error between the exact and the inexact gradients as

$$e^k = \mathcal{A}(\Phi(\mathcal{P}_{U^k}(\mathcal{B}(x^k)))) - \nabla F(x^k). \tag{3.10}$$

According to Lemma 4 and the boundedness of $\mathcal{A}$, we obtain

$$\|e^k\|_2 \leq \frac{c_2 \cdot 2^{q_k}}{2^{q_k d_k \min\{l,1\}}}, \tag{3.11}$$

where c_2 is a constant. Hence, if d_k satisfies (2.22), then the error $\|e^k\|_2$ can be bounded as $\|e^k\|_2 \leq 1/k$.

For simplicity, we define the noisy residual function as

$$r_\tau(x, e) = \frac{1}{\tau}(x - \text{prox}_{\tau R}(x - \tau(\nabla F(x) + e))).$$

Then the update (3.7) can be written as

$$x^k - x^{k+1} = \tau_k r_{\tau_k}(x^k, e^k). \tag{3.12}$$

Due to the definition of $\text{prox}_{\tau R}(\cdot)$ and the optimality condition, we have

$$\frac{1}{\tau_k}(x^k - x^{k+1}) - \nabla F(x^k) - e^k \in \partial R(x^{k+1}). \tag{3.13}$$

Inserting (3.12) into (3.13) gives

$$r_{\tau_k}(x, e^k) - \nabla F(x^k) - e^k \in \partial R(x^{k+1}). \tag{3.14}$$

It follows from the Lipschitz continuity of ∇F and [1, Theorem 18.15] that

$$h(x^{k+1}) \leq F(x^k) - \tau_k \nabla F(x^k)^T (r_{\tau_k}(x^k, e^k)) + \frac{L_F \tau_k^2}{2}\|r_{\tau_k}(x^k, e^k)\|_2^2 + R(x^{k+1}). \tag{3.15}$$

By the convexity of $F(x)$, the convexity of $R(x)$, and (3.14), we have

$$\begin{aligned} h(x^{k+1}) \leq F(x^*) + \nabla F(x^k)^T(x^k - x^*) - \tau_k \nabla F(x^k)^T (r_{\tau_k}(x^k, e^k)) + \frac{L_F \tau_k^2}{2}\|r_{\tau_k}(x^k, e^k)\|_2^2 \\ + R(x^*) + (r_{\tau_k}(x^k, e^k) - \nabla F(x^k) - e^k)^T (x^{k+1} - x^*). \end{aligned}$$

Setting the step size $\tau_k = \tau \leq \frac{1}{L_F}$ yields

$$\begin{aligned} h(x^{k+1}) - h(x^*) \leq \nabla F(x^k)^T(x^k - x^*) - \tau \nabla F(x^k)^T (r_\tau(x^k, e^k)) + \frac{\tau}{2}\|r_\tau(x^k, e^k)\|_2^2 \\ + (r_\tau(x^k, e^k) - \nabla F(x^k) - e^k)^T (x^{k+1} - x^*). \end{aligned}$$

After rearranging and applying (3.12), we have

$$\begin{aligned} h(x^{k+1}) - h(x^*) &\leq \frac{\tau}{2}\|r_\tau(x^k, e^k)\|_2^2 + (r_\tau(x^k, e^k) - e^k)^T(x^{k+1} - x^*) \\ &= \frac{1}{2\tau}\|x^k - x^{k+1}\|_2^2 + \frac{1}{\tau}(x^k - x^{k+1})^T(x^{k+1} - x^*) - (e^k)^T(x^{k+1} - x^*) \\ &= \frac{1}{2\tau}(\|x^k - x^*\|_2^2 - \|x^{k+1} - x^*\|_2^2) - (e^k)^T(x^{k+1} - x^*) \\ &\leq \frac{1}{2\tau}(\|x^k - x^*\|_2^2 - \|x^{k+1} - x^*\|_2^2) + (C + \|x^*\|_2)\|e^k\|_2, \end{aligned}$$

where the last line follows from the Cauchy inequality and $\|x^k\|_2 \leq C$. Then we conclude that

$$0 \leq h(\bar{x}^K) - h(x^*) \leq \frac{1}{K}\sum_{k=1}^{K}(h(x^k) - h(x^*)) \leq \frac{1}{2K\tau}\|x^0 - x^*\|_2^2 + \frac{C + \|x^*\|_2}{K}\sum_{k=1}^{K}\|e^k\|_2, \tag{3.16}$$

where the first inequality is given by the optimality of x^* and the second inequality is due to the convexity of $h(x)$. Since $\|e^k\|_2 \le 1/k$, we have $\lim_{K\to\infty} \frac{1}{K}\sum_{k=1}^{K} \|e^k\|_2 = 0$. Thus the right-hand side of (3.16) goes to zero as $K \to \infty$, which completes the proof. ∎

The following theorem gives a linear convergence rate under the strong convexity assumption on $F(x)$.

THEOREM 8. *Suppose that Assumptions* 2(i)–(iv) *hold, R is convex, and F is μ-strongly convex. Let $\tau_k = \tau \le \frac{1}{L_F}$. Then we have $\|x^{k+1} - x^*\|_2^2 \le \eta \|x^k - x^*\|_2^2$ for $\eta < 1$ if*

$$d_k \ge \frac{-c \log(\mu/4 \cdot \|x^{k+1} - x^*\|_2) + b}{\min\{l, 1\}}, \tag{3.17}$$

where c and b are constants independent of μ, l, x^{k+1}, and x^*.

Proof. As deduced in the proof of Theorem 7, we have the quadratic bound (3.15):

$$h(x^{k+1}) \le F(x^k) - \tau_k \nabla F(x^k)^T (r_{\tau_k}(x^k, e^k)) + \frac{L_F \tau_k^2}{2} \|r_{\tau_k}(x^k, e^k)\|_2^2 + R(x^{k+1}).$$

By the strong convexity of $F(x)$, the convexity of $R(x)$, and (3.14), we have

$$\begin{aligned} h(x^{k+1}) \le F(x^*) &+ \nabla F(x^k)^T (x^k - x^*) - \frac{\mu}{2}\|x^k - x^*\|_2^2 - \tau_k \nabla F(x^k)^T (r_{\tau_k}(x^k, e^k)) \\ &+ \frac{L_F \tau_k^2}{2} \|r_{\tau_k}(x^k, e^k)\|_2^2 + R(x^*) + (r_{\tau_k}(x^k, e^k) - \nabla F(x^k) - e^k)^T (x^{k+1} - x^*). \end{aligned}$$

Using the choice of the step size $\tau_k = \tau \le \frac{1}{L_F}$, we have

$$\begin{aligned} h(x^{k+1}) - h(x^*) &\le \nabla F(x^k)^T (x^k - x^*) - \tau \nabla F(x^k)^T (r_\tau(x^k, e^k)) + \frac{\tau}{2}\|r_\tau(x^k, e^k)\|_2^2 \\ &\quad - \frac{\mu}{2}\|x^k - x^*\|_2^2 + (r_\tau(x^k, e^k) - \nabla F(x^k) - e^k)^T (x^{k+1} - x^*) \\ &= \frac{1}{2\tau}((1 - \mu\tau)\|x^k - x^*\|_2^2 - \|x^{k+1} - x^*\|_2^2) - (e^k)^T (x^{k+1} - x^*) \\ &\le \frac{1}{2\tau}((1 - \mu\tau)\|x^k - x^*\|_2^2 - \|x^{k+1} - x^*\|_2^2) + \|e^k\|_2 \|x^{k+1} - x^*\|_2, \end{aligned}$$

where the equality follows from (3.12). Together with $h(x^{k+1}) \ge h(x^*)$, we have

$$\|x^{k+1} - x^*\|_2^2 \le (1 - \mu\tau)\|x^k - x^*\|_2^2 + 2\tau \|e^k\|_2 \|x^{k+1} - x^*\|_2. \tag{3.18}$$

Since the error satisfies (3.11), the choice of d_k (3.17) guarantees $\|e^k\|_2 \le (\mu/4) \cdot \|x^{k+1} - x^*\|_2$. Plugging this upper bound on $\|e^k\|_2$ into (3.18) yields

$$\|x^{k+1} - x^*\|_2^2 \le \eta \|x^k - x^*\|_2^2,$$

where $\eta = \frac{1-\mu\tau}{1-\mu\tau/2} < 1$, which completes the proof. ∎

3.2.3. Warm-start analysis for PFPG.

The warm-start property takes U^k as an approximation to $V^{k+1}_{\mathcal{I}_{k+1}}$. The numerical performance of PFPG is often improved by using this information. We now show that the PFPG method also has a linear convergence rate as long as d_k is large enough (constant order) if the exact proximal gradient method converges linearly.

