Proximal Gradient Method for Manifold Optimization

Shixiang Chen∗ Shiqian Ma† Anthony Man-Cho So‡ Tong Zhang§

November 5, 2018

Abstract

This paper considers manifold optimization problems with nonsmooth and nonconvex objective function. Existing methods for solving this kind of problems can be classified into two classes. Algorithms in the first class rely on information of the subgradients of the objective function, which leads to slow convergence rate. Algorithms in the second class are based on operator-splitting techniques, but they usually lack rigorous convergence guarantees. In this paper, we propose a retraction-based proximal gradient method for solving this class of problems. We prove that the proposed method globally converges to a stationary point. Iteration complexity for obtaining an $\epsilon$-stationary solution is also analyzed. Numerical results on solving sparse PCA and compressed modes problems are reported to demonstrate the advantages of the proposed method.

Keywords— Manifold Optimization; Nonsmooth; Proximal Gradient Method; Iteration Complexity; Semi-smooth Newton Method; Stiefel Manifold; Sparse PCA; Compressed Modes

1 Introduction

Optimization over Riemannian manifolds has recently drawn a lot of attention due to its applications in many different fields, including low-rank matrix completion [17, 67], phase retrieval [9, 65], phase synchronization [16, 50], blind deconvolution [40], and dictionary learning [22, 64]. Manifold optimization seeks to minimize an objective function over a smooth manifold. Some widely used manifolds include sphere, Stiefel manifold, Grassmann manifold, and Hadamard manifold. The recent monograph by Absil et al. [4] studies this topic in depth. In particular, it studies several important classes of algorithms for manifold optimization with smooth objective, including line-search method, Newton’s method, and trust-region method. There are also many gradient-based algorithms for solving manifold optimization problems, including [70, 60, 61, 49, 42, 78]. However, all these methods require computing the derivatives of the objective function and do not apply to the case where the objective function is nonsmooth.

In this paper, we consider the following manifold optimization problem with nonsmooth and nonconvex objective:

$$\min F(X) := f(X) + h(X), \quad \text{s.t.,} \, X \in \mathcal{M},$$

(1.1)

where $f$ is smooth, possibly nonconvex, and its gradient $\nabla f(X)$ is Lipschitz continuous with Lipschitz constant $L$, $h$ is convex, possibly nonsmooth and Lipschitz continuous with Lipschitz constant $L_h$, $\mathcal{M}$ denotes a Riemannian manifold embedded in the Euclidean space. Here the smoothness, Lipschitz continuity and convexity are interpreted when the function in question is considered as a function in the ambient Euclidean space.

∗Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong
†Department of Mathematics, University of California, Davis
‡Department of Systems Engineering and Engineering Management, and, by courtesy, CUHK-BGI Innovation Institute of Trans-omics, The Chinese University of Hong Kong
§Tencent AI Lab
There are many interesting applications that give rise to problems of the form (1.1). In the following we briefly discuss some representative ones. For more examples of manifold optimization with nonsmooth objective, we refer the interested readers to [3].

**Example 1. Sparse Principal Component Analysis.** Principal Component Analysis (PCA), proposed by Pearson [56] and later developed by Hotelling [39], is one of the most fundamental statistical tools in analyzing high-dimensional data. Sparse PCA seeks principal components with very few nonzero components. For given data matrix $A \in \mathbb{R}^{m \times n}$, the sparse PCA that seeks the leading $r$ ($r < \min\{m, n\}$) sparse loading vectors can be formulated as

$$\min_{X \in \mathbb{R}^{n \times r}} \quad -\text{Tr}(X^\top A^\top AX) + \mu \|X\|_1$$

$$\text{s.t.} \quad X^\top X = I_r,$$  

where $\text{Tr}(Y)$ denotes the trace of matrix $Y$, the $\ell_1$ norm is defined as $\|X\|_1 = \sum_{ij} |X_{ij}|$, $\mu > 0$ is a weighting parameter and $I_r$ denotes the $r \times r$ identity matrix. Note that the constraint set $\text{St}(n, r) \equiv \{X \in \mathbb{R}^{n \times r} \mid X^\top X = I_r\}$ is known as the Stiefel manifold. This is the original formulation of sparse PCA as proposed by Jolliffe et al. in [43], where the model is called SCoTLASS, and it imposes both sparsity and orthogonality to the loading vectors simultaneously. When $\mu = 0$, (1.2) reduces to computing the leading $r$ eigenvalues and corresponding eigenvectors of $A^\top A$. When $\mu > 0$, the $\ell_1$ norm $\|X\|_1$ can promote sparsity of the loading vectors. There are many numerical algorithms for solving (1.2) when $r = 1$. In this case, (1.2) is relatively easier to solve because $X$ reduces to a vector and the constraint reduces to a sphere constraint. However, there has been very limited literature for the case $r > 1$. Existing works, including [85, 24, 62, 44, 51], do not impose orthogonal loading directions. As discussed in [44], “Simultaneously enforcing sparsity and orthogonality seems to be a hard (and perhaps questionable) task.” We refer [86] to the interested readers for more details on existing algorithms for solving sparse PCA. As we will discuss later, our algorithm can solve (1.2) for $r > 1$ efficiently, which imposes sparsity and orthogonality simultaneously.

**Example 2. Compressed Modes in Physics.** This problem seeks spatially localized (“sparse”) solutions of the independent-particle Schrödinger’s equation. Sparsity is achieved by adding an $L_1$ regularization of the wave functions, which leads to solutions with compact support (“compressed modes”). For 1D free-electron case, after proper discretization, this problem can be formulated as

$$\min_{X \in \mathbb{R}^{n \times r}} \quad -\text{Tr}(X^\top H X) + \mu \|X\|_1$$

$$\text{s.t.} \quad X^\top X = I_r,$$  

where $H$ denotes the discretized Schrödinger operator. Note that the $L_1$ regularization becomes $\ell_1$ norm of $X$ after discretization. We refer to [55] for more details of this problem. Note that (1.2) and (1.3) are different in the way that $H$ and $A^\top A$ have totally different structures.

**Example 3. Unsupervised Feature Selection.** It is much more difficult to select the discriminative features in unsupervised learning due to the lack of label information than supervised learning. There are some recent works that model this task as a manifold optimization problem in the form of (1.1). For instance, [76] and [66] suggest the following model:

$$\min_{W \in \mathbb{R}^{n \times r}} \quad -\text{Tr}(W^\top MW) + \mu \|W\|_{2,1}$$

$$\text{s.t.} \quad W^\top W = I_r,$$  

where $M$ is a given matrix computed from the input data, $W$ denotes a linear classifier that classifies each data point to a class, and the $\ell_{2,1}$ norm is defined as $\|W\|_{2,1} = \sum_{i=1}^n \|W(i,:)\|_2$ with $W(i,:)$ being the $i$-th row of $W$. We refer to [76] and [66] for more details.

**Example 4. Robust PCA.** This problems seeks decomposing a given matrix $M \in \mathbb{R}^{m \times n}$ into the superposition of a low-rank matrix $L$ and a sparse matrix $S$. If the rank of $L$ is known to be $r$ ($r < \min\{m, n\}$), then we can factorize $L = UV^\top$, where $U \in \mathbb{R}^{m \times r}$, $V \in \mathbb{R}^{n \times r}$. As a result, robust PCA can be formulated as the following optimization problem:

$$\min_{U,V,S} \quad \|UV^\top + S - M\|_F + \mu \|S\|_1$$

$$\text{s.t.} \quad U \in \mathcal{G}(m, r),$$  

where $\mathcal{G}(m, r)$ denotes the Grassmann manifold. Note that a similar formulation for matrix completion is given in [17]. Problem (1.5) is of the form (1.1) with $M$ being the Cartesian product of $\mathcal{G}(m, r)$, $\mathbb{R}^{n \times r}$ and $\mathbb{R}^{m \times n}$.
Example 5. Sparse Blind Deconvolution. Given the observations
\[ y = a_0 \otimes x_0 \in \mathbb{R}^m, \]
how to recover both the convolution kernel \( a_0 \in \mathbb{R}^k \) and signal \( x_0 \in \mathbb{R}^m \)? Here \( x_0 \) is assumed to have a sparse and random support. This problem is known as sparse blind deconvolution. Some recent works on this topic suggest the following optimization formulation to recover \( a_0 \) and sparse \( x_0 \) (see, e.g., \([81]\)):
\[
\min_{a,x} \quad \|y - a \otimes x\|_2^2 + \mu \|x\|_1 \\
\text{s.t.} \quad \|a\|_2 = 1,
\]
which is of the form (1.1) with \( \mathcal{M} \) being the Cartesian product of a sphere and \( \mathbb{R}^m \).

Example 6. Nonconvex Regularizer. Problem (1.1) also allows nonconvex regularizer function. For example, instead of using \( \ell_1 \) norm to promote the sparsity, we can use the MCP (minimax concave penalty) function \([77]\), which has been widely used in statistics. The MCP function is nonconvex and given as follows:
\[
P(x) = \begin{cases} 
\lambda |x| - \frac{x^2}{2\lambda}, & \text{if } |x| \leq \gamma \lambda \\
\frac{1}{2\gamma} |x|^2, & \text{otherwise},
\end{cases}
\]
where \( \lambda \) and \( \gamma \) are pre-given parameters, and \( x \in \mathbb{R} \). If we replace the \( \ell_1 \) norm in sparse PCA (1.2) by MCP, it reduces to:
\[
\min_{X \in \mathbb{R}^{m \times r}} \quad -\text{Tr}(X^T A^T A X) + \mu \sum_{ij} P(X_{ij}) \\
\text{s.t.} \quad X^T X = I_r.
\]
It is easy to see that the objective function of (1.8) can be rewritten as \( f_1(X) + f_2(X) \), with \( f_1(X) = -\text{Tr}(X^T A^T A X) + \mu(\sum_{ij} P(X_{ij}) - \lambda \|X\|_1) \) and \( f_2(X) = \mu \|X\|_1 \). Note that \( f_1(X) \) is smooth and its gradient is Lipschitz continuous. Therefore, (1.8) also takes the form (1.1) and can be solved by our algorithm.

Our Contributions. Due to the needs of the above mentioned applications, it is highly desirable to design an efficient algorithm for solving (1.1). In this paper, we propose a proximal gradient method for solving it. The proposed method, named ManPG (Manifold Proximal Gradient Method), is based on the proximal gradient method with a retraction operation to keep the iterates feasible with respect to the manifold constraint. Each step of ManPG involves solving a well-structured convex optimization problem, which can be done efficiently by the semi-smooth Newton method. We prove that ManPG converges to a stationary point of (1.1) globally. We also analyze the iteration complexity of ManPG for obtaining an \( \epsilon \)-stationary point. Numerical results on sparse PCA and compressed modes problems show that our ManPG algorithm compares favorably with existing methods.