Let $\{s_1, s_2, \dots, s_n\}$ be a permutation of $\{1, 2, \dots, n\}$ such that $|\rho_k(\lambda_{s_i}(\mathcal{B}(x^k))|$ are in decreasing order, and define

$$\eta_k = \frac{|\rho_k(\lambda_{s_{p+1}}(\mathcal{B}(x^k))|}{|\rho_k(\lambda_{s_p}(\mathcal{B}(x^k))|}, \tag{3.19}$$

where p is the dimension of the eigenspace Range$(V^k_{\mathcal{I}_k})$. We should point out that under Assumption 2(iii), η_k is strictly smaller than 1 since $\mathcal{G}_k \geq l > 0$. An additional assumption is needed if we intend to refine the subspace from the previous one.

Assumption 3. $\|\sin\Theta(V^{k+1}_{\mathcal{I}_{k+1}}, V^k_{\mathcal{I}_k})\|_2 \leq c_4\|x^{k+1} - x^k\|_2 \ \forall k$.

Assumption 3 means that the eigenspaces at two consecutive iteration points are close enough. This is necessary since we use the eigenspace at the previous step as an initial value of the polynomial filter. This assumption is satisfied when the gap $\min_{i\in\mathcal{I}_k, j\notin\mathcal{I}_k} |\lambda_i - \lambda_j|$ is large enough by the Davis–Kahan $\sin\Theta$ theorem [3].

The next lemma shows the relationship between the two iterations in a recursive form, which plays a key role in our proof.

Lemma 9. *Suppose that Assumptions* 2(i)–(iv) *and* 3 *hold. Let $\mathcal{X}$ be the set of optimal solutions of problem* (3.2)*, and let* $\mathrm{dist}(x, \mathcal{X}) = \inf_{y\in\mathcal{X}} \|x-y\|_2$ *be the distance between x and $\mathcal{X}$. Then we have*

$$\|\sin\Theta(V^{k+1}_{\mathcal{I}_{k+1}}, U^{k+1})\|_2 \leq \eta^{q_k}_{k+1}(c_6\|\sin\Theta(V^k_{\mathcal{I}_k}, U^k)\|_2 + c_7\mathrm{dist}(x^k, \mathcal{X})), \tag{3.20}$$

where c_6 and c_7 are constants.

Proof. The update (3.8) can be seen as one iteration in the orthogonal iteration. Then it follows from [10, Theorem 8.2.2] that

$$\|\sin\Theta(V^{k+1}_{\mathcal{I}_{k+1}}, U^{k+1})\|_2 \leq \eta^{q_k}_{k+1}\|\tan\Theta(V^{k+1}_{\mathcal{I}_{k+1}}, U^k)\|_2 \leq \frac{\eta^{q_k}_{k+1}}{\sqrt{1-\gamma^2}}\|\sin\Theta(V^{k+1}_{\mathcal{I}_{k+1}}, U^k)\|_2, \tag{3.21}$$

where γ is the upper bound of $\|\sin\Theta(V^{k+1}_{\mathcal{I}_{k+1}}, U^k)\|_2$ defined in Assumption 2(i) and the second inequality follows from the trigonometric formula. Let e^k be defined as (3.11). Due to the boundedness of $\mathcal{A}$ and (2.21), we have

$$\|e^k\|_2 \leq c_5\|\sin\Theta(V^k_{\mathcal{I}_k}, U^k)\|_2, \tag{3.22}$$

where c_5 is a constant. Let x^* be the orthogonal projection of x^k onto the optimal set $\mathcal{X}$. According to the nonexpansiveness of the proximal operator, we obtain

$$\begin{aligned}
\|r_{\tau_k}(x^k, 0)\|_2 &= \|r_{\tau_k}(x^k, 0) - r_{\tau_k}(x^*, 0)\|_2 \\
&= \frac{1}{\tau_k}\|x^k - \mathrm{prox}_{\tau_k R}(x^k - \tau_k\nabla F(x^k)) - x^* + \mathrm{prox}_{\tau_k R}(x^* - \tau_k\nabla F(x^*))\|_2 \\
&\leq \frac{1}{\tau_k}(\|x^k - x^*\|_2 + \|\mathrm{prox}_{\tau_k R}(x^k - \tau_k\nabla F(x^k)) - \mathrm{prox}_{\tau_k R}(x^* - \tau_k\nabla F(x^*))\|_2) \\
&\leq \frac{1}{\tau_k}(\|x^k - x^*\|_2 + \|x^k - \tau_k\nabla F(x^k) - x^* + \tau_k\nabla F(x^*)\|_2) \\
&\leq \frac{2 + \tau_k L_F}{\tau_k}\|x^k - x^*\|_2 = \frac{2 + \tau_k L_F}{\tau_k}\mathrm{dist}(x^k, \mathcal{X}).
\end{aligned} \tag{3.23}$$

Then the upper bound of $\|x^k - x^{k+1}\|_2$ is given by

$$\begin{aligned}
&(3.24\text{a}) && \|x^k - x^{k+1}\|_2 \le \tau_k(\|r_{\tau_k}(x^k, e^k) - r_{\tau_k}(x^k, 0)\|_2 + \|r_{\tau_k}(x^k, 0)\|_2) \\
&(3.24\text{b}) && \le \tau_k\|e^k\|_2 + (2 + \tau_k L_F)\text{dist}(x^k, \mathcal{X}) \\
&(3.24\text{c}) && \le \tau_k c_5 \|\sin\Theta(V^k_{\mathcal{I}_k}, U^k)\|_2 + (2 + \tau_k L_F)\text{dist}(x^k, \mathcal{X}),
\end{aligned}$$

where (3.24a) follows from (3.12) and the triangle inequality, (3.24b) is obtained by applying (3.23) and the nonexpansiveness of $\text{prox}_{\tau_k R}$, and finally (3.24c) holds due to (3.22). From the identity (2.20), we obtain

(3.25)

$$\begin{aligned}
\|\sin\Theta(V^{k+1}_{\mathcal{I}_{k+1}}, U^k)\|_2 &= \|V^{k+1}_{\mathcal{I}_{k+1}}(V^{k+1}_{\mathcal{I}_{k+1}})^T - U^k(U^k)^T\|_2 \\
&\le \|V^k_{\mathcal{I}_k}(V^k_{\mathcal{I}_k})^T - U^k(U^k)^T\|_2 + \|V^{k+1}_{\mathcal{I}_{k+1}}(V^{k+1}_{\mathcal{I}_{k+1}})^T - V^k_{\mathcal{I}_k}(V^k_{\mathcal{I}_k})^T\|_2 \\
&= \|\sin\Theta(V^k_{\mathcal{I}_k}, U^k)\|_2 + \|\sin\Theta(V^{k+1}_{\mathcal{I}_{k+1}}, V^k_{\mathcal{I}_k}\|_2.
\end{aligned}$$

Finally, combining (3.21), (3.25), and (3.24) yields

$$\begin{aligned}
&(3.26\text{a}) && \|\sin\Theta(V^{k+1}_{\mathcal{I}_{k+1}}, U^{k+1})\|_2 \le \frac{\eta^{q_k}_{k+1}}{\sqrt{1-\gamma^2}}\|\sin\Theta(V^{k+1}_{\mathcal{I}_{k+1}}, U^k)\|_2 \\
&(3.26\text{b}) && \le \frac{\eta^{q_k}_{k+1}}{\sqrt{1-\gamma^2}}(\|\sin\Theta(V^k_{\mathcal{I}_k}, U^k)\|_2 + \|\sin\Theta(V^{k+1}_{\mathcal{I}_{k+1}}, V^k_{\mathcal{I}_k})\|_2) \\
&(3.26\text{c}) && \le \frac{\eta^{q_k}_{k+1}}{\sqrt{1-\gamma^2}}(\|\sin\Theta(V^k_{\mathcal{I}_k}, U^k)\|_2 + c_4\|x^k - x^{k+1}\|_2) \\
&(3.26\text{d}) && \le \frac{\eta^{q_k}_{k+1}}{\sqrt{1-\gamma^2}}((1 + \tau_k c_4 c_5)\|\sin\Theta(V^k_{\mathcal{I}_k}, U^k)\|_2 + c_4(2 + \tau_k L_F)\text{dist}(x^k, \mathcal{X})),
\end{aligned}$$

where inequality (3.26c) follows from Assumption 3. This completes the proof. □

Lemma 9 means that the principle angle between $V^{k+1}_{\mathcal{I}_{k+1}}$ and U^{k+1} is determined by the angle in the previous iteration and how far x^k is from the optimal set $\mathcal{X}$. As an intuitive interpretation, the polynomial-filtered subspace becomes more and more accurate when PFPG is close to convergence.

Assume that the exact proximal gradient method for (3.2) is linearly convergent, i.e.,

$$(3.27) \qquad \text{dist}(\text{prox}_{\tau R}(x^k - \tau_k \nabla F(x^k)), \mathcal{X}) \le \nu\,\text{dist}(x^k, \mathcal{X}), \nu \in (0, 1),$$

and $\nu^2 + \tau_k^2 c_5^2 < 1$. Define the error vector s^k and the matrix R^k by

$$(3.28) \qquad s^k = \begin{bmatrix} \text{dist}(x^k, \mathcal{X}) \\ \|\sin\Theta(V^k_{\mathcal{I}_k}, U^k)\|_2 \end{bmatrix} \quad \text{and} \quad R^k = \begin{bmatrix} \nu & \tau_k c_5 \\ \eta^{q_k}_{k+1} c_6 & \eta^{q_k}_{k+1} c_7 \end{bmatrix},$$

where c_5, c_6, c_7 are the constants in Lemma 9 and inequality (3.22). It can be verified that the spectral norm of R^k satisfies

$$\begin{aligned}
(3.29) \quad \|R^k\|_2^2 = &\frac{\nu^2 + \tau_k^2 c_5^2 + \eta^{2q_k}_{k+1}(c_6^2 + c_7^2)}{2} \\
&+ \sqrt{\left(\frac{\nu^2 + \tau_k^2 c_5^2 - \eta^{2q_k}_{k+1}(c_6^2 + c_7^2)}{2}\right)^2 + 4\eta^{2q_k}_{k+1}(c_6\nu + \tau_k c_7 c_5)^2}.
\end{aligned}$$

We are now ready to show that the PFPG method has a linear convergence rate if the polynomial degree is a constant, independent of the iteration k.

THEOREM 10. *Suppose that Assumptions* 2(i)–(iv) *and* 3 *as well as condition* (3.27) *hold. Then there exists a sufficiently large polynomial degree d_k independent of the iteration number k such that*

$$\|R^k\|_2 < \rho < 1. \tag{3.30}$$

Consequently, we have

$$\|s^{k+1}\|_2 \leq \rho \|s^k\|_2 \quad \forall k. \tag{3.31}$$

Proof. We first prove (3.30). Due to definition (3.29), we have $\lim_{\eta^k \to 0} \|R^k\|_2 = \nu^2 + \tau_k^2 c_5^2 < 1$. By Lemma 2 and Assumption 2(iii), η^k exponentially decays to zero as d^k goes to infinity. This, together with the continuity, gives (3.30).

Since $\mathcal{X}$ is closed, we denote by x_{prj} the orthogonal projection of $\text{prox}_{\tau_k R}(x^k - \tau_k \nabla F(x^k))$ onto the optimal set $\mathcal{X}$. Then we have

$$\begin{aligned}
\text{dist}(x^{k+1}, \mathcal{X}) &\leq \|\text{prox}_{\tau_k R}(x^k - \tau_k(\nabla F(x^k) + e^k)) - x_{prj}\|_2 \\
&\leq \|\text{prox}_{\tau_k R}(x^k - \tau_k \nabla F(x^k)) - \text{prox}_{\tau_k R}(x^k - \tau_k(\nabla F(x^k) + e^k))\|_2 \\
&\quad + \|\text{prox}_{\tau_k R}(x^k - \tau_k \nabla F(x^k)) - x_{prj}\|_2 \\
&\leq \text{dist}(\text{prox}_{\tau_k R}(x^k - \tau_k \nabla F(x^k)), \mathcal{X}) + \tau_k \|e^k\|_2 \\
&\leq \text{dist}(\text{prox}_{\tau_k R}(x^k - \tau_k \nabla F(x^k)), \mathcal{X}) + \tau_k c_5 \|\sin\Theta(V^k_{\mathcal{I}_k}, U^k)\|_2 \\
&\leq \nu \text{dist}(x^k, \mathcal{X}) + \tau_k c_5 \|\sin\Theta(V^k_{\mathcal{I}_k}, U^k)\|_2,
\end{aligned} \tag{3.32}$$

where the first inequality is from the definition of the distance function, the second inequality is from the triangle inequality, the third inequality follows from the nonexpansiveness of the proximal operator, the fourth inequality is shown in (3.22), and the last inequality is due to the linear convergence of the proximal gradient method.

Due to the definition of R^k, the two recursive formula (3.20) and (3.32) can be rewritten as $s^{k+1} \leq R^k s^k$. Since s^k and R^k are positive, (3.31) holds by taking the ℓ_2-norm. This completes the proof. □

Theorem 10 requires the linear convergence of the exact proximal gradient method. This condition is satisfied if $F(x)$ is μ-strongly convex and its gradient is Lipschitz continuous with constant L_F, in which case the constant ν in (3.27) is $1-\tau\mu$. The step size $\tau < \frac{\mu}{\mu^2 + c_5^2}$ is a sufficient condition to make $(1-\tau\mu)^2 + c_5^2\tau^2 < 1$ hold. From the proof, we also observe that $\text{dist}(x^k, \mathcal{X})$ and $\|\sin\Theta(V^k_{\mathcal{I}_k}, U^k)\|_2$ are recursively bounded by each other. By choosing a suitable d_k, the two errors can be guaranteed to decay simultaneously.

4. Application on the ADMM.

4.1. The PFAM method.

Consider the following standard SDP:

$$\min \langle C, X \rangle \ \text{ s.t. } \mathcal{A}X = b, X \succeq 0, \tag{4.1}$$

where $\mathcal{A}$ is a bounded linear operator defined in (3.1). Let $F(X) = 1_{\{X \succeq 0\}}(X)$ and $G(X) = 1_{\{\mathcal{A}X=b\}}(X) + \langle C, X \rangle$, where $1_\Omega(X)$ is the indicator function on a set Ω.

Then the Douglas–Rachford splitting (DRS) method [6] on the primal SDP (4.1) can be written as

$$Z^{k+1} = T_{\mathrm{DRS}}(Z^k) \triangleq \mathrm{prox}_{tG}(2\mathrm{prox}_{tF}(Z^k) - Z^k) - \mathrm{prox}_{tF}(Z^k) + Z^k,$$

which is equivalent to the ADMM on the dual problem. The explicit forms of $\mathrm{prox}_{tF}(Z)$ and $\mathrm{prox}_{tG}(Y)$ can be written as

$$\begin{aligned}\mathrm{prox}_{tF}(Z) &= \mathcal{P}_+(Z),\\ \mathrm{prox}_{tG}(Y) &= (Y + tC) - \mathcal{A}^*(\mathcal{A}\mathcal{A}^*)^{-1}(\mathcal{A}Y + t\mathcal{A}C - b),\end{aligned}$$

where $\mathcal{P}_+(Z)$ is the orthogonal projection operator onto the positive semidefinite cone.

The DRS update is also a special case of (1.1) with

$$\begin{aligned}\mathcal{T}(x, W) &= \mathrm{prox}_{tG}(2W - x) - W + x,\\ \Psi(X) &= \mathcal{P}_+(X),\\ \mathcal{B}(x) &= x.\end{aligned} \tag{4.2}$$

Note that both x and X are matrix variables. By Example 2.1, we conclude that Assumption 1 holds if Z^k only has a few positive eigenvalues, in which case $\Psi = \Phi = \mathcal{P}_+$. Thus (4.1) can be viewed as a low-rank problem if $\mathcal{P}_+(Z^k)$ is numerically observed to be low-rank during the DRS update.