Notation. The following notation is adopted throughout this paper. The tangent space to \( \mathcal{M} \) at point \( X \) is denoted by \( T_X \mathcal{M} \). We use \( \langle A, B \rangle = \text{Tr}(A^T B) \) to denote the Euclidean inner product of two matrices \( A, B \), and we use the Euclidean inner product as the metric on the tangent space. We use \( \|X\|_F \) to denote the Frobenius norm of \( X \). The Euclidean gradient of a smooth function \( f(X) \) is denoted as \( \nabla f(X) \) and the Riemannian gradient of \( f(X) \) is denoted as \( \text{grad} f(X) \). Note that \( \text{grad} f(X) = \text{Proj}_{T_X \mathcal{M}} \nabla f(X) \), the orthogonal projection of \( \nabla f(X) \) onto the tangent space. For a convex function \( h(X) \), its Euclidean subgradient is denoted by \( \partial h(X) \). We use \( \text{vec}(X) \) to denote the vector formed by stacking the column vectors of \( X \). For symmetric matrix \( X \in \mathbb{R}^{r \times r} \), \( \text{vec}(X) \) denotes the \( \frac{1}{2} r(r + 1) \)-dimensional vector obtained from \( \text{vec}(X) \) by eliminating all super-diagonal elements of \( X \). We denote \( Z \succeq 0 \) if \( (Z + Z^T)/2 \) is positive semidefinite. The proximal mapping of \( h \) at point \( X \) is defined by \( \text{prox}_h(X) = \text{argmin}_{Z} \frac{1}{2} \|Y - X\|_F^2 + h(Y) \).

Organization. The rest of this paper is organized as follows. In Section 2 we briefly review existing works on solving manifold optimization problems with nonsmooth objective functions. We introduce some preliminaries of manifolds in Section 3. The main algorithm, ManPG, is presented in Section 4 and the semi-smooth Newton method for solving the subproblem is discussed in Section 4.2. In Section 5, we establish the global convergence of ManPG and analyze its iteration complexity for obtaining an \( \epsilon \)-stationary solution. Numerical results of ManPG on solving compressed modes problems in physics and sparse PCA are reported in Section 6. Finally, we draw some conclusions in Section 7.
2 Nonsmooth Optimization over Riemannian Manifold

Unlike manifold optimization with smooth objective, which has been studied extensively in the monograph [4], the literature on manifold optimization with nonsmooth objective has been relatively limited. Numerical algorithms for solving manifold optimization with nonsmooth objective can be roughly classified into two categories. Algorithms in the first category include the ones proposed in [36, 38, 8, 25, 15, 35] and they all rely on subgradient information. Specifically, Dirr et al. [25] and Borckmans et al. [15] proposed subgradient methods on manifold. Hosseini [36] used the Kurdyka-Łojasiewicz (KL) inequality to establish the global convergence of some subgradient-oriented descent methods. Hosseini and Ushmajew [38] proposed a Riemannian gradient sampling algorithm. Grohs and Hosseini [35] generalized a nonsmooth trust region method to manifold optimization, which also relies on subgradient information to construct a particular function used in the algorithm. Proximal point algorithms for solving manifold optimization are also studied in the literature. Bacak et al. [8] proposed a cyclic proximal point algorithm on Hadamard manifold. Bento et al. [11] proposed a proximal point algorithm that can solve some more general manifold optimization problems and proved its convergence by assuming the KL inequality. Bento et al. [12] studied the iteration complexity of proximal point methods for manifold optimization under the assumption that the objective function is convex. In [10], Bento et al. analyzed convergence of some inexact descent methods based on the KL inequality, including the proximal point method and steepest descent method. One possible issue of these works is that the subproblem, which requires minimizing the original objective function plus a proximal term over manifold, can be difficult to solve. As a result, it is still suggested to use the subgradient method to solve the subproblems (see, e.g., [8]). Since subgradient algorithm is known to be slower than the gradient algorithm and proximal gradient algorithm in Euclidean space, it is expected that these subgradient-based algorithms are not as efficient as gradient algorithms and proximal gradient algorithms on manifold in practice. Moreover, the proximal point methods discussed above are rather abstract and it is not clear whether they perform well numerically in practice.

Algorithms in the second category explore more structures of problem (1.1) and do not need subgradient information. These works mainly focus on problems over the Stiefel manifold. Note that (1.1) is challenging because of the combination of two difficult terms: Riemannian manifold and nonsmooth objective. If only one of them is present, then it is relatively easier to solve. Therefore, the alternating direction method of multipliers (ADMM) becomes a natural choice for solving (1.1). ADMM for solving convex optimization problems with two-block variables is closely related to the famous Douglas-Rachford operator splitting method that has a long history [32, 29, 48, 28, 31, 26]. The renaissance of ADMM was initiated by several papers around 2007-2008, where it was successfully applied to solving signal processing problems [23] and image processing problems [73, 33, 6]. The recent survey paper [20] popularized this method in many areas. Recently, there have been some emerging interests in ADMM for solving manifold optimization of the form (1.1) (see, e.g., [46, 45, 69, 80]). However, the algorithms presented in these papers either lack convergence guarantee (46, 45) or their convergence needs further conditions that do not apply to (1.1) (69, 80).

Here we briefly describe the SOC method (Splitting method for Orthogonality Constrained problems) presented in [46]. The SOC method aims to solve

$$\min J(X), \text{s.t.}, X \in \mathcal{M}. \tag{2.1}$$

To solve this problem, SOC introduces an auxiliary variable $P$ and considers the following reformulation:

$$\min J(P), \text{s.t.}, P = X, X \in \mathcal{M}. \tag{2.2}$$

By associating a Lagrange multiplier $\Lambda$ to the linear equality constraint, the augmented Lagrangian function of (2.2) can be written as

$$\mathcal{L}_\beta(X, P; \Lambda) := J(P) - \langle \Lambda, P - X \rangle + \frac{\beta}{2} ||P - X||^2_P,$$

where $\beta > 0$ is a penalty parameter. The SOC algorithm iterates as follows:

$$\begin{align*}
P^{k+1} &:= \arg\min_P \mathcal{L}_\beta(P, X^k; \Lambda^k), \\
X^{k+1} &:= \arg\min_X \mathcal{L}_\beta(P^{k+1}, X; \Lambda^k), \text{s.t.}, X \in \mathcal{M}, \\
\Lambda^{k+1} &:= \Lambda^k - \beta(P^{k+1} - X). \tag{2.3}
\end{align*}$$
Note that the $X$-subproblem corresponds to the projection onto $\mathcal{M}$, and the $P$-subproblem is an unconstrained problem and hence can be solved relatively easily depending on the structure of $J$. In particular, if $J$ is smooth, then the $P$-subproblem can be solved iteratively by the gradient method; if $J$ is nonsmooth and has easy proximal mapping, then the $P$-subproblem can be solved directly by computing the proximal mapping of $J$.

The MADMM (manifold ADMM) algorithm presented in [45] aims to solve the following problem:

$$\min_{X,Z} f(X) + g(Z), \text{ s.t., } Z = AX, \ X \in \text{St}(n,r),$$

(2.4)

where $f$ is smooth and $g$ is nonsmooth with easily computable proximal mapping. The augmented Lagrangian function of (2.4) is

$$\mathcal{L}_\beta(X, Z; \Lambda) := f(X) + g(Z) - \langle \Lambda, Z - AX \rangle + \frac{\beta}{2} \|Z - AX\|_F^2.$$

The MADMM algorithm (Manifold ADMM) proposed in [45] iterates as follows:

$$X^{k+1} := \arg\min_X \mathcal{L}_\beta(X, Z^k; \Lambda^k), \text{ s.t., } X \in \text{St}(n,r),$$

$$Z^{k+1} := \arg\min_Z \mathcal{L}_\beta(X^{k+1}, Z; \Lambda^k),$$

$$\Lambda^{k+1} := \Lambda^k - \beta(Z^{k+1} - AX^{k+1}).$$

(2.5)

Note that the $X$-subproblem is a smooth optimization problem on the Stiefel manifold, and the authors suggested to use the Manopt toolbox [19] to solve it. The $Z$-subproblem corresponds to the proximal mapping of function $g$.

As far as we know, however, the convergence guarantees of SOC and MADMM are still missing in the literature. Though there are some recent works that analyze the convergence of ADMM for nonconvex problems [69, 80], their results need further conditions that do not apply to (1.1) and its reformulations (2.2) and (2.4).

More recently, some other variants of the augmented Lagrangian method are proposed to deal with (1.1). In [21], Chen et al. proposed a PAMAL method which hybridizes an augmented Lagrangian method with proximal alternating minimization. More specifically, PAMAL solves the following reformulation of (1.1):

$$\min_{X,Q,P} f(P) + h(Q), \text{ s.t., } Q = X, P = X, X \in \text{St}(n,r).$$

(2.6)

By associating Lagrange multipliers $\Lambda_1$ and $\Lambda_2$ to the two linear equality constraints, the augmented Lagrangian function of (2.6) can be written as

$$\mathcal{L}_\beta(X, Q, P; \Lambda_1, \Lambda_2) := f(P) + h(Q) - \langle \Lambda_1, Q - X \rangle - \langle \Lambda_2, P - X \rangle + \frac{\beta}{2} \|Q - X\|_F^2 + \frac{\beta}{2} \|P - X\|_F^2,$$

where $\beta > 0$ is a penalty parameter. The augmented Lagrangian method for solving (2.6) iterates as

$$(X^{k+1}, Q^{k+1}, P^{k+1}) := \arg\min_{X,Q,P} \mathcal{L}_\beta(X, Q, P; \Lambda_1^k, \Lambda_2^k), \text{ s.t., } X \in \text{St}(n,r),$$

$$\Lambda_1^{k+1} := \Lambda_1^k - \beta(Q^{k+1} - X^{k+1}),$$

$$\Lambda_2^{k+1} := \Lambda_2^k - \beta(P^{k+1} - X^{k+1}).$$

(2.7)

Note that the subproblem in (2.7) is still difficult to solve. Therefore, the authors of [21] suggested to use the proximal alternating minimization method to solve the subproblem in (2.7) inexactly. They named the augmented Lagrangian method (2.7) with the subproblems being solved by proximal alternating minimization method as PAMAL. They proved that under certain conditions, any limit point of the sequence generated by PAMAL is a KKT point of (2.6). It needs to be pointed out that the proximal alternating minimization procedure involves many parameters that need to be tuned in order to solve the subproblem inexactly. Our numerical results in Section 6 indicate that the performance of PAMAL significantly depends on the setting of these parameters.

In [84], Zhu et al. studied another algorithm called EPALMAL for solving (1.1) that is based on the augmented Lagrangian method and the PALM algorithm [14]. The difference between EPALMAL and PAMAL
is that they use different algorithms to minimize the augmented Lagrangian function inexactly. In particular, EPALMAL uses PALM algorithm, and PAMAL uses proximal alternating minimization algorithm. It is also shown in [84] that any limit point of the sequence generated by EPALMAL is a KKT point. However, their result assumes that the iterate sequence is bounded, which holds automatically if the manifold in question is bounded but is hard to verify otherwise.

3 Preliminaries on Manifold Optimization

We first introduce some preliminaries of Riemannian manifold and problem (1.1), which will be useful for developing the first-order necessary optimality conditions of (1.1). A function $F$ is said to be locally Lipschitz continuous if for any $X \in \mathcal{M}$, it is Lipschitz continuous in a neighborhood of $X$. Note that if $F$ is locally Lipschitz continuous in Euclidean space, then it is also locally Lipschitz continuous when restricted to the embedded submanifold $\mathcal{M}$.

Definition 3.1. (Generalized Clarke subdifferential [37]) For a locally Lipschitz function $F$ on $\mathcal{M}$, the Riemannian generalized directional derivative of $F$ at $X \in \mathcal{M}$ in direction $V$ is defined by

$$F^\circ(X, D) = \limsup_{Y \to X, t \downarrow 0} \frac{F \circ \phi^{-1}(\phi(Y) + tD\phi(X)[V]) - f \circ \phi^{-1}(\phi(Y))}{t},$$

(3.1)

where $(\phi, U)$ is a coordinate chart at $X$. The generalized gradient or the Clarke subdifferential of $F$ at $X \in \mathcal{M}$, denoted by $\partial F(X)$, is given by

$$\partial F(X) = \{\xi \in T_X \mathcal{M} : \langle \xi, V \rangle \leq F^\circ(X, V), \forall V \in T_X \mathcal{M}\}.$$ (3.2)

Definition 3.2. ([75]) A function $f$ is said to be regular at $X \in \mathcal{M}$ along $T_X \mathcal{M}$ if

- for all $V \in T_X \mathcal{M}$, $f'(X; V) = \lim_{t \downarrow 0} \frac{f(X+tV)-f(X)}{t}$ exists, and
- for all $V \in T_X \mathcal{M}$, $f'(X; V) = f^\circ(X; V)$.

For smooth function $f$, we know that $\nabla f(X) = \operatorname{Proj}_{T_X \mathcal{M}} \nabla f(X)$. According to Theorem 5.1 in [75], for a regular function $F$, we have $\partial F(X) = \operatorname{Proj}_{T_X \mathcal{M}} (\partial F(X))$. Moreover, the function $F(X) = f(X) + h(X)$ in problem (1.1) is regular according to Lemma 5.1 in [75]. Therefore, we have $\partial F(X) = \operatorname{Proj}_{T_X \mathcal{M}} (\nabla f(X) + \partial h(X)) = \nabla f(X) + \operatorname{Proj}_{T_X \mathcal{M}} (\partial h(X))$. By Theorem 4.1 in [75], the first-order necessary condition of problem (1.1) is given by $0 \in \nabla f(X) + \operatorname{Proj}_{T_X \mathcal{M}} (\partial h(X))$.

Definition 3.3. A point $X \in \mathcal{M}$ is called a stationary point of problem (1.1) if it satisfies the first-order necessary condition, i.e., $0 \in \nabla f(X) + \operatorname{Proj}_{T_X \mathcal{M}} (\partial h(X))$.

A very important concept in manifold optimization is the retraction operation, which is defined as follows.

Definition 3.4. ([4, Definition 4.1.1]) A retraction on a differentiable manifold $\mathcal{M}$ is a smooth mapping $\operatorname{Retr}_X$ from the tangent bundle $T_X \mathcal{M}$ onto $\mathcal{M}$ satisfying the following two conditions, where $\operatorname{Retr}_X$ denotes the restriction of $\operatorname{Retr}$ onto $T_X \mathcal{M}$.

1. $\operatorname{Retr}_X(0) = X, \forall X \in \mathcal{M}$, where $0$ denotes the zero element of $T_X \mathcal{M}$.

2. For any $X \in \mathcal{M}$, it holds that

$$\lim_{T_X \mathcal{M} \ni \xi \to 0} \frac{\|\operatorname{Retr}_X(\xi) - (X + \xi)\|_F}{\|\xi\|_F} = 0.$$

Remark 3.5. The second condition ensures that $\operatorname{Retr}_X(\xi) = X + \xi + o(\|\xi\|_F)$ and $D\operatorname{Retr}_X(0) = \operatorname{Id}$, where $D\operatorname{Retr}_X$ is the differential of retraction and $\operatorname{Id}$ denotes the identity mapping. For more details about retraction, we refer to [4, 18] and references therein.
The retraction onto the Euclidean space is simply the identity mapping, i.e., $\text{Retr}_X(\xi) = X + \xi$. For the Stiefel manifold $\text{St}(n, r)$, common retractions include the exponential mapping \cite{27}

$$\text{Retr}_X^{\text{exp}}(t\xi) = [X, Q] \exp\left( t \begin{bmatrix} -X^T\xi & -R^T \\ R & 0 \end{bmatrix} \right) I_r,$$

where $QR = -(I_n - XX^T)\xi$ is the unique QR factorization; the polar decomposition

$$\text{Retr}_X^{\text{polar}}(\xi) = (X + \xi)(I_r + \xi^T\xi)^{-1/2};$$

the QR decomposition

$$\text{Retr}_X^{\text{QR}}(\xi) = qf(X + \xi),$$

where $qf(A)$ is the $Q$ factor of the QR factorization of $A$, and the Cayley transformation

$$\text{Retr}_X^{\text{cayley}}(\xi) = \left( I_n - \frac{1}{2} W(\xi) \right)^{-1} \left( I_n + \frac{1}{2} W(\xi) \right) X,$$

where $W(\xi) = (I_n - \frac{1}{2} XX^T)\xi X^T - X^T(I_n - \frac{1}{2} XX^T)$.

For a given matrix $Y \in \mathbb{R}^{n \times r}$, $r \leq n$, its orthogonal projection onto the Stiefel manifold is given by $UI_{r}V^T$, where $U, V$ are the truncated left and right singular vectors of $Y$. If $Y$ has full rank, then the projection can be computed by $Y(Y^TY)^{-1/2}$, which is the same as polar decomposition. The total computational cost of projection by $UI_{r}V^T$ is $8nr^2 + O(r^3)$ flops, where the SVD needs $6nr^2 + O(r^3)$ flops \cite{34} and forming $UI_{r}V^T$ needs $2nr^2$ flops. In comparison, if $Y = X + \xi$ and $\xi \in T_X\mathcal{M}$, the exponential mapping takes $8nr^2 + O(r^3)$ flops, and the polar decomposition above has $3nr^2 + O(r^3)$ flops, where $\xi^T\xi$ needs $nr^2$ flops and the remaining $2nr^2 + O(r^3)$ comes from the final assembly. Thus, polar decomposition is cheaper than the projection. Moreover, the QR decomposition of $X + \xi$ takes $2nr^2 + O(r^3)$ flops, but the MATLAB implementation of QR is slower than the polar decomposition \cite{5}. For the Cayley transformation of $X + \xi$, the total cost is $7nr^2 + O(r^3)$ \cite{70, 41}. In our algorithm that will be introduced later, we need to perform one retraction operation in each iteration. We need to point out that retractions may also affect the overall convergence speed of the algorithm. As a result, determining the most efficient retraction used in the algorithm is still an interesting question to investigate in practice; see also the discussion after Theorem 3 of \cite{49}.

The retraction Retr has the following properties that are useful for our convergence analysis.

\textbf{Lemma 3.6.} Let $\overline{\mathcal{M}} := \{ X \in \mathcal{M} \mid \| X \|_F \leq \psi \}$ be a compact subset of $\mathcal{M}$ and $K := \{ \xi \in T\mathcal{M} \mid \| \xi \|_F \leq \phi \}$ be a bounded subset of the tangent bundle $T\mathcal{M}$, where $\psi, \phi > 0$ are given scalars. For all $X \in \overline{\mathcal{M}}$ and $\xi \in K$, there exist constants $M_1 > 0$ and $M_2 > 0$ such that the following two inequalities hold:

\begin{align*}
\| \text{Retr}_X(\xi) - X \|_F &\leq M_1 \| \xi \|_F, \forall X \in \overline{\mathcal{M}}, \xi \in K, \quad (3.3) \\
\| \text{Retr}_X(\xi) - (X + \xi) \|_F &\leq M_2 \| \xi \|_F, \forall X \in \overline{\mathcal{M}}, \xi \in K. \quad (3.4)
\end{align*}

\textbf{Proof.} From Definition 3.4, we know that the retraction is smooth on $K := \{ \xi \in T\mathcal{M} \mid \| \xi \|_F \leq \phi \}$. For all $\xi \in K$, we have

$$\| \text{Retr}_X(\xi) - X \|_F = \left\| \int_0^1 \frac{d}{dt} \text{Retr}_X(t\xi) dt \right\| \leq \int_0^1 \left\| \frac{d}{dt} \text{Retr}_X(t\xi) \right\| dt = \int_0^1 \| D\text{Retr}_X(t\xi) \|_F dt \leq \int_0^1 \max_{\xi \in \mathcal{M}, \xi \in K} \| D\text{Retr}_X(\xi) \|_F \| \xi \|_F dt = \max_{\xi \in \mathcal{M}, \xi \in K} \| D\text{Retr}_X(\xi) \|_F \| \xi \|_F,$$

where we use $D\text{Retr}_X(0) = \text{Id}$ in the first equality and the norm $\| \cdot \|_F$ denotes the operator norm. Now (3.3) is proved with $M_1 = \max_{\xi \in \mathcal{M}, \xi \in K} \| D\text{Retr}_X(\xi) \|_F$. 

7
We now prove inequality (3.4). For all $\xi \in K$, we have

$$\|\text{Retr}_X(\xi) - X - \xi\|_F \leq \int_0^1 \left\| \frac{d}{dt}(\text{Retr}_X(t\xi) - X - t\xi) \right\| dt = \int_0^1 \|D\text{Retr}_X(t\xi)[\xi] - \xi\|_F dt$$

$$\leq \int_0^1 \|D\text{Retr}_X(t\xi) - \Id\|\|\xi\|_F dt \leq \int_0^1 \int_0^1 \left\| \frac{d}{ds}D\text{Retr}_X(s\xi) \right\| \|\xi\|_F ds dt$$

$$\leq \max_{x \in M, \xi \in K} \frac{1}{2}\|D^2\text{Retr}_X(\xi)\|\|\xi\|_F^2.$$ 

Thus (3.4) holds with $M_2 = \max_{x \in M, \xi \in K} \frac{1}{2}\|D^2\text{Retr}_X(\xi)\|_F$. \hfill \Box

**Remark 3.7.** In [18], Boumal et al. proved that (3.3) and (3.4) hold for compact manifold. In this paper, we do not require the manifold to be compact; we only require that (3.3) and (3.4) hold for compact subset of the manifold.

### 4 Proximal Gradient Method on Riemannian Manifold

#### 4.1 The ManPG Algorithm

For manifold optimization problems with a smooth objective, Riemannian gradient descent (RGD) method [1, 4, 54] has been one of the methods of choice. A generic updating formula of RGD for solving

$$\min_X F(X), \text{ s.t., } X \in M$$

is

$$X_{k+1} = X_k - \text{Retr}_{X_k}(\alpha_k D_k), \quad (4.1)$$

where $D_k$ is a descent direction of $F$ in the tangent space $T_{X_k} M$, and $\alpha_k$ is a step size. Recently, Boumal et al. [18] have analyzed the sublinear convergence rate of RGD for achieving points with the norm of Riemannian gradient smaller than $\epsilon$, i.e., $\|\text{grad}F(X_k)\|_F < \epsilon$. Liu et al. [49] proved the linear convergence rate of RGD for quadratic minimization over Stiefel manifold. Other methods for solving manifold optimization with smooth objective have also been studied in the literature, including conjugate gradient methods [4, 2], trust region methods [4, 18], and Newton-type methods [4, 59].