Therefore, the PFAM method can be written as

$$Z^{k+1} = \mathrm{prox}_{tG}(2\mathcal{P}_+(\mathcal{P}_{U^k}(Z^k)) - Z^k) - \mathcal{P}_+(\mathcal{P}_{U^k}(Z^k)) + Z^k, \tag{4.3}$$

$$U^{k+1} = \mathrm{orth}(\rho_{k+1}^{q_{k+1}}(Z^{k+1})U^k). \tag{4.4}$$

4.2. Convergence analysis.

The convergence of the PFAM method (4.3)–(4.4) can be established directly by applying Theorem 5.

COROLLARY 11. *Suppose that Assumptions* 2(i)–(iii) *hold and* $\{Z^k\}$ *is generated by the PFAM method* (4.3)–(4.4)*. Then we have*

$$\|Z^k - T_{DRS}(Z^k)\|_F = o\left(1/\sqrt{k}\right)$$

if d_k *satisfies* (2.22).

Proof. The proof follows from Theorem 5 by verifying Assumption 2(iv)–(v) for the operator $\mathcal{T}$ defined in (4.2) and the averageness of the operator T_{DRS}. According to [4, sections 1.4, Proposition 2], T_{DRS} is actually $\frac{1}{2}$-averaged. The spectral operator Φ in DRS is $\mathcal{P}_+$, which is also the proximal operator of $1_{\{X \succeq 0\}}$. Since the proximal operators are nonexpansive, Φ is Lipschitz continuous with constant 1. Together with the nonexpansiveness of prox_{tG} [4, Proposition 1], we obtain

$$\begin{aligned}\|\mathcal{T}(x, W_1) - \mathcal{T}(x, W_2)\|_F &= \|\mathrm{prox}_{tG}(2W_1 - x) - W_1 - \mathrm{prox}_{tG}(2W_2 - x) + W_2\|_F\\ &\le \|\mathrm{prox}_{tG}(2W_1 - x) - \mathrm{prox}_{tG}(2W_2 - x)\|_F + \|W_1 - W_2\|_F\\ &\le 2\|W_1 - W_2\|_F + \|W_1 - W_2\|_F = 3\|W_1 - W_2\|_F,\end{aligned}$$

which completes the proof. ∎

5. Numerical experiments.

This section reports numerical results on a set of applications based on the spectral optimization problems. All experiments are performed on a Linux server with two twelve-core Intel Xeon E5-2680 v3 processors at 2.5 GHz and a total amount of 128 GB shared memory. All reported time is wall-clock time in seconds.

5.1. Implementation details. First we give a brief description of the implementation details of the subspace update step (2.10) in the general framework. The polynomial d_k is manually set to a fixed value, which we find effective enough under the warm-start setting. We use $d_k = 5$ throughout the iterations unless another value is specified. For better robustness of the framework, a few extra columns (guard vectors) are added to U^k in (2.10) for a slightly larger subspace. Assumption 1 still holds for the cases with guard vectors since the choice of $\mathcal{I}$ is not unique. In practice, the total number of columns of U^k is set to $\max\{5, \lceil 1.2 r_k \rceil\}$, where r_k is an estimate of $|\mathcal{I}_k|$.

The choice of the filtering interval $[a, b]$ is problem dependent. Generally speaking, it can be estimated either by inexact solvers or the eigenvalues from the previous iteration. For example, if all positive eigenvalues are required, then the smallest eigenvalue λ_n is a good approximation of a, and 0 naturally serves the estimation of b. In our implementation λ_n is updated by `eigs` with a low accuracy every 10 iterations. If p largest eigenvalues are required, then the same technique is used to estimate a, and b can be approximated by λ_p from the previous iteration. This strategy can also be applied to other applications that are not considered in this paper.

5.2. Nearest correlation estimation. Given a matrix $G \in \mathcal{S}^n$, the unweighted nearest correlation estimation (NCE) problem is to solve the semidefinite optimization problem

$$\min \ \frac{1}{2}\|G - X\|_F^2 \quad \text{s.t.} \quad \operatorname{diag}(X) = \mathbf{1}_n, X \succeq 0. \tag{5.1}$$

The dual problem of (5.1) is

$$\min \frac{1}{2}\|\mathcal{P}_+(G + \operatorname{Diag}(x))\|_F^2 - \mathbf{1}_n^T x. \tag{5.2}$$

Note that (5.2) can be rewritten as a spectral function minimization problem as (3.2), whose objective function and gradient are

$$\min F(x) := f \circ \lambda(G + \operatorname{Diag}(x)) - \mathbf{1}_n^T x, \tag{5.3}$$

$$\nabla F(x) \ = \ \operatorname{diag}(\mathcal{P}_+(G + \operatorname{Diag}(x))) - \mathbf{1}_n, \tag{5.4}$$

where $f(\lambda) = \frac{1}{2}\sum_{i=1}^n \max(\lambda_i, 0)^2$.

Problem (5.2) can be solved via the gradient method, i.e.,

$$x^{k+1} = x^k - \tau \nabla F(x^k) = x^k - \tau \left(\operatorname{diag}(\mathcal{P}_+(G + \operatorname{Diag}(x^k))) - \mathbf{1}_n\right),$$

which can be regarded as a special case of (1.1) with

$$\mathcal{T}(x, W) = x - \tau\left(\operatorname{diag}(W) - \mathbf{1}_n\right), \quad \Psi(X) = \mathcal{P}_+(X), \quad \mathcal{B}(x) = G + \operatorname{Diag}(x).$$

As argued in Example 2.1, Assumption 1 is satisfied if $\mathcal{P}_+(G + \operatorname{Diag}(x^k))$ is low rank for k sufficiently large (by numerical observation).

We generate synthetic NCE problem data based on the first three examples in [20].

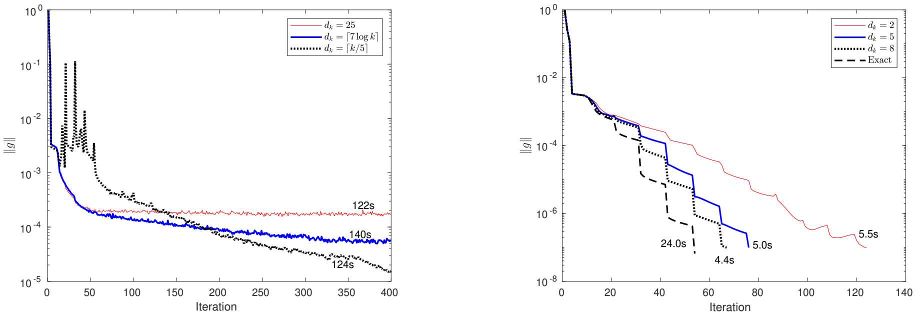
Example 5.1. $C \in \mathcal{S}^n$ is a randomly generated correlation matrix using the MATLAB function `gallery('randcorr', n)`. R is an n-by-n random matrix whose entries R_{ij} are sampled from the uniform distribution in $[-1, 1]$. Then the matrix G is set to $G = C + R$.

Example 5.2. $G \in \mathcal{S}^n$ is a randomly generated matrix with G_{ij} satisfying the uniform distribution in $[-1, 1]$ if $i \neq j$. The diagonal elements G_{ii} is set to 1 for all i.

Example 5.3. $G \in \mathcal{S}^n$ is a randomly generated matrix with G_{ij} satisfying the uniform distribution in $[0, 2]$ if $i \neq j$. The diagonal elements G_{ii} is set to 1 for all i.

It is worth mentioning that we apply an extrapolation-based acceleration technique proposed in [22] to overcome the slow convergence of the first-order methods. Briefly, we perform a linear combination of the points x^k every $l + 2$ iterations to obtain a better estimate $\tilde{x} = \sum_{i=0}^{l} \tilde{c}_i x^{k-l+i}$, where $\tilde{c}$ contains $l + 1$ precomputed coefficients. One can refer to [22] for more details.

First, we demonstrate the impact of the polynomial degree d_k on PFPG on Example 5.3 with $n = 2000$. We run PFPG using different d_k without and with the warm-start setting, and then we report the relative norm of the gradient $\|g\|$ versus iteration in Figures 2(a) and 2(b), respectively. In Figure 2(a) for the case without warm start, a Gaussian random matrix is used as U^k for each iteration. If d_k is set as a constant 25, then the PFPG does not converge. The norm $\|g\|$ stagnates around $4 \cdot 10^{-4}$. If d_k is chosen as $d_k = \lceil 7 \log k \rceil$, then a sublinear convergence rate can be observed, which is consistent with Theorems 5 and 7. Finally, if d_k has a linear growth, e.g., $d_k = \lceil k/5 \rceil$, then a linear convergence rate is achieved, which justifies Theorem 8. The behavior under the warm-start setting is shown in Figure 2(b). Theorem 10 indicates that d_k can be set as a constant. For $d_k = 2, 5, 8$, the performance of PFPG is nearly the same as the exact gradient method during the first 10 iterations. However, the polynomial degree has an impact on the extrapolation-based acceleration step. We point out that for general cases, using a smaller d_k often leads to more iterations even if the extrapolation step is not considered.