We now propose our ManPG algorithm for solving manifold optimization with nonsmooth objective (1.1). ManPG first computes a descent direction $D_k$ by solving the following subproblem:

$$\min_D \left\langle \text{grad}f(X_k), D \right\rangle + \frac{1}{2t} \|D\|_F^2 + h(X_k + D) \quad \text{s.t. } \quad D \in T_{X_k} M, \quad (4.2)$$

where $t > 0$ is a step size, $\text{grad}f$ is the Riemannian gradient. Note that $D_k$ can be viewed as a Riemannian proximal gradient.

Following the definition of $\text{grad}f$, we have

$$\left\langle \text{grad}f(X_k), D \right\rangle = \langle \nabla f(X_k), D \rangle, \quad \forall D \in T_{X_k} M, \quad (4.3)$$

which implies that (4.2) can be rewritten as

$$\min_D \langle \nabla f(X_k), D \rangle + \frac{1}{2t} \|D\|_F^2 + h(X_k + D) \quad \text{s.t. } \quad D \in T_{X_k} M. \quad (4.4)$$

As a result, we do not need to compute the Riemannian gradient $\text{grad}f$, and only the Euclidean gradient $\nabla f$ is needed. Note that without considering the constraint $D \in T_{X_k} M$, (4.4) computes a proximal gradient step. Therefore, (4.4) can be viewed as a proximal gradient step restricted to the tangent space $T_{X_k} M$. Since $X_k + \alpha_k D_k$ does not necessarily lie on the manifold $M$, we perform a retraction to bring it back to $M$.

Our ManPG algorithm for solving (1.1) is described in Algorithm 1. Note that ManPG involves an Amijo line search procedure to determine the step size $\alpha$. As we will show in Section 5, this backtracking line search procedure is well defined, i.e., it will terminate after finite number of steps.
Algorithm 1 Manifold Proximal Gradient Method (ManPG) for Solving (1.1)

1: Input: Initial point $X_0 \in \mathcal{M}$, $\delta \in (0, 1)$, $\gamma \in (0, 1)$, Lipschitz constant $L$
2: for $k = 0, 1, \ldots$ do
3: Obtain $D_k$ by solving the subproblem (4.4) with $t \in (0, 1/L]$;
4: Set $\alpha = 1$
5: while $F(\text{Retr}_{X_k}(\alpha D_k)) > F(X_k) - \delta \alpha \|D_k\|_F^2$ do
6: $\alpha = \gamma \alpha$
7: end while
8: Set $X_{k+1} = \text{Retr}_{X_k}(\alpha D_k)$
9: end for

4.2 Regularized Semi-Smooth Newton Method for Subproblem (4.4)

The main computational effort of Algorithm 1 lies in solving the convex subproblem (4.4). We conducted extensive numerical experiments and found that the semi-smooth Newton method (SSN) is very efficient for solving this problem. The notion of semi-smoothness was originally introduced by Mifflin [53] for real-valued functions and extended to vector-valued mappings by Qi and Sun [58]. A pioneer work on the SSN method was due to Solodov and Svaiter [63], where the authors proposed a globally convergent Newton method by exploiting the structure of monotonicity, and local superlinear rate was established under the conditions that the function is differentiable and its generalized Jacobian is non-singular. The convergence rate is extended in [83] to the setting where the generalized Jacobian is not necessarily non-singular. Recently, SSN has received significant attentions due to its success in solving structured convex problems to a high accuracy. In particular, it has been successfully applied to solving SDP [82, 74], Lasso [47], nearest correlation matrix estimation [57], clustering [68], sparse inverse covariance selection [72], and composite convex minimization [71].

In the following we show how to apply the SSN method to solve the subproblem (4.4). For the ease of presentation, without loss of generality, we assume that there exists a linear operator $A_k$ such that the constraint $D \in T_{X_k} \mathcal{M}$ can be equivalently represented as $A_k(D) = 0$. That is, (4.4) reduces to

$$
\min_D \langle \nabla f(X_k), D \rangle + \frac{1}{2t} \|D\|_F^2 + h(X_k + D) \\
\text{s.t. } A_k(D) = 0.
$$

(4.5)

For instance, when $\mathcal{M}$ is the Stiefel manifold, $A_k(D) = D^T X_k + X_k^T D$. By associating a Lagrange multiplier $\Lambda$ to the linear equality constraint, the Lagrangian function of (4.5) can be written as

$$
L(D; \Lambda) = \langle \nabla f(X_k), D \rangle + \frac{1}{2t} \|D\|_F^2 + h(X_k + D) - \langle A_k(D), \Lambda \rangle,
$$

(4.6)

and the KKT system of (4.5) is given by

$$
0 \in \partial_D L(D; \Lambda), \quad A_k(D) = 0.
$$

(4.7)

The first condition in (4.7) implies that $D$ can be computed by

$$
D(\Lambda) = \text{prox}_{th}(B(\Lambda)) - X_k, \quad \text{with } B(\Lambda) = X_k - t(\nabla f(X_k) - A_k^*(\Lambda)),
$$

(4.8)

where $A_k^*$ denotes the adjoint operator of $A_k$. By substituting (4.8) into the second condition in (4.7), we get that $\Lambda$ satisfies

$$
E(\Lambda) = A_k(D(\Lambda)) = 0,
$$

(4.9)

with $D(\Lambda)$ given in (4.8). We will use SSN to solve (4.9). To do so, we need to first show that the operator
$E$ is monotone. For any $\Lambda_1, \Lambda_2 \in S^{r \times r}$, we have
\[
\begin{align*}
\langle E(\Lambda_1) - E(\Lambda_2), \Lambda_1 - \Lambda_2 \rangle \\
= \langle D(\Lambda_1) - D(\Lambda_2), A_k^*(\Lambda_1 - \Lambda_2) \rangle \\
= \frac{1}{t} \langle \text{prox}_{th}(B(\Lambda_1)) - \text{prox}_{th}(B(\Lambda_2)), B(\Lambda_1) - B(\Lambda_2) \rangle \\
\geq \frac{1}{t} \| \text{prox}_{th}(B(\Lambda_1)) - \text{prox}_{th}(B(\Lambda_2)) \|^2_F \geq 0,
\end{align*}
\]
where the first inequality holds since the proximal mapping is firmly non-expansive. Therefore, $E$ is indeed monotone and we can use SSN to solve the nonlinear monotone equation (4.9) to get $\Lambda$ and then compute $D$ via (4.8).

In order to apply SSN, we need to compute the generalized Jacobian of $E$. In the following we show how to compute it. By vectorizing $E(\Lambda)$, we have
\[
\text{vec}(E(\Lambda)) = \text{vec}[A_k(D(\Lambda))] = A_k \text{vec}(D(\Lambda))
= A_k \left[ \text{prox}_{th}(\text{vec}(X_k - t\nabla f(X_k)) + tA_k^T \text{vec}(\Lambda)) - \text{vec}(X_k) \right],
\]
where $A_k$ denotes the corresponding matrix of $A_k$ under the standard basis after vectorizing. By the chain rule, the generalized Jacobian of $\text{vec}(E(\Lambda))$ is given by
\[
G(\text{vec}(\Lambda)) = tA_k \mathcal{J}(y)|_{y=\text{vec}(B(\Lambda))} A_k^T,
\]
where $\mathcal{J}(y)$ is the generalized Jacobian of $\text{prox}_{th}(\cdot)(y)$, which is a diagonal matrix. For example, if $h(X) = \mu \|X\|_1$, then $\mathcal{J}$ is given by
\[
\mathcal{J}(y)_{ii} = \begin{cases} 1, & \text{if } |y_i| > \mu t, \\ 0, & \text{otherwise}. \end{cases} \tag{4.10}
\]

It should be pointed out that $G(\text{vec}(\Lambda))$ can be singular. Therefore, the vanilla SSN cannot be applied directly and we need to resort to a regularized SSN proposed in [63] and further studied in [83, 71]. We found that the adaptive regularized SSN (ASSN) proposed in [71] is very suitable for solving (4.9). ASSN first computes the Newton’s direction $d_k$ by solving
\[
(G(\text{vec}(\Lambda_k)) + \eta I) d = -\text{vec}(E(\Lambda_k)) \tag{4.11}
\]
with a regularization parameter $\eta$. If the matrix size is large, then the equation (4.11) can be solved inexactly by the conjugate gradient method. The authors then designed a strategy to decide whether to accept this $d_k$ or not. Roughly speaking, if $\| E(\Lambda_{k+1}) \|_2$ is sufficiently decreased from $\| E(\Lambda_k) \|_2$, then we accept $d^k$, where
\[
\text{vec}(\Lambda_{k+1}) = \text{vec}(\Lambda_k) + d_k.
\]
Otherwise, a safeguard step is taken. For more details on ASSN, we refer to [71].

5 Global Convergence and Iteration Complexity

In this section, we analyze the convergence and iteration complexity of our ManPG algorithm (Algorithm 1) for solving (1.1). We first make the following assumption on the objective function $F$.

**Assumption 5.1.** The objective function $F$ of (1.1) satisfies the following properties.

- $F(X)$ is coercive, i.e. $F(X) \to +\infty$ when $\|X\| \to \infty$. An immediate consequence of this is that the sub-level set $\{X \mid F(X) \leq F(X_0)\}$ is bounded.
- $F(X)$ is lower bounded on $\mathcal{M}$, i.e., there exists a constant $F_*$ such that $F(X) \geq F_*$ for all $X \in \mathcal{M}$.

The following lemma shows that $D_k$ obtained by solving (4.5) is indeed a descent direction in the tangent space.
Lemma 5.2. Let $D_k$ be the minimizer of problem (4.5). The following holds for any $\alpha \in [0,1)$:

$$f(X_k + \alpha D_k) + h(X_k + \alpha D_k) \leq f(X_k) + h(X_k) - \frac{\alpha}{2t} \|D_k\|_F^2. \quad (5.1)$$

Proof. Since $D_k$ is the minimizer of (4.5), it holds that for any $\alpha \in [0,1)$,

$$\langle \nabla f(X_k), \alpha D_k \rangle + \frac{1}{2t} \|\alpha D_k\|_F^2 + h(X_k + \alpha D_k) \geq \langle \nabla f(X_k), D_k \rangle + \frac{1}{2t} \|D_k\|_F^2 + h(X_k + D_k),$$

which implies

$$(1 - \alpha)\langle \nabla f(X_k), D_k \rangle + \frac{1 - \alpha^2}{2t} \|D_k\|_F^2 + h(X_k + D_k) - h(X_k + \alpha D_k) \leq 0. \quad (5.2)$$

Combining with the convexity of $h(X)$, (5.2) yields

$$\langle \nabla f(X_k), D_k \rangle + \frac{1 + \alpha}{2t} \|D_k\|_F^2 + h(X_k + D_k) - h(X_k) \leq 0.$$

By letting $\alpha \to 1$, we get

$$\langle \nabla f(X_k), D_k \rangle + h(X_k + D_k) - h(X_k) \leq -\frac{1}{t} \|D_k\|_F^2. \quad (5.3)$$

Combining (5.3) and the $L$-smoothness of $f$, we get

$$f(X_k + \alpha D_k) - f(X_k) + h(X_k + \alpha D_k) - h(X_k) \leq \langle \nabla f(X_k), \alpha D_k \rangle + \frac{\alpha}{2t} \|D\|_F^2 + \alpha (h(X_k + D_k) - h(X_k)) \leq -\frac{\alpha}{2t} \|D\|_F^2,$$

and (5.1) follows immediately. \square

The following lemma shows that if one cannot make any progress by solving (4.5), i.e., $D_k = 0$, then a stationary point is found.

Lemma 5.3. If $D_k = 0$, then $X_k$ is a stationary point of problem (1.1).

Proof. By Theorem 4.1 in \[75\], the optimality conditions for the subproblem (4.2) are given by

$$0 \in \tilde{\partial}_D \mathcal{L}(D, \Lambda) = \frac{1}{t} D + \text{grad} f(X_k) + \text{Proj}_{T_{X_k} \mathcal{M}} \partial h(X_k + D), \quad \text{and } D \in T_{X_k} \mathcal{M},$$

where $\mathcal{L}$ denotes the Lagrangian function of (4.2). If $D_k = 0$, it follows that

$$0 \in \text{grad} f(X_k) + \text{Proj}_{T_{X_k} \mathcal{M}} \partial h(X_k),$$

which is exactly the first-order necessary condition of problem (1.1) since $X_k \in \mathcal{M}$. \square

The following lemma shows that $\{F(X_k)\}$ is monotonically decreasing, where $\{X_k\}$ is generated by Algorithm 1.

Lemma 5.4. There exist constants $\bar{\alpha} > 0$ and $\bar{\beta} > 0$ such that for any $0 < \alpha \leq \min\{1, \bar{\alpha}\},$ the sequence $\{X_k\}$ generated by Algorithm 1 satisfies the following inequality:

$$F(X_{k+1}) - F(X_k) \leq -\bar{\beta} \|D_k\|_F^2. \quad (5.4)$$
Proof. We prove it by induction. Define $X_k^+ = X_k + \alpha D_k$. For $k = 0$, the set $\Omega := \{X \mid F(X) \leq F(X_0)\}$ is compact and $D_0$ is bounded, which are due to Assumption 5.1 and Lemma 5.2. By Lemma 3.6, (3.3) and (3.4) hold for $X_0$. Note that $X_{k+1} = \text{Retr}_{X_k}(\alpha D_k)$. From the Lipschitz continuity of $\nabla f(X)$, we have

$$f(X_{k+1}) - f(X_k) \leq \langle \nabla f(X_k), X_{k+1} - X_k \rangle + \frac{L}{2} \|X_{k+1} - X_k\|^2_F$$

$$= \langle \nabla f(X_k), X_{k+1} - X_k^+ + X_k^+ - X_k \rangle + \frac{L}{2} \|X_{k+1} - X_k\|^2_F$$

$$\leq M_2 \|\nabla f(X_k)\|_F \|\alpha D_k\|_F^2 + \alpha \langle \nabla f(X_k), D_k \rangle + \frac{LM_1}{2} \|\alpha D_k\|_F^2,$$

where the last inequality is due to (3.3) and (3.4). Since $\nabla f(X)$ is continuous on the compact set $\Omega$, there exists a constant $G > 0$ such that $\|\nabla f(X)\|_F \leq G$ for all $X \in \Omega$. It then follows from (5.5) that

$$f(X_{k+1}) - f(X_k) \leq c_0 \alpha^2 \|D_k\|_F^2 + \alpha \langle \nabla f(X_k), D_k \rangle,$$

where $c_0 = M_2 G + LM_1/2$. From (5.6) we can show the following inequalities:

$$F(X_{k+1}) - F(X_k) \leq \alpha \langle \nabla f(X_k), D_k \rangle + c_0 \alpha^2 \|D_k\|_F^2 + h(X_{k+1}) - h(X_k)$$

$$\leq \alpha \langle \nabla f(X_k), D_k \rangle + c_0 \alpha^2 \|D_k\|_F^2 + L_h \|X_{k+1} - X_k\|_F^2 + \alpha (h(X_k + D_k) - h(X_k))$$

$$\leq (c_0^2 + L_h M_2^2) \alpha^2 \|D_k\|_F^2 + \beta \|X_k\|_F^2 + \alpha X_k + D_k - h(X_k))$$

$$\leq \left[(c_0 + L_h M_2^2) \alpha^2 - L \alpha \right] \|D_k\|_F^2,$$

where the second inequality follows from the Lipschitz continuity of $h(X)$. Define function $\beta(\alpha) = -\langle c_0 + L_h M_2 \alpha^2 + L \alpha \rangle = 2(c_0 + L_h M_2) \alpha$. It is easy to see from (5.7) that

$$F(X_{k+1}) - F(X_k) \leq -\beta \|D_k\|_F^2, \quad \text{if } 0 < \alpha \leq \min\{1, \alpha\},$$

where

$$\beta = \begin{cases} \beta(\alpha_1) & \text{if } \alpha \leq 1, \\ \beta(1) & \text{if } \alpha > 1. \end{cases}$$

Thus, (5.4) holds for $k = 0$. By Assumption 5.1, $X_1 \in \Omega$ and $D_1$ is bounded. Suppose that (5.4) holds for $k \geq 1$, with the same argument, it follows that (5.4) holds $k + 1$ and $X_{k+1} \in \Omega$.

From Lemma 5.3, we know that $D_k = 0$ implies the optimality of $X_k$ for problem (1.1). As a result, we can define an $\epsilon$-stationary point of (1.1) as follows.

**Definition 5.5.** $X_k$ is called an $\epsilon$-stationary point of (1.1) if $D_k$ returned by (4.5) satisfies $\|D_k\|_F \leq \epsilon$.

We can thus use $\|D_k\|_F \leq \epsilon$ as a stopping criterion of Algorithm 1. From the sufficient decrease property Lemma 5.4, we obtain a similar result as that in the smooth setting [18, Theorem 2].

**Theorem 5.6.** Any limit point of the sequence $\{X_k\}$ generated by Algorithm 1 is a stationary point of problem (1.1). Furthermore, Algorithm 1 returns $X_k$ satisfying $\|D_k\|_F \leq \epsilon$ in at most $(F(X_0) - F^*)/(\beta \epsilon^2)$ iterations.

**Proof.** By Lemma 5.4 and the lower boundedness of $F(X)$, we have

$$\lim_{k \to \infty} \beta \|D_k\|_F^2 = 0.$$

Combining with Lemma 5.3, it follows that any limit point of $\{X_k\}$ is a stationary point. Moreover, since the sub-level set $\Omega = \{X \mid F(X) \leq F(X_0)\}$ is compact, there exists at least one limit point of the sequence $\{X_k\}$.

Furthermore, suppose that Algorithm 1 does not terminate after $K$ iterations, i.e., $\|D_k\|_F > \epsilon$ for all $k = 0, 1, \ldots, K - 1$. In this case, we have $F(X_0) - F^* \geq F(X_0) - F(X_K) \geq \beta \sum_{k=0}^{K-1} \|D_k\|_F^2 > \beta K \epsilon^2$. Therefore, Algorithm 1 terminates, i.e., finds an $\epsilon$-stationary point, after $K \geq (F(X_0) - F^*)/(\beta \epsilon^2)$ iterations. √
Remark 5.7. We remark here that when $F$ in (1.1) is smooth, i.e., the nonsmooth function $h$ vanishes, the iteration complexity in Theorem 5.6 matches the result given by Boumal et al. in [18], which mainly discusses iteration complexity of algorithms for solving manifold optimization with smooth objective. Zhang and Sra [79] analyzed the iteration complexity for some first-order methods, but they assumed that the objectives are geodesically convex. Such an assumption is rather restrictive, as it is known that every smooth function that is geodesically convex on a compact Riemannian manifold is constant [13]. Bento et al. [12] also established some iteration complexity results of gradient, subgradient, and proximal point methods. However, their results on gradient and subgradient methods require the objective function to be convex and the manifold to be of nonnegative curvature, and their results on proximal point methods only apply to convex function over Hadamard manifold.

6 Numerical Experiments

In this section, we apply our ManPG algorithm to solve the compressed modes (CM) (1.3) and sparse PCA (1.2) problems. We compare ManPG with two existing methods SOC [46] and PAMAL [21]. For both problems, we set the parameters $\delta = 0.01$ and $\gamma = 0.5$ in ManPG (Algorithm 1). Moreover, we use the polar decomposition as the retraction mapping. The codes were written in MATLAB and run on a standard PC with 3.20 GHz I5 Intel microprocessor and 16GB of memory.

6.1 Numerical results on CMs

For CM problems (1.3), both SOC (see [46]) and PAMAL (see [21]) rewrite the problem as

$$
\min_{X,Q,P \in \mathbb{R}^{n \times r}} \quad \text{Tr}(P^T HP) + \mu \|Q\|_1
$$

s.t. \quad Q = P, X = P, X^T X = I_r. \tag{6.1}

SOC employs a three-block ADMM to solve (6.1). PAMAL uses an inexact augmented Lagrangian method to solve (6.1) with the augmented Lagrangian function being minimized by a proximal alternating minimization algorithm proposed in [7].

Our ManPG algorithm solves (1.3) directly. Since the manifold is $\text{St}(n, r)$, its tangent space is given by

$$
T_{X} \mathcal{M} = \{ D \mid D^\top X + X^\top D = 0 \}. 
$$

Moreover, the SSN for solving the subproblem (4.2) when $\mathcal{M}$ is the Stiefel manifold can be presented in a more explicit way. Specifically, the monotone operator $E$ in (4.9) reduces to

$$
E(\Lambda) \equiv D(\Lambda)\top X_k + X_k^\top D(\Lambda). \tag{6.2}
$$

The vectorization of $E(\Lambda)$ can be represented by

$$
\text{vec}(E(\Lambda)) = (X_k^\top \otimes I_r)K_{nr}\text{vec}(D(\Lambda)) + (I_r \otimes X_k^\top)\text{vec}(D(\Lambda))
= (K_{rr} + I_r) (I_r \otimes X_k^\top) \left[ \text{prox}_h(\mu) \left( \text{vec}(X_k - t\nabla f(X_k)) + 2t(I_r \otimes X_k)\text{vec}(\Lambda) \right) - \text{vec}(X_k) \right],
$$

where $K_{mn}$ denotes the commutation matrix. By the chain rule, the generalized Jacobian of $\text{vec}(E(\Lambda))$ is given by

$$
\mathcal{G}(\text{vec}(\Lambda)) = 2t(K_{rr} + I_r)(I_r \otimes X_k^\top)\mathcal{J}(y)|_{y=\text{vec}(B(\Lambda))}(I_r \otimes X_k),
$$