(a) Without warm start

(b) With warm start

FIG. 2. *Convergence behavior of PFPG.*

We next compare the efficiency of three methods on NCE problems, namely, the gradient method with a fixed step size (Grad) and our proposed method (PFPG), and the semismooth Newton method (Newton) proposed in [20]. In all three examples, the dimension n is set to various numbers from 500 to 4000. The initial point x^0 is set to $\mathbf{1}_n - \mathrm{diag}(G)$. The stopping criteria is $\|\nabla F(x)\| \leq 10^{-7} \|\nabla F(x_0)\|$. The reason not to use a higher accuracy is that first-order methods (such as Grad) are not designed for obtaining high accuracy solutions. We also report the number of iterations (denoted by "Iter"), the objective function value f, and the relative norm of the gradient $\|g\| = \|\nabla F(x)\| / \|\nabla F(x^0)\|$. The results are shown in Table 1.

Table 1
Results of NCE problems.

Results of Example 5.1

n	Grad				PFPG				Newton			
	Time	Iter	f	$\|g\|$	Time	Iter	f	$\|g\|$	Time	Iter	f	$\|g\|$
500	0.8	28	1.1e+05	7.9e-08	0.7	40	1.1e+05	4.1e-08	0.6	8	1.1e+05	1.7e-08
1000	2.6	28	3.2e+05	5.5e-08	1.1	46	3.2e+05	2.6e-08	1.4	8	3.2e+05	1.2e-08
1500	6.2	28	5.9e+05	7.0e-08	2.5	46	5.9e+05	8.2e-08	3.3	8	5.9e+05	1.0e-08
2000	13.5	31	9.2e+05	8.0e-08	4.2	46	9.2e+05	4.9e-08	5.9	8	9.2e+05	8.8e-09
2500	24.9	34	1.3e+06	1.7e-08	7.2	47	1.3e+06	6.9e-08	9.6	8	1.3e+06	7.9e-09
3000	43.2	34	1.7e+06	2.8e-08	13.2	52	1.7e+06	1.8e-08	15.7	8	1.7e+06	7.2e-09
4000	110.6	34	2.7e+06	1.4e-08	25.0	52	2.7e+06	2.4e-08	36.1	8	2.7e+06	6.3e-09

Results of Example 5.2

n	Grad				PFPG				Newton			
	Time	Iter	f	$\|g\|$	Time	Iter	f	$\|g\|$	Time	Iter	f	$\|g\|$
500	0.5	16	8.9e+03	2.2e-09	0.6	16	8.9e+03	1.0e-08	0.5	5	8.9e+03	1.6e-07
1000	1.6	16	2.6e+04	7.4e-08	1.0	17	2.6e+04	8.3e-08	1.2	6	2.6e+04	1.2e-07
1500	4.0	17	5.0e+04	6.6e-08	2.2	20	5.0e+04	9.1e-08	2.5	6	5.0e+04	9.6e-08
2000	7.5	17	7.8e+04	8.9e-08	4.8	22	7.8e+04	2.4e-08	4.7	6	7.8e+04	8.3e-08
2500	13.8	19	1.1e+05	7.2e-08	7.4	22	1.1e+05	4.5e-08	7.6	6	1.1e+05	7.5e-08
3000	25.9	21	1.5e+05	8.9e-08	12.0	22	1.5e+05	8.1e-08	12.3	6	1.5e+05	6.9e-08
4000	72.5	22	2.3e+05	3.4e-09	27.2	25	2.3e+05	9.9e-08	28.6	6	2.3e+05	6.0e-08

Results of Example 5.3

n	Grad				PFPG				Newton			
	Time	Iter	f	$\|g\|$	Time	Iter	f	$\|g\|$	Time	Iter	f	$\|g\|$
500	0.9	33	1.3e+05	7.4e-08	0.7	43	1.3e+05	1.3e-08	0.6	8	1.3e+05	1.8e-07
1000	3.8	43	5.0e+05	2.0e-08	1.1	54	5.0e+05	3.0e-08	1.6	9	5.0e+05	1.3e-07
1500	11.3	54	1.1e+06	2.6e-08	2.4	65	1.1e+06	7.1e-08	4.0	10	1.1e+06	1.0e-07
2000	22.6	54	2.0e+06	4.3e-08	5.0	76	2.0e+06	8.3e-08	7.2	10	2.0e+06	9.0e-08
2500	60.9	87	3.1e+06	3.6e-08	10.6	120	3.1e+06	4.2e-08	12.1	10	3.1e+06	8.0e-08
3000	104.0	87	4.5e+06	6.2e-08	16.1	129	4.5e+06	7.8e-08	19.3	10	4.5e+06	7.4e-08
4000	278.0	91	8.0e+06	7.9e-08	32.8	142	8.0e+06	7.3e-08	44.2	10	8.0e+06	6.4e-08

The following observations can be drawn from Table 1:

- All three methods reach the required accuracy in all test instances. For large problems such as $n = 4000$, PFPG usually provides at least three times speedup over Grad. For some large problems such as $n = 4000$ in Example 5.3, PFPG has nearly nine times speedup.
- The Newton method usually converges within 10 iterations, since it enjoys the quadratic convergence property. The main cost of the Newton method is full EVDs and solving linear systems, which make each iteration slow. However, PFPG benefits from the subspace extraction so that the cost of each iteration is greatly reduced, making it very competitive against the Newton method.
- PFPG usually needs more iterations than Grad, which results from the inexactness of the Chebyshev filters. The error can be controlled by their degree. Using higher order polynomials will reduce the number of iterations but increase the cost per iteration. Moreover, the number of iterations does not increase much thanks to the warm-start property of the filters.

It should be pointed out that the benefit brought by PFPG is highly dependent on the number of positive eigenvalues r_+ at $G + \mathrm{Diag}(x^*)$. We have checked this number and found out that for Examples 5.1 and 5.2, r_+ is approximately $0.1n$, and $r_+ \approx 0.05n$ for Example 5.3. That is the reason PFPG is able to provide higher speedups in Example 5.3. As a conclusion we observe that PFPG is not suitable for $r_+ \approx 0.5n$.

Finally we test the three methods on real datasets. The test matrices are selected from the invalid correlation matrix repository.[1] Since we are more interested in large NCE problems, we choose three largest matrices from the collection, whose details are presented in Table 2. The label "Opt. rank" denotes the rank of the optimal point X^* in (5.1). Note that for matrix "cor3120" the solution is not low rank. However, the orthogonal projection can be performed with respect to the negative definite part of the variable, which is low rank in this case.

Table 2
Invalid correlation matrices from real applications.

Name	Size	Opt. rank	Description
cor1399	1399 × 1399	154	matrix constructed from stock data
cor3120	3120 × 3120	3025	matrix constructed from stock data
bccd16	3250 × 3250	5	matrix from EU bank data

Again, we compare the performance of Grad, PFPG, and Newton on the three matrices. The stopping criteria is set to $\|\nabla F(x)\| \le 10^{-5}\|\nabla F(x_0)\|$. The results are shown in Table 3.

Table 3
Results of NCE problems on real data.

Matrix	Grad				PFPG				Newton			
	Time	Iter	f	$\|g\|$	Time	Iter	f	$\|g\|$	Time	Iter	f	$\|g\|$
cor1399	6.0	32	6.5e+04	1.3e-06	2.6	32	6.5e+04	4.1e-06	1.9	5	6.5e+04	4.4e-06
cor3120	76.0	43	9.3e+04	2.4e-06	12.3	43	9.3e+04	4.8e-06	8.9	4	9.3e+04	3.8e-06
bccd16	18.9	16	1.4e+06	3.3e-09	4.2	16	1.4e+06	6.2e-07	5.0	2	1.4e+06	3.4e-07

For real data, PFPG is still able to perform multifold speedups over the traditional gradient method. Though not always faster than Newton, PFPG is very competitive against the second-order method.