where $\mathcal{J}(y)$ is the generalized Jacobian of $\text{prox}_h(\mu)(y)$ defined in (4.10). Note that since $\Lambda$ is a symmetric matrix, we can work on only the lower triangular part of $\Lambda$ and remove the duplicated entries in the upper triangular part. To do so, we use $\text{vec}(\Lambda)$ to denote the $\frac{1}{2}r(r+1)$-dimensional vector obtained from $\text{vec}(\Lambda)$ by eliminating all super-diagonal elements of $\Lambda$. It is known that there exists a unique $r^2 \times \frac{1}{2}r(r+1)$ matrix $U_r$, which is called the duplication matrix [52, Ch 3.8], such that $U_r\text{vec}(\Lambda) = \text{vec}(\Lambda)$. The Moore-Penrose inverse of $U_r$ is $U_r^+ = (U_r^\top U_r)^{-1}U_r^\top$ and satisfies $U_r^+\text{vec}(\Lambda) = \text{vec}(\Lambda)$. Note that both $U_r$ and $U_r^+$ have only $r^2$ nonzero elements. As a result, the generalized Jacobian of $\text{vec}(E(U_r\text{vec}(\Lambda)))$ is given by

$$
\mathcal{G}(\text{vec}(\Lambda)) = tU_r^+ \mathcal{G}(\text{vec}(\Lambda))U_r = 4tU_r^+(I_r \otimes X_k^\top)\mathcal{J}(y)|_{y=\text{vec}(B(\Lambda))}(I_r \otimes X_k)U_r, \tag{6.3}
$$
where we used the identity $K_r + I_r = 2U_r U_r^\top$. For each column of $B(\Lambda)$, we define the diagonal matrix $\Delta_i, i = 1, \ldots, r$, with diagonal entries given by

$$(\Delta_i)_{jj} = \begin{cases} 1, & \text{if } |B(\Lambda)_{ji}| > \mu t, \\ 0, & \text{otherwise.} \end{cases}$$

Therefore, (6.3) can be simplified to

$$G(\text{vec}(\Lambda)) = 4tU_r^\top \left( \begin{array}{ccc} X_k^\top \Delta_1 X_k \\ \vdots \\ X_k^\top \Delta_r X_k \end{array} \right) U_r. \quad (6.4)$$

Although $G(\text{vec}(\Lambda))$ is not symmetric, it can be shown that $G(\text{vec}(\Lambda)) + G(\text{vec}(\Lambda))^\top$ is symmetric and positive semidefinite. The rest of the procedure for applying ASSN is the same as what we described in Section 4.2.

In our numerical experiments, we tested the same problems as in [55] and [21]. In particular, we consider time-independent Schrödinger equation

$$\hat{H}\phi(x) = \lambda\phi(x), x \in \Omega,$$

where $\hat{H} = -\frac{1}{2}\Delta$ denotes the Hamiltonian and $\Delta$ denotes the Laplacian operator. $H$ is generated as a symmetric matrix formed by discretizing the Hamiltonian $\hat{H}$. We will focus on the 1D free-electron (FE) model. The FE model describes the behavior of valence electron in a crystal structure of a metallic solid and has $\hat{H} = -\frac{1}{2}\partial_x^2$. We consider the system on a domain $\Omega = [0, 50]$ with periodic boundary condition and discretize the domain with $n$ equally spaced nodes. The stepsize $t$ in Algorithm 1 was set to $1/(2\lambda_{\max}(\hat{H}))$, where $\lambda_{\max}(\hat{H})$ denotes the largest eigenvalue of $\hat{H}$.

In the SOC and PAMAL, we used the default setting of the parameters, i.e., the one suggested by [55] and [21]. Since the matrix $H$ is circulant, we used FFT to solve the linear systems in SOC and PAMAL, which is more efficient than directly inverting the matrices. We terminated ManPG when $\|D_k\|_F \leq 10^{-3}$ or the maximum iteration number 5000 was reached. For the inner iteration of ManPG, i.e., ASSN for solving (4.4), we terminated it when $\|E(\Lambda)\|_F^2 \leq 10^{-9}$ or the maximum iteration number 100 was reached. In all the tests of CMs, we ran ManPG first and denote $F_M$ as the returned objective value. We then ran SOC and PAMAL and terminated them when $F(X_k) \leq F_M$ and

$$\|Q_k - P_k\|_F/\max\{1, \|Q_k\|_F, \|P_k\|_F\} + \|X_k - P_k\|_F/\max\{1, \|X_k\|_F, \|P_k\|_F\} \leq 10^{-4}. \quad (6.5)$$

The latter measures the constraint violation of the reformulation (6.1). If (6.5) was not satisfied in 5000 iterations, we also terminated SOC and PAMAL.

For different settings of $(n, r, \mu)$, we run the three algorithms with 10 random initializations projected onto $St(n, r)$. We report the averaged CPU time, objective value, sparsity $sp$ and $Errr$ in Tables 1 and 2, where $Errr = \|Q_k - P_k\|_F + \|X_k - P_k\|_F$, $sp$ is the percentage of zeros and when computing $sp$, $X$ is truncated by zeroing out its entries whose magnitude is smaller than $10^{-5}$. From Tables 1 and 2 we see that to achieve the same function value, ManPG is faster than the other two algorithms in most cases. Moreover, we found that SOC does not converge when $\mu$ is large, and the stepsize of SOC needs to be tuned very carefully in order to guarantee convergence.

### 6.2 Numerical Results on Sparse PCA

In this section, we compare the three algorithms for solving sparse PCA (1.2). Note that there are other algorithms for sparse PCA such as the ones proposed in [43, 24], but these methods work only for the special case when $r = 1$, i.e., the constraint is a sphere $\{x \mid ||x||_2 = 1\}$. The algorithm proposed in [30] needs to smooth the $\ell_1$ norm in order to apply existing gradient-type methods and thus the sparsity of the solution is no longer guaranteed. Algorithms proposed in [85, 62, 44] do not impose orthogonal loading directions. In other words, they cannot impose both sparsity and orthogonality on the same variable. Therefore, we chose not to compare our ManPG with these algorithms.

14
| $r$ | $\mu$ | $F(X)$ | sp | CPU | $F(X)$ | sp | CPU | Err | $F(X)$ | sp | CPU | Err |
|-----|-----|------|---|----|------|---|----|---|------|---|----|---|
| 0.01 | 0.1501 | 58 | 0.06 | 0.1501 | 57 | 0.06 | 6.82e-05 | 0.1498 | 53 | 0.02 | 1.34e-04 |
| 0.02 | 0.5417 | 79 | 0.02 | 0.5416 | 71 | 0.01 | 1.50e-04 | 0.5417 | 67 | 0.23 | 1.27e-04 |
| 2 | 0.1 | 0.9427 | 84 | 0.02 | 0.943 | 71 | 0.04 | 7.87e-04 | 0.9425 | 78 | 0.27 | 1.24e-04 |
| 0.5 | 3.4020 | 92 | 0.02 | 3.4021 | 90 | 0.23 | 1.54e+00 | 3.4020 | 91 | 0.39 | 7.58e-05 |
| 1 | 5.8950 | 95 | 0.02 | 5.8950 | 94 | 0.23 | 1.56e+00 | 5.8960 | 94 | 1.65 | 1.79e-05 |
| 0.01 | 0.4028 | 39 | 0.12 | 0.426 | 38 | 0.37 | 1.64e-05 | 0.4564 | 30 | 4.37 | 3.90e-06 |
| 0.02 | 1.3590 | 73 | 0.11 | 1.358 | 71 | 0.08 | 1.35e-04 | 1.3680 | 71 | 6.13 | 3.02e-05 |
| 5 | 0.1 | 2.3590 | 82 | 0.09 | 2.357 | 76 | 0.05 | 2.46e-04 | 2.3590 | 79 | 1.53 | 9.07e-05 |
| 0.5 | 8.5050 | 92 | 0.15 | 8.5050 | 91 | 0.36 | 2.45e+00 | 8.5110 | 91 | 2.47 | 4.11e-05 |
| 1 | 14.7400 | 94 | 0.06 | 14.7400 | 94 | 0.27 | 1.56e+00 | 14.7400 | 94 | 5.44 | 5.40e-07 |
| 0.01 | 0.8044 | 35 | 0.2 | 0.903 | 25 | 0.5 | 2.48e-05 | 0.9548 | 17 | 7.37 | 2.14e-06 |
| 0.02 | 2.2370 | 63 | 0.54 | 2.236 | 64 | 0.25 | 8.70e-05 | 2.2910 | 67 | 9.6 | 1.36e-05 |
| 8 | 0.1 | 3.7870 | 78 | 0.59 | 3.786 | 76 | 0.16 | 2.73e-04 | 3.8200 | 77 | 12.14 | 9.46e-06 |
| 0.5 | 13.6100 | 92 | 0.38 | 13.6100 | 91 | 0.49 | 3.09e+00 | 13.6500 | 91 | 9.13 | 1.67e-05 |
| 1 | 23.5900 | 95 | 0.21 | 23.5900 | 94 | 0.49 | 3.12e+00 | 23.6100 | 94 | 11.96 | 4.69e-06 |

The random data matrices $A \in \mathbb{R}^{m \times n}$ considered in this section were generated by MATLAB function `randn(m, n)`, which means that the entries of $A$ follow standard Gaussian distribution. We then shifted the columns of $A$ such that their mean is equal to 0, and finally the columns were normalized.

We found that for sparse PCA, the subproblem (4.4) needs to be solved more accurately in order to obtain a good solution. As a result, we terminated ManPG when $\|D_k\|_F \leq 10^{-4}$ and terminated solving the subproblem when $\|E(\Lambda)\|_F^2 \leq 10^{-11}$. For ManPG, we set the stepsize $t = 1/(2\sigma_{\text{max}}^2)$, where $\sigma_{\text{max}}$ denotes the largest singular value of $A$. We used the same stopping criteria for SOC and PAMAL as for the CMs problems.

We report the objective value, sparsity, and CPU time of the three algorithms in Tables 3 and 4. In particular, the results in Table 3 correspond to initial points generated by the MATLAB function `randn(n, r)` and then projected onto $\text{St}(n, r)$, while results in Table 4 correspond to initial points set as the leading $r$ singular vectors of $A$. From Tables 3 and 4 we see that ManPG is much faster than SOC and PAMAL for obtaining solutions with the same objective value. Note that the parameter $\mu$ was chosen in a way that solutions with meaningful sparsity can be obtained.

### 7 Conclusions

Manifold optimization has attracted a lot of attention recently. In this paper, we proposed a proximal gradient method (ManPG) for solving manifold optimization problems with nonsmooth objectives. Different from the existing methods, our ManPG algorithm relies on proximal gradient information on the tangent space rather than subgradient information. Our experiments suggest that when combined with a regularized semi-smooth Newton method for finding the descent direction, ManPG performs efficiently and robustly. We believe that ManPG has great potential in solving many manifold optimization problems with nonsmooth objectives. We have discussed some representative applications in the introduction, and will work on solving these specific problems in follow-up papers.
Table 2: Comparison of ManPG, SOC and PAMAL for CMs, $n = 256$.

| $r$ | $\mu$ | ManPG | SOC | PAMAL |
|-----|------|-------|------|-------|
|     |      | $F(X)$ | sp CPU | $F(X)$ | sp CPU | Err | $F(X)$ | sp CPU | Err |
| 0.01| 0.05 | 0.1507 | 77 0.05 | 0.1507 | 77 0.09 | 5.74e-05 | 0.1528 | 76 0.19 | 1.04e-04 |
| 0.02| 0.03 | 0.5419 | 89 0.03 | 0.5417 | 86 0.01 | 1.53e-04 | 0.5417 | 86 0.01 | 1.64e-04 |
| 0.1 | 0.02 | 0.9425 | 92 0.03 | 0.9425 | 91 0.02 | 1.57e-04 | 0.9425 | 91 0.23 | 1.10e-04 |
| 0.5 | 0.02 | 3.4020 | 96 0.02 | 13.9800 | 0 0.28 | 1.53e+00 | 3.4020 | 96 0.32 | 6.27e-05 |
| 1   | 0.03 | 5.8950 | 97 0.03 | 27.5600 | 0 0.28 | 1.53e+00 | 5.8960 | 97 3.18 | 2.27e-05 |
| 0.01| 0.13 | 0.3823 | 74 0.10 | 0.3977 | 71 0.5  | 2.56e-05 | 0.4439 | 64 6.58 | 9.19e-06 |
| 0.02| 0.1  | 1.3560 | 88 0.1  | 1.3560 | 87 0.06 | 1.58e-04 | 1.3580 | 87 1.19 | 1.15e-04 |
| 0.1 | 0.09 | 2.3580 | 91 0.09 | 2.3570 | 89 0.04 | 2.20e-04 | 2.3570 | 90 1.55 | 1.32e-04 |
| 0.5 | 0.16 | 8.5050 | 96 0.16 | 31.7200 | 0 0.51 | 2.42e+00 | 8.5060 | 96 1.84 | 8.59e-05 |
| 1   | 0.08 | 14.7400| 97 0.08 | 66.3200 | 0 0.52 | 2.43e+00 | 14.7400| 97 10.55| 6.50e-13 |
| 0.01| 0.43 | 0.6259 | 70 0.43 | 0.7345 | 58 0.7  | 5.41e-05 | 0.8468 | 45 10.02| 1.02e-05 |
| 0.02| 0.5  | 2.1760 | 86 0.5  | 2.1760 | 86 0.26 | 8.45e-05 | 2.2040 | 86 14.42| 8.29e-06 |
| 0.1 | 0.41 | 3.7730 | 91 0.41 | 3.7730 | 89 0.12 | 2.44e+00 | 3.7820 | 91 10.17| 1.32e-05 |
| 0.5 | 0.21 | 13.6100| 96 0.21 | 52.1600 | 0 0.7  | 3.06e+00 | 13.6100| 96 4.34 | 6.74e-05 |
| 1   | 0.16 | 23.5800| 97 0.16 | 103.4000| 0 0.7  | 3.07e+00 | 23.6100| 97 18.91| 6.44e-13 |

Acknowledgements

We would like to thank Rongjie Lai for discussions on SOC and compressed modes problems, Lingzhou Xue for discussions on sparse PCA, Zaiwen Wen for discussions on semi-smooth Newton method, and Wen Huang and Ke Wei for discussions on manifolds.

References

[1] T. E. Abrudan, J. Eriksson, and V. Koivunen. Steepest descent algorithms for optimization under unitary matrix constraint. *IEEE Transactions on Signal Processing*, 56(3):1134–1147, 2008.
[2] T. E. Abrudan, J. Eriksson, and V. Koivunen. Conjugate gradient algorithm for optimization under unitary matrix constraint. *Signal Processing*, 89(9):1704–1714, 2009.
[3] P.-A. Absil and S. Hosseini. A collection of nonsmooth Riemannian optimization problems. *Springer International Series of Numerical Mathematics (ISNM)*, 2017.
[4] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.
[5] P.-A. Absil and I. V Oseledets. Low-rank retractions: a survey and new results. *Computational Optimization and Applications*, 62(1):5–29, 2015.
[6] M. Afonso, J. Bioucas-Dias, and M. Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE Transactions on Image Processing*, 19(9):2345–2356, 2010.
[7] H. Attouch, Z. Chbani, J. Peypouquet, and P. Redont. Fast convergence of inertial dynamics and algorithms with asymptotic vanishing viscosity. *Mathematical Programming*, 168(1-2):123–175, 2018.
[8] M. Bacak, R. Bergmann, G. Steidl, and A. Weinmann. A second order nonsmooth variational model for restoring manifold-valued images. *SIAM Journal on Scientific Computing*, 38:A567–A597, 2016.
[9] T. Bendory, Y. C. Eldar, and N. Boumal. Non-convex phase retrieval from STFT measurements. *IEEE Transactions on Information Theory*, 2018.
Table 3: Numerical results on sparse PCA (1.2) with random initialization

|       | ManPG          | SOC            | PAMAL          |       | ManPG          | SOC            | PAMAL          |
|-------|----------------|----------------|----------------|-------|----------------|----------------|----------------|
|       | $F(X)$         | sp  CPU        | $F(P)$         | sp  CPU | err       | $F(P)$         | sp  CPU        | err       |
| $r$   | $\mu$          |                | $\sigma$      |       |            |                | $\sigma$      |       |
|       | $\mu$          | $\sigma$      | $\mu$          | $\sigma$ | err       | $\mu$          | $\sigma$      | err       |
| $n=500, m=10$ |        |                |       |       |            |                |       |
| 2     | 0.5            | -2.336e+1      | 23.9           | 0.12  | -2.336e+1  | 23.9           | 0.17           | 9.1e-5   |
|       | 1              | -9.899e+0      | 49.8           | 0.05  | -9.899e+0  | 49.7           | 0.09           | 5.0e-7   |
| 5     | 0.5            | -5.014e+1      | 28.2           | 0.30  | -5.014e+1  | 28.3           | 0.66           | 2.5e-5   |
|       | 1              | -1.742e+1      | 55.2           | 0.19  | -1.748e+1  | 54.1           | 0.09           | 2.6e-4   |
| 10    | 0.5            | -7.967e+1      | 38.9           | 2.08  | -7.974e+1  | 38.9           | 0.84           | 3.4e-4   |
|       | 1              | -1.863e+1      | 70.4           | 1.97  | -2.279e+1  | 63.6           | 0.64           | 3.7e-4   |
| $n=2000, m=10$ |        |                |       |       |            |                |       |
| 2     | 0.5            | -9.415e+1      | 14.7           | 0.39  | -9.415e+1  | 14.7           | 4.67           | 1.4e-4   |
|       | 1              | -6.333e+1      | 30.0           | 0.20  | -6.333e+1  | 30.1           | 2.25           | 1.3e-4   |
| 5     | 0.5            | -2.234e+2      | 15.7           | 1.08  | -2.234e+2  | 15.8           | 10.89          | 5.2e-5   |
|       | 1              | -1.462e+2      | 31.6           | 1.45  | -1.468e+2  | 31.3           | 12.84          | 2.5e-4   |
| 10    | 0.5            | -3.776e+2      | 25.8           | 3.06  | -3.776e+2  | 25.8           | 8.83           | 3.4e-4   |
|       | 1              | -2.423e+2      | 40.2           | 3.18  | -2.420e+2  | 40.2           | 25.74          | 1.6e-5   |
| $n=5000, m=10$ |        |                |       |       |            |                |       |
| 2     | 0.3            | -4.893e+1      | 24.0           | 3.06  | -4.893e+1  | 23.9           | 19.89          | 1.5e-4   |
|       | 0.6            | -2.246e+1      | 47.1           | 5.00  | -2.246e+1  | 47.1           | 59.57          | 6.9e-6   |
| 5     | 0.3            | -1.208e+2      | 24.8           | 11.08 | -1.208e+2  | 24.8           | 99.21          | 2.5e-5   |
|       | 0.6            | -5.436e+1      | 48.4           | 7.11  | -5.454e+1  | 48.5           | 56.22          | 2.6e-4   |
| 10    | 0.3            | -2.356e+2      | 25.7           | 18.40 | -2.356e+2  | 25.7           | 92.93          | 3.4e-4   |
|       | 0.6            | -1.053e+2      | 49.4           | 16.51 | -1.059e+2  | 49.5           | 77.99          | 3.7e-4   |
Table 4: Numerical results on SPCA: initialization with singular vectors of $A$

| $n$ | $m$ | $r$ | $\mu$ | $\text{ManPG}$ | $\text{SOC}$ | $\text{PAMAL}$ |
|-----|-----|-----|------|----------------|-------------|--------------|
|     |     |     |      | $F(X)$ | sp | CPU | $F(P)$ | sp | CPU | err | $F(P)$ | sp | CPU | err |
| 500 | 10  | 0.5 | -2.336e+1 | 23.9 | 0.14 | -2.336e+1 | 23.9 | 0.09 | 1.5e-4 | -2.317e+1 | 24.2 | 4.62 | 1.2e-6 |
| 500 | 100 | 0.5 | -5.047e+1 | 28.7 | 0.47 | -5.047e+1 | 28.8 | 0.78 | 8.9e-6 | -5.047e+1 | 28.8 | 0.78 | 8.9e-6 |
| 2000 | 10 | 0.5 | -2.078e+1 | 53.6 | 0.37 | -2.078e+1 | 53.6 | 0.37 | 3.9e-6 | -2.078e+1 | 53.6 | 0.37 | 3.9e-6 |
| 2000 | 100 | 0.5 | -7.958e+1 | 56.3 | 0.35 | -7.958e+1 | 56.3 | 0.35 | 4.0e-6 | -7.958e+1 | 56.3 | 0.35 | 4.0e-6 |
| 10000 | 10 | 0.5 | -2.078e+1 | 53.6 | 0.37 | -2.078e+1 | 53.6 | 0.37 | 3.9e-6 | -2.078e+1 | 53.6 | 0.37 | 3.9e-6 |
| 10000 | 100 | 0.5 | -7.958e+1 | 56.3 | 0.35 | -7.958e+1 | 56.3 | 0.35 | 4.0e-6 | -7.958e+1 | 56.3 | 0.35 | 4.0e-6 |

18
[10] G. C. Bento, J. X. Cruz Neto, and P. R. Oliveira. Convergence of inexact descent methods for nonconvex optimization on Riemannian manifolds. https://arxiv.org/abs/1103.4828v1, 2011.

[11] G. C. Bento, J. X. Cruz Neto, and P. R. Oliveira. A new approach to the proximal point method: convergence on general Riemannian manifolds. J. Optim Theory Appl, 168:743–755, 2016.

[12] G. C. Bento, O. P. Ferreira, and J. G. Melo. Iteration-complexity of gradient, subgradient and proximal point methods on Riemannian manifolds. J. Optim Theory Appl, 173:548–562, 2017.

[13] R. L. Bishop and B. O’Neill. Manifolds of Negative Curvature. Transactions of the American Mathematical Society, 145:1–49, 1969.

[14] J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Mathematical Programming, 146:459–494, 2014.

[15] P. B. Borckmans, S. Easter Selvan, N. Boumal, and P.-A. Absil. A Riemannian subgradient algorithm for economic dispatch with valve-point effect. J. Comput. Appl. Math., 255:848–866, 2014.