5.3. Max eigenvalue optimization. Consider the max eigenvalue optimization problem with the form

$$\min_x F(x) + R(x) := \lambda_1(\mathcal{B}(x)) + R(x), \tag{5.5}$$

where $\mathcal{B}(x) = G + \mathcal{A}^*(x)$. The subgradient of $F(x)$ is

$$\partial F(x) = \{\mathcal{A}(U_1 S U_1^T) \mid S \succeq 0, \operatorname{tr}(S) = 1\},$$

where $U_1 \in \mathbb{R}^{n \times r_1}$ is the subspace spanned by eigenvectors of $\lambda_1(\mathcal{B}(x))$ with multiplicity r_1. If $r_1 = 1$, then $\partial F(x)$ contains only one element, and hence the subgradient becomes the gradient. If $r_1 > 1$, $F(x)$ is only subdifferentiable. We point out that the $r_1 > 1$ case is generally harder than $r_1 = 1$. We only consider $r_1 = 1$ (differentiable case) in this subsection.

Since $F(x)$ is differentiable, the PFPG method can be applied to solve (5.5), which is given by

$$x^{k+1} = \operatorname{prox}_{\tau R}(x^k - \tau\mathcal{A}(u_1 u_1^T)),$$

where u_1 is the eigenvector of $\lambda_1(\mathcal{B}(x^k))$. Without loss of generality, we assume $\lambda_1(\mathcal{B}(x)) > 0$ $\forall x = x^k$. Otherwise, $\mathcal{B}(x)$ can be replaced by $\mathcal{B}(x) + tI$ for some $t > 0$,

[1] Available at https://github.com/higham/matrices-correlation-invalid.

but the solution of (5.5) is still the same one. Consequently, we have

$$\mathcal{T}(x, W) = \mathrm{prox}_{\tau R}(x - \tau \mathcal{A}(W)), \quad \Psi(X) = u_1 u_1^T.$$

Also, $\Psi(\cdot)$ satisfies the low-rank property (Assumption 1) around x^* with $\mathcal{I} = \{1\}$ and

$$(\psi(\lambda))_i = (\phi(\lambda))_i = \begin{cases} 1, & i = 1, \\ 0 & \text{otherwise.} \end{cases}$$

The phase recovery formulation in [8] can be regarded as an application of problem (5.5). The smooth part $F(x) = \lambda_1(\mathcal{B}(x)) := \lambda_1(\mathcal{A}^*(x))$ is defined as follows. Let the known diagonal matrices $C_k \in \mathbb{C}^{n \times n}$ be the encoding diffraction patterns corresponding to the kth "mask" ($k = 1, \ldots, L$). The measurements of an unknown signal $x_0 \in \mathbb{C}^n$ are given by

$$b = \mathcal{A}(x_0 x_0^*) := \mathrm{diag}\left[\begin{pmatrix} \mathcal{F}C_1 \\ \vdots \\ \mathcal{F}C_L \end{pmatrix} (x_0 x_0^*) \begin{pmatrix} \mathcal{F}C_1 \\ \vdots \\ \mathcal{F}C_L \end{pmatrix}^* \right], \tag{5.6}$$

where $\mathcal{F}$ is the unitary discrete Fourier transform (DFT). The adjoint of $\mathcal{A}$ can be written as

$$\mathcal{A}^* y := \sum_{k=1}^{L} C_k^* \mathcal{F}^* \mathrm{Diag}(y) \mathcal{F} C_k. \tag{5.7}$$

The nonsmooth part $R(x)$ is defined as the indicator function of the set $\{x \mid \langle b, x \rangle - \|x\|_2 \geq 1\}$.

In the experiments of the phase retrieval problem, both synthetic and real data are considered. The noiseless synthetic data are generated as follows. For each value of $L = 7, \ldots, 12$, we generate random complex Gaussian vectors x_0 of length $n = 4096$ and a set of L random complex Gaussian masks C_k.

We use the GAUGE algorithm proposed in [8] to solve the phase recovery problem, in which the MATLAB built-in function `eigs` is used as the eigenvalue subproblem solver. Then we insert our polynomial filters to the original GAUGE implementation to obtain the polynomial-filtered GAUGE algorithm (PFGAUGE). It should be mentioned that though GAUGE is essentially a gradient method, it has many variants to handle different cases. A highlight of GAUGE is a so-called primal-dual refinement step, which is able to reduce the iteration steps drastically. This feature also has great impact on the performance of GAUGE and PFGAUGE. We should point out that the framework proposed in [12, Algorithm 1] can be applied to (5.5) if the subproblem is solved efficiently. Since tuning a good solver requires many efforts, we restrict our comparison between GAUGE and PFGAUGE.

The results of noiseless synthetic data are presented in Table 4, where "Iter" denotes the number of outer iteration. In addition, we record the number of DFT calls (nDFT) and the number of eigensolvers or polynomial filters evaluated, which is denoted by nEigs and nPF, respectively. The label "Err" stands for the relative mean squared error of the solution $\|xx^* - x_0 x_0^*\|_F / \|x_0\|_2$. The polynomial degree d_k is set to 8.

The following observations should be clear from the results:

- Both GAUGE and PFGAUGE obtain the required accuracy in all test cases. The problem becomes harder to solve as L decreases from 12 to 7.

Table 4
Random Gaussian signals, noiseless, $n = 4096$.

	GAUGE				PFGAUGE			
L	Time	Iter	nDFT (nEigs)	Err	Time	Iter	nDFT (nPF)	Err
12	10.85	4	57966 (12)	1.4e-06	9.66	11	74442 (30)	2.0e-06
11	15.25	5	83298 (16)	2.0e-06	10.43	11	77880 (36)	2.0e-06
10	22.63	8	118530 (20)	2.3e-06	11.20	14	79435 (37)	2.6e-06
9	35.89	11	176864 (26)	2.0e-06	16.78	20	118377 (56)	3.4e-06
8	49.88	15	235720 (36)	2.8e-06	15.52	23	105660 (53)	3.0e-06
7	160.64	40	779559 (83)	2.6e-06	33.66	40	208915 (84)	3.4e-06

- In general, it takes more iterations for PFGAUGE to converge compared with the original GAUGE. An exception is the $L = 7$ case, in which the iterations are nearly the same.
- PFGAUGE gradually becomes faster as L decreases. For example, it gives 5 times speedup at $L = 7$. This can be also concluded from nDFT. After examining the iteration process, we find out that the eigenvalue subproblem is hard to solve when L is small, and thus it takes more iterations for `eigs` to converge. However, since we only apply a polynomial filter to the iteration point, the cost of DFT can be greatly reduced. For the easy cases such as $L = 12$, the performances of the two solvers are similar.

Next we conduct experiments on 2D real data in order to assess the speedup of Chebyshev filters on problems with larger size. In this scenario the measured signal x_0 is grayscale images, summarized in Table 5. For simplicity, the images are numbered from 1 to 12. Cases 1 and 2 are selected from the MATLAB image database; cases 3 to 12 are from the HubbleSite Gallery.[2] The largest problem (no. 12) has the size $n = 4 \cdot 10^6$, which brings a huge eigenvalue problem in each iteration. For each example we generate 10 and 15 octanary masks. The results are summarized in Table 6. Column "f" records the values of the dual objective function. "Gap" stands for the duality gap of the problem. The other columns have the same meanings as in Table 4. Since the data size is very large, we terminate the algorithms as soon as they exceed a timeout threshold, which is set to 18,000 seconds in the experiment.

Table 5
Image data for 2D signals.

No.	Name	Size	No.	Name	Size
1	coloredChips	518 × 391	2	lighthouse	480 × 640
3	asteriods(S)	500 × 500	4	giantbubble(S)	600 × 570
5	supernova(S)	640 × 426	6	nebula(S)	800 × 675
7	crabnebula(S)	1000 × 1000	8	asteriods(L)	1000 × 1000
9	giantbubble(L)	1200 × 1140	10	nebula(L)	1600 × 1350

Table 6 actually shows the effectiveness of PFGAUGE:

- For all test cases, PFGAUGE successfully converges within the time limit, while the original GAUGE method times out in most test cases with large data. GAUGE also fails in case 2 with $L = 10$. After checking the output of the algorithm, the reason is that the eigensolver fails to converge in some iterations.

[2] See http://hubblesite.org/images/gallery.