[16] N. Boumal. Nonconvex phase synchronization. SIAM Journal on Optimization, 26(4):2355–2377, 2016.

[17] N. Boumal and P.-A. Absil. RTRMC: A Riemannian trust-region method for low-rank matrix completion. In NIPS, 2011.

[18] N. Boumal, P.-A. Absil, and C. Cartis. Global rates of convergence for nonconvex optimization on manifolds. IMA Journal of Numerical Analysis, 2018.

[19] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. The Journal of Machine Learning Research, 15(1):1455–1459, 2014.

[20] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning, 3(1):1–122, 2011.

[21] W. Chen, H. Ji, and Y. You. An augmented Lagrangian method for $\ell_1$-regularized optimization problems with orthogonality constraints. SIAM Journal on Scientific Computing, 38(4):B570–B592, 2016.

[22] A. Cherian and S. Sra. Riemannian dictionary learning and sparse coding for positive definite matrices. IEEE Trans. Neural Networks and Learning Systems, 2016.

[23] P. L. Combettes and J.-C. Pesquet. A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery. IEEE Journal of Selected Topics in Signal Processing, 1(4):564–574, 2007.

[24] A. d’Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. SIAM Review, 49(3):434–448, 2007.

[25] G. Dirr, U. Helmke, and C. Lageman. Nonsmooth Riemannian optimization with applications to sphere packing and grasping. Lagrangian and Hamiltonian Methods for Nonlinear Control, pages 28–45, 2006.

[26] J. Eckstein. Splitting methods for monotone operators with applications to parallel optimization. PhD thesis, Massachusetts Institute of Technology, 1989.

[27] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl., 20(2):303–353 (electronic), 1999.

[28] M. Fortin and R. Glowinski. Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems. North-Holland Pub. Co., 1983.

[29] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite-element approximations. Comp. Math. Appl., 2:17–40, 1976.

[30] M. Genicot, W. Huang, and N. T. Trendafilov. Weakly correlated sparse components with nearly orthonormal loadings. Geometric Science of Information, pages 484–490, 2015.

[31] R. Glowinski and P. Le Tallec. Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics. SIAM, Philadelphia, Pennsylvania, 1989.
[32] R. Glowinski and A. Marrocco. Approximation par éléments finis d'ordre un et résolution par pénalisation-dualité d'une classe de problèmes non linéaires. *R.A.I.R.O., R2*, pages 41–76, 1975.

[33] T. Goldstein and S. Osher. The split Bregman method for L1-regularized problems. *SIAM J. Imaging Sci.*, 2:323–343, 2009.

[34] G. H. Golub and C. F. Van Loan. *Matrix Computations*. JHU Press, 2012.

[35] P. Grohs and S. Hosseini. Nonsmooth trust region algorithms for locally Lipschitz functions on Riemannian manifolds. *IMA J. Numer. Anal.*, 36:1167–1192, 2016.

[36] S. Hosseini. Convergence of nonsmooth descent methods via Kurdyka-Lojasiewicz inequality on Riemannian manifolds. 2017.

[37] S. Hosseini and M. R. Pouryayevali. Generalized gradients and characterization of epi-lipschitz sets in Riemannian manifolds. *Nonlinear Analysis: Theory, Methods & Applications*, 72(12):3884–3895, 2011.

[38] S. Hosseini and A. Uschmajew. A Riemannian gradient sampling algorithm for nonsmooth optimization on manifolds. *SIAM J. Optim.*, 27(1):173–189, 2017.

[39] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

[40] W. Huang and P. Hand. Blind deconvolution by a steepest descent algorithm on a quotient manifold. *https://arxiv.org/pdf/1710.03309.pdf*, 2018.

[41] B. Jiang and Y.-H. Dai. A framework of constraint preserving update schemes for optimization on Stiefel manifold. *Mathematical Programming*, 153(2):535–575, 2015.

[42] B. Jiang, S. Ma, A. M.-C. So, and S. Zhang. Vector transport-free SVRG with general retraction for Riemannian optimization: Complexity analysis and practical implementation. *https://arxiv.org/abs/1705.09059v1*, 2017.

[43] I. Jolliffe, N. Trendafilov, and M. Uddin. A modified principal component technique based on the lasso. *Journal of computational and Graphical Statistics*, 12(3):531–547, 2003.

[44] M. Journee, Yu. Nesterov, P. Richtarik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *J. Mach. Learn. Res.*, 11:517–553, 2010.

[45] A. Kovnatsky, K. Glashoff, and M. M. Bronstein. MADMM: a generic algorithm for non-smooth optimization on manifolds. In *European Conference on Computer Vision*, pages 680–696. Springer, 2016.

[46] R. Lai and S. Osher. A splitting method for orthogonality constrained problems. *Journal of Scientific Computing*, 58(2):431–449, 2014.

[47] X. Li, D. Sun, and K.-C. Toh. A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems. *SIAM J. Optim.*, 28:433–458, 2018.

[48] P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16:964–979, 1979.

[49] H. Liu, A. M.-C. So, and W. Wu. Quadratic optimization with orthogonality constraint: Explicit Lojasiewicz exponent and linear convergence of retraction-based line-search and stochastic variance-reduced gradient methods. *Mathematical Programming, Series A*, 2018.

[50] H. Liu, M.-C. Yue, and A. M.-C. So. On the estimation performance and convergence rate of the generalized power method for phase synchronization. *SIAM J. Optim.*, 27(4):2426–2446, 2017.

[51] S. Ma. Alternating direction method of multipliers for sparse principal component analysis. *Journal of the Operations Research Society of China*, 1(2):253–274, 2013.

[52] J. R. Magnus and H. Neudecker. Matrix differential calculus with applications in statistics and econometrics. *Wiley series in probability and mathematical statistics*, 2018.
[53] R. Mifflin. Semismooth and semiconvex functions in constrained optimization. *SIAM J. Control Optim.*, 15:959–972, 1977.

[54] Y. Nishimori and S. Akaho. Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold. *Neurocomputing*, 67:106–135, 2005.

[55] V. Ozolinš, R. Lai, R. Caflisch, and S. Osher. Compressed modes for variational problems in mathematics and physics. *Proceedings of the National Academy of Sciences*, 110(46):18368–18373, 2013.

[56] K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[57] H. Qi and D. Sun. An augmented Lagrangian dual approach for the H-weighted nearest correlation matrix problem. *IMA Journal of Numerical Analysis*, 31:491–511, 2011.

[58] L. Qi and J. Sun. A nonsmooth version of Newton’s method. *Math. Program.*, 58:353–367, 1993.

[59] B. Savas and L.-H. Lim. Quasi-Newton methods on Grassmannians and multilinear approximations of tensors. *SIAM Journal on Scientific Computing*, 32(6):3352–3393, 2010.

[60] O. Shamir. A stochastic PCA and SVD algorithm with an exponential convergence rate. In *ICML*, 2015.

[61] O. Shamir. Fast stochastic algorithms for SVD and PCA: Convergence properties and convexity. In *ICML*, 2016.

[62] H. Shen and J. Z. Huang. Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis*, 99(6):1015–1034, 2008.

[63] M. V. Solodov and B. F. Svaiter. A globally convergent inexact Newton method for systems of monotone equations. In *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, pages 355–369. Springer, 1998.

[64] J. Sun, Q. Qu, and J. Wright. Complete dictionary recovery over the sphere i: Overview and the geometric picture. *IEEE Trans. Information Theory*, 63(2):853–884, 2017.

[65] J. Sun, Q. Qu, and J. Wright. A geometrical analysis of phase retrieval. *Foundations of Computational Mathematics*, 18(5):1131–1198, 2018.

[66] J. Tang and H. Liu. Unsupervised feature selection for linked social media data. In *SIGKDD*, pages 904–912. ACM, 2012.

[67] B. Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM J. Optim.*, 23(2):1214–1236, 2013.

[68] C. Wang, D. Sun, and K.-C. Toh. Solving log-determinant optimization problems by a Newton-CG primal proximal point algorithm. *SIAM J. Optimization*, 20:2994–3013, 2010.

[69] Y. Wang, W. Yin, and J. Zeng. Global convergence of ADMM in nonconvex nonsmooth optimization. *http://arxiv.org/abs/1511.06324*, 2015.

[70] Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434, 2013.

[71] X. Xiao, Y. Li, Z. Wen, and L. Zhang. A regularized semi-smooth Newton method with projection steps for composite convex programs. *Journal of Scientific Computing*, 76(1):364–389, 2018.

[72] J. Yang, D. Sun, and K.-C. Toh. A proximal point algorithm for log-determinant optimization with group Lasso regularization. *SIAM J. Optim.*, 23:857–893, 2013.

[73] J. Yang, W. Yin, Y. Zhang, and Y. Wang. A fast algorithm for edge-preserving variational multichannel image restoration. *SIAM Journal on Imaging Sciences*, 2(2):569–592, 2008.

[74] L. Yang, D. Sun, and K.-C. Toh. SDPNAL+: a majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints. *Mathematical Programming Computation*, 7(3):331–366, 2015.
[75] W. H. Yang, L.-H. Zhang, and R. Song. Optimality conditions for the nonlinear programming problems on Riemannian manifolds. *Pacific J. Optimization*, 10(2):415–434, 2014.

[76] Y. Yang, H. Shen, Z. Ma, Z. Huang, and X. Zhou. $\ell_{2,1}$-norm regularized discriminative feature selection for unsupervised learning. In *IJCAI*, volume 22, page 1589, 2011.

[77] C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, pages 894–942, 2010.

[78] H. Zhang, S. Reddi, and S. Sra. Fast stochastic optimization on Riemannian manifolds. In *NIPS*, 2016.

[79] H. Zhang and S. Sra. First-order methods for geodesically convex optimization. *Proceedings of the 29th Conference on Learning Theory*, 49:1617–1638, 2016.

[80] J. Zhang, S. Ma, and S. Zhang. Primal-dual optimization algorithms over Riemannian manifolds: an iteration complexity analysis. *https://arxiv.org/abs/1710.02236*, 2017.

[81] Y. Zhang, Y. Lau, H.-W. Kuo, S. Cheung, A. Pasupathy, and J. Wright. On the global geometry of sphere-constrained sparse blind deconvolution. In *CVPR*, 2017.

[82] X. Zhao, D. Sun, and K.-C. Toh. A Newton-CG augmented Lagrangian method for semidefinite programming. *SIAM Journal on Optimization*, 20:1737–1765, 2010.

[83] G. Zhou and K.-C. Toh. Superlinear convergence of a Newton-type algorithm for monotone equations. *Journal of optimization theory and applications*, 125(1):205–221, 2005.

[84] H. Zhu, X. Zhang, D. Chu, and L. Liao. Nonconvex and nonsmooth optimization with generalized orthogonality constraints: An approximate augmented Lagrangian method. *Journal of Scientific Computing*, 72(1):331–372, 2017.

[85] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *J. Comput. Graph. Stat.*, 15(2):265–286, 2006.

[86] H. Zou and L. Xue. A selective overview of sparse principal component analysis. *Proceedings of the IEEE*, 106(8):1311–1320, 2018.