Table 6
Phase retrieval comparisons on 2D real signal.

		GAUGE					PFGAUGE				
No.	L	Time	Iter	nDFT (nEigs)	f	Gap	Time	Iter	nDFT (nPF)	f	Gap
1	15	1155.89	4	3e+05 (19)	1.0e+05	5.0e-06	549.97	7	2e+05 (29)	1.0e+05	1.4e-06
	10	3704.66	9	1e+06 (26)	1.0e+05	1.0e-05	210.36	9	6e+04 (26)	1.0e+05	7.9e-06
2	15	3043.72	4	1e+06 (19)	1.1e+05	5.7e-06	722.95	6	2e+05 (26)	1.1e+05	1.5e-06
	10	21898.10	11	8e+06 (87)	1.1e+05	NaN	158.82	8	5e+04 (25)	1.1e+05	5.0e-06
3	15	276.71	4	8e+04 (18)	1.4e+04	4.9e-06	230.50	5	7e+04 (21)	1.4e+04	6.2e-06
	10	385.09	13	1e+05 (34)	1.4e+04	9.5e-06	208.24	10	6e+04 (28)	1.4e+04	9.4e-06
4	15	9583.33	12	3e+06 (35)	2.3e+04	1.3e-06	295.84	6	8e+04 (26)	2.3e+04	3.6e-06
	10	7433.17	18	2e+06 (46)	2.3e+04	7.9e-06	238.08	7	6e+04 (24)	2.3e+04	9.7e-06
5	15	1735.88	5	5e+05 (20)	2.2e+04	3.9e-06	622.25	9	2e+05 (29)	2.2e+04	7.5e-06
	10	1872.62	6	5e+05 (22)	2.2e+04	1.5e-06	291.96	7	7e+04 (24)	2.2e+04	1.8e-06
6	15	643.35	7	1e+05 (24)	6.7e+04	2.7e-06	523.80	9	8e+04 (30)	6.7e+04	6.4e-06
	10	21943.32	15	3e+06 (42)	6.7e+04	1.2e-02	989.89	4	1e+05 (10)	6.7e+04	6.7e-06
7	15	2252.18	10	2e+05 (30)	6.3e+04	8.3e-06	1116.42	9	1e+05 (30)	6.3e+04	7.6e-06
	10	2530.10	22	2e+05 (54)	6.3e+04	8.4e-06	981.21	17	9e+04 (46)	6.3e+04	6.9e-06
8	15	18543.56	5	1e+06 (22)	5.8e+04	1.0e-03	4050.70	9	3e+05 (30)	5.8e+04	5.8e-06
	10	19371.82	11	1e+06 (33)	5.8e+04	1.0e-03	1209.17	11	9e+04 (34)	5.8e+04	8.9e-06
9	15	20239.04	2	1e+06 (17)	9.1e+04	7.8e-02	7046.67	7	5e+05 (27)	9.1e+04	9.9e-06
	10	19610.56	8	1e+06 (19)	9.1e+04	6.0e-01	3892.26	6	2e+05 (14)	9.1e+04	4.7e-06
10	15	7680.27	10	3e+05 (32)	2.7e+05	4.1e-06	4287.95	9	2e+05 (30)	2.7e+05	5.8e-06
	10	21958.19	5	8e+05 (31)	2.7e+05	1.7e-01	4042.50	24	1e+05 (58)	2.7e+05	4.8e-06

- For cases where both algorithms converge, PFGAUGE is able to provide over 20 times speedup. This can be verified by the nDFT column of the two methods as well. The reason is that the traditional eigensolver `eigs` is not scalable enough to deal with large problems. The convergence is very slow, and thus the accuracy of the solutions is not tolerable. However, our polynomial-filtered algorithm is able to extract a proper low-rank subspace that contains the desired eigenvectors quickly, and hence the performance is much better.
- PFGAUGE sometimes needs fewer iterations than GAUGE. This is because `eigs` fails to converge during the iterations, and thus GAUGE produces wrong updates.

5.4. ADMM for SDP. In this section we show the numerical results of our PFAM. First we test PFAM on the two-body reduced density matrix (2-RDM) problem. The problem has a block diagonal structure with respect to the variable X, with each block being a low-rank matrix. Thus the polynomial filters can be applied to each block to reduce the cost. We use the preprocessed dataset in [15]. Only cases with maximum block size greater than 1000 are selected. The details of the data can be found in [17].

Table 7 contains the numerical results of ADMM and PFAM. Columns “Pobj,” “Pinf,” “Dinf,” and “Gap” record the primal objective function value, primal infeasibility, dual infeasibility, and duality gap at the solution, respectively. The overall tolerance of each algorithm is set to 10^{-4}. A high accuracy is not needed in this application because we use ADMM to generate initial solutions for second-order methods such as SSNSDP [15].

As observed in Table 7, PFAM is two times faster than ADMM in large 2-RDM problems, such as CH2, C2, and NH3. We mention that the speedup provided by

TABLE 7
Results of ADMM and PFAM on 2-RDM problems.

Data	ADMM						PFAM					
Name	Time	Itr	Pobj	Pinf	Dinf	Gap	Time	Itr	Pobj	Pinf	Dinf	Gap
AlH	73	373	2.46e+02	2.1e-05	9.9e-05	4.8e-05	71	377	2.46e+02	2.1e-05	9.9e-05	4.7e-05
B2	428	800	5.73e+01	9.6e-05	9.9e-05	1.2e-04	257	908	5.73e+01	8.1e-05	9.9e-05	1.3e-04
BH	476	1058	2.73e+01	9.6e-05	9.5e-05	3.7e-04	322	1026	2.73e+01	9.1e-05	9.8e-05	3.5e-04
BH3O	727	437	1.31e+02	6.9e-05	9.9e-05	2.0e-04	530	458	1.31e+02	2.8e-05	9.9e-05	1.7e-04
BN	144	748	9.33e+01	3.0e-05	9.9e-05	1.1e-04	93	739	9.33e+01	4.2e-05	9.9e-05	1.2e-04
BeO	91	488	1.02e+02	7.3e-05	9.9e-05	8.8e-05	71	479	1.02e+02	4.8e-05	9.9e-05	9.6e-05
C2	1763	831	9.10e+01	5.8e-05	9.9e-05	1.8e-04	1032	911	9.10e+01	5.2e-05	9.9e-05	1.7e-04
CH	453	969	4.12e+01	9.7e-05	9.8e-05	3.6e-04	303	1009	4.12e+01	9.8e-05	9.6e-05	3.4e-04
CH2	1516	1350	4.50e+01	8.0e-05	9.9e-05	3.9e-04	754	1339	4.50e+01	8.3e-05	9.9e-05	3.6e-04
CH2	1698	1496	4.48e+01	9.8e-05	9.2e-05	3.4e-04	782	1428	4.48e+01	9.8e-05	9.5e-05	3.5e-04
CH3	2028	1113	4.94e+01	5.5e-05	9.9e-05	1.4e-04	1010	1149	4.94e+01	6.7e-05	9.9e-05	2.5e-04
CH3N	756	450	1.27e+02	4.8e-05	9.9e-05	1.6e-04	491	456	1.27e+02	5.9e-05	9.9e-05	1.5e-04
CN	83	439	1.11e+02	8.9e-05	9.9e-05	8.8e-05	68	437	1.11e+02	9.8e-05	9.5e-05	8.3e-05
F-	265	648	9.96e+01	9.9e-05	9.5e-05	2.9e-04	189	639	9.96e+01	9.0e-05	9.9e-05	2.4e-04
H2O	819	717	8.54e+01	9.5e-05	9.9e-05	3.4e-04	495	809	8.54e+01	9.8e-05	9.0e-05	3.4e-04
HF	268	576	1.05e+02	9.9e-05	9.3e-05	2.2e-04	210	575	1.05e+02	9.6e-05	9.8e-05	2.1e-04
HLi2	189	622	1.91e+01	7.5e-05	9.8e-05	1.4e-04	126	656	1.91e+01	7.9e-05	9.8e-05	1.5e-04
HNO	251	451	1.60e+02	7.4e-05	9.9e-05	4.1e-05	207	446	1.60e+02	7.8e-05	9.9e-05	3.9e-05
LiH	472	1030	9.00e+00	9.0e-05	4.2e-05	1.1e-04	369	1037	9.00e+00	9.1e-05	5.4e-05	1.2e-04
LiOH	141	468	9.57e+01	9.8e-05	8.5e-05	1.4e-04	107	452	9.57e+01	9.5e-05	9.6e-05	1.6e-04
NH	443	943	5.86e+01	8.6e-05	9.9e-05	3.0e-04	298	946	5.86e+01	8.8e-05	9.9e-05	3.1e-04
NH	405	870	5.86e+01	8.9e-05	9.9e-05	2.8e-04	290	937	5.86e+01	9.8e-05	9.1e-05	2.7e-04
NH2-	967	849	6.32e+01	9.8e-05	9.3e-05	4.0e-04	502	854	6.32e+01	9.7e-05	9.7e-05	3.9e-04
NH3	3775	925	6.82e+01	7.3e-05	9.9e-05	3.2e-04	1819	967	6.82e+01	9.7e-05	9.8e-05	3.4e-04
SiH4	212	292	3.12e+02	2.8e-05	9.9e-05	3.8e-05	193	292	3.12e+02	2.8e-05	9.9e-05	2.8e-05

PFAM depends on the number of large low-rank variable blocks. One may notice that PFAM is less effective on some problems like AlH and BeO. The main reason is that there are very few large blocks in these datasets, and polynomial filters are not designed for small matrices. In fact, it is always observed that a full eigenvalue decomposition is faster than any other truncated eigensolvers or polynomial-filtered methods in these small cases. Another minor reason is that some blocks are not low rank, causing the performance of PFAM to be limited. This observation again addresses the feature of PFAM: it is the most suitable for large-scale low-rank problems.

6. Conclusion.

In this paper, we propose a framework of polynomial-filtered methods for low-rank optimization problems. Our motivation is based on the key observation that the iteration points lie in a low-rank subspace of $\mathbb{R}^{n\times n}$. Therefore, the strategy is to extract this subspace approximately, and then perform one update based on the orthogonal projection of the current iteration point onto this approximate subspace. Polynomials are also applied to increase the accuracy. Intuitively, the target subspaces between any two iterations should be close enough under some conditions. We have considered two concrete examples, namely, PFPG and PFAM, in order to show the basic structure of polynomial-filtered methods. It is easy to observe that this kind of method couples the subspace refinement and the main iteration together. In the theoretical part, we analyze the convergence of PFPG and PFAM. A key assumption is that the initial subspace should not be orthogonal to the target subspace to be used in the next iteration. Together with the Chebyshev polynomials we are able to estimate the approximation error of the subspace. The main convergence result indicates that the degree of the polynomial can remain constant as the iterations proceed, which is meaningful in real applications. Even if the warm-start property is not considered, the degree grows very slowly (about order $\mathcal{O}(\log k)$) to ensure the convergence. Our numerical experiments show that the polynomial-filtered algorithms are pretty effective on low-rank problems compared to the original methods, since they successfully reduce the computational cost of large-scale EVDs. Meanwhile, the number of iterations is barely increased. These observations coincide with our theoretical results.

Acknowledgment. The authors are grateful to Prof. Stefan Guettel and two anonymous referees for their valuable comments and suggestions.

REFERENCES

[1] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York, 2011.

[2] S. Becker, V. Cevher, and A. Kyrillidis, *Randomized Low-Memory Singular Value Projection*, preprint, https://arxiv.org/abs/1303.0167, 2013.

[3] C. Davis and W. M. Kahan, *The rotation of eigenvectors by a perturbation*. III, SIAM J. Numer. Anal., 7 (1970), pp. 1–46, https://doi.org/10.1137/0707001.

[4] D. Davis and W. Yin, *Convergence rate analysis of several splitting schemes*, in Splitting Methods in Communication, Imaging, Science, and Engineering, Springer, 2016, pp. 115–163.

[5] C. Ding, D. Sun, J. Sun, and K.-C. Toh, *Spectral operators of matrices*, Math. Program., 168 (2018), pp. 509–531.

[6] J. Douglas and H. H. Rachford, *On the numerical solution of heat conduction problems in two and three space variables*, Trans. Amer. Math. Soc., 82 (1956), pp. 421–439.

[7] D. A. Flanders and G. Shortley, *Numerical determination of fundamental modes*, J. Appl. Phys., 21 (1950), pp. 1326–1332.

[8] M. P. Friedlander and I. Macêdo, *Low-rank spectral optimization via gauge duality*, SIAM J. Sci. Comput., 38 (2016), pp. A1616–A1638, https://doi.org/10.1137/15M1034283.

[9] M. X. Goemans and D. P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. ACM, 42 (1995), pp. 1115–1145.

[10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Vol. 3, Johns Hopkins University Press, 2012.

[11] F. Kangal, K. Meerbergen, E. Mengi, and W. Michiels, *A subspace method for large-scale eigenvalue optimization*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 48–82, https://doi.org/10.1137/16M1070025.

[12] D. Kressner, D. Lu, and B. Vandereycken, *Subspace acceleration for the Crawford number and related eigenvalue optimization problems*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 961–982, https://doi.org/10.1137/17M1127545.

[13] A. S. Lewis, *Derivatives of spectral functions*, Math. Oper. Res., 21 (1996), pp. 576–588.

[14] A. S. Lewis and H. S. Sendov, *Twice differentiable spectral functions*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 368–386, https://doi.org/10.1137/S089547980036838X.

[15] Y. Li, Z. Wen, C. Yang, and Y.-x. Yuan, *A semismooth Newton method for semidefinite programs and its applications in electronic structure calculations*, SIAM J. Sci. Comput., 40 (2018), pp. A4131–A4157, https://doi.org/10.1137/18M1188069.

[16] Z. Lin, M. Chen, and Y. Ma, *The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices*, J. Struct. Biol., 181 (2013), pp. 116–127.

[17] M. Nakata, H. Nakatsuji, M. Ehara, M. Fukuda, K. Nakata, and K. Fujisawa, *Variational calculations of fermion second-order reduced density matrices by semidefinite programming algorithm*, J. Chem. Phys., 114 (2001), pp. 8282–8292.

[18] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, *Conic optimization via operator splitting and homogeneous self-dual embedding*, J. Optim. Theory Appl., 169 (2016), pp. 1042–1068.

[19] N. Parikh and S. Boyd, *Proximal algorithms*, Found. Trends Optim., 1 (2014), pp. 127–239.

[20] H. Qi and D. Sun, *A quadratically convergent Newton method for computing the nearest correlation matrix*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 360–385, https://doi.org/10.1137/050624509.

[21] Y. Saad, *Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems*, Math. Comp., 42 (1984), pp. 567–588.

[22] D. Scieur, A. d'Aspremont, and F. Bach, *Regularized nonlinear acceleration*, in Advances in Neural Information Processing Systems, 2016, pp. 712–720.

[23] P. Sirković and D. Kressner, *Subspace acceleration for large-scale parameter-dependent Hermitian eigenproblems*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 695–718, https://doi.org/10.1137/15M1017181.

[24] M. Soltani and C. Hegde, *Fast Low-Rank Matrix Estimation without the Condition Number*, preprint, https://arxiv.org/abs/1712.03281, 2017.

[25] B. Vandereycken, *Low-rank matrix completion by Riemannian optimization*, SIAM J. Optim., 23 (2013), pp. 1214–1236, https://doi.org/10.1137/110845768.

[26] Z. Wen, D. Goldfarb, and W. Yin, *Alternating direction augmented Lagrangian methods for semidefinite programming*, Math. Program. Comput., 2 (2010), pp. 203–230.

[27] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, *Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization*, in Advances in Neural Information Processing Systems 22, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, eds., Curran Associates, 2009, pp. 2080–2088.

[28] X. Xiao, Y. Li, Z. Wen, and L. Zhang, *A regularized semi-smooth Newton method with projection steps for composite convex programs*, J. Sci. Comput., 76 (2018), pp. 364–389.

[29] Y. Ying and P. Li, *Distance metric learning with eigenvalue optimization*, J. Mach. Learn. Res., 13 (2012), pp. 1–26.

[30] T. Zhou and D. Tao, *GoDec: Randomized low-rank & sparse matrix decomposition in noisy case*, in Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Omnipress, 2011, pp. 33–40.

[31] Y. Zhou, Y. Saad, M. L. Tiago, and J. R. Chelikowsky, *Self-consistent-field calculations using Chebyshev-filtered subspace iteration*, J. Comput. Phys., 219 (2006), pp. 172–184.