Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

SIAM Review
Vol. 66, No. 2, pp. 319–352

© 2024 Society for Industrial and Applied Mathematics

Nonsmooth Optimization over the Stiefel Manifold and Beyond: Proximal Gradient Method and Recent Variants*

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

Abstract. We consider optimization problems over the Stiefel manifold whose objective function is the summation of a smooth function and a nonsmooth function. Existing methods for solving this class of problems converge slowly in practice, involve subproblems that can be as difficult as the original problem, or lack rigorous convergence guarantees. In this paper, we propose a manifold proximal gradient method (ManPG) for solving this class of problems. We prove that the proposed method converges globally to a stationary point and establish its iteration complexity for obtaining an ϵ-stationary point. Furthermore, we present numerical results on the sparse PCA and compressed modes problems to demonstrate the advantages of the proposed method. We also discuss some recent advances related to ManPG for Riemannian optimization with nonsmooth objective functions.

Key words. manifold optimization, Stiefel manifold, nonsmooth, proximal gradient method, iteration complexity, semismooth Newton method, stochastic algorithms, zeroth-order algorithms

MSC code. 90C30

DOI. 10.1137/24M1628578

Contents

*Published electronically May 9, 2024. This paper originally appeared in *SIAM Journal on Optimization*, Volume 30, Number 1, 2020, pages 210–239, under the title "Proximal Gradient Method for Nonsmooth Optimization over the Stiefel Manifold."
https://doi.org/10.1137/24M1628578
Funding: The work of the first author was supported in part by a startup fund from the University of Science and Technology of China. The work of the second author was supported in part by National Science Foundation grants DMS-2243650, CCF-2308597, CCF-2311275, and ECCS-2326591 and by a startup fund from Rice University. The work of the third author was supported in part by the Hong Kong Research Grants Council (RGC) General Research Fund (GRF) project CUHK 14204823.

†School of Mathematical Sciences, University of Science and Technology of China, Anhui, Hefei, China (shxchen@ustc.edu.cn).
‡Department of Computational Applied Mathematics and Operations Research, Rice University, Houston, TX 77005 USA (sqma@rice.edu).
§Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Sha Tin, New Territories, Hong Kong (manchoso@se.cuhk.edu.hk).
¶Hong Kong University of Science and Technology, Clear Water Bay, New Territories, Hong Kong (tongzhang@tongzhang-ml.org).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

1. Introduction. Optimization over Riemannian manifolds has recently drawn a lot of attention due to its application in many different fields, including low-rank matrix completion [18, 94], phase retrieval [9, 90], group synchronization [16, 71, 72], blind deconvolution [51], and dictionary learning [25, 89]. Manifold optimization seeks to minimize an objective function over a smooth manifold. Some commonly encountered manifolds include the sphere, the Stiefel manifold, the Grassmann manifold, and the Hadamard manifold. The recent monographs by Absil, Mahony, and Sepulchre [4] and Boumal [17] studied this topic in depth. In particular, several important classes of algorithms for manifold optimization with smooth objective, including line-search methods, Newton's method, and trust-region methods, have been studied. There are also many gradient-based algorithms for solving manifold optimization problems, including [101, 84, 85, 70, 54, 110]. However, all these methods require computing the derivative of the objective function and do not apply to the case where the objective function is nonsmooth.

In this paper, we focus on a class of nonsmooth nonconvex optimization problems over the Stiefel manifold that takes the form

$$\begin{aligned} \min \quad & F(X) := f(X) + h(X) \\ \text{s.t.} \quad & X \in \mathcal{M} := \mathrm{St}(n,r) = \{X : X \in \mathbb{R}^{n\times r}, X^\top X = I_r\}, \end{aligned} \tag{1.1}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

where I_r denotes the $r \times r$ identity matrix ($r \leq n$). Throughout this paper, we make the following assumptions about (1.1).

ASSUMPTION 1.1.

(i) *The function f is smooth and possibly nonconvex, and its gradient ∇f is Lipschitz continuous with Lipschitz constant L.*

(ii) *The function h is convex, possibly nonsmooth, and Lipschitz continuous with constant L_h.*

Note that here the smoothness, Lipschitz continuity, and convexity are interpreted when the function in question is considered as a function in the ambient Euclidean space.

We restrict our discussions in this paper to (1.1) because it already finds many important applications in practice. In the following we briefly mention some representative applications of (1.1).

Example 1. Sparse Principal Component Analysis. Principal component analysis (PCA), proposed by Pearson [79] and later developed by Hotelling [49], is one of the most fundamental statistical tools in analyzing high-dimensional data. Sparse PCA seeks principal components with very few nonzero components. For a 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

$$\begin{array}{ll} \min\limits_{X\in\mathbb{R}^{n\times r}} & -\mathrm{Tr}(X^\top A^\top AX) + \mu\|X\|_1 \\ \text{s.t.} & X^\top X = I_r, \end{array} \tag{1.2}$$

where $\mathrm{Tr}(Y)$ denotes the trace of matrix Y, the ℓ_1 norm is defined by $\|X\|_1 = \sum_{ij} |X_{ij}|$, and $\mu > 0$ is a weighting parameter. This is the original formulation of sparse PCA as proposed by Jolliffe, Trendafilov, and Uddin in [55], where the model is called SCoTLASS and imposes sparsity and orthogonality on the loading vectors simultaneously. When $\mu = 0$, problem (1.2) reduces to computing the leading r eigenvalues and the corresponding eigenvectors of $A^\top A$. When $\mu > 0$, the ℓ_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, problem (1.2) is relatively easy to solve because X reduces to a vector and the constraint set reduces to a sphere. However, the literature is very limited for the case $r > 1$. Existing works, including [118, 27, 86, 56, 73], do not impose orthogonal loading directions. As discussed in [56], "Simultaneously enforcing sparsity and orthogonality seems to be a hard (and perhaps questionable) task." We refer the interested reader to [119] for more details on existing algorithms for solving sparse PCA. As we will discuss later, our algorithm can efficiently solve (1.2) with $r > 1$ (i.e., imposing sparsity and orthogonality simultaneously).

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

$$\begin{array}{ll} \min\limits_{X\in\mathbb{R}^{n\times r}} & \mathrm{Tr}(X^\top HX) + \mu\|X\|_1 \\ \text{s.t.} & X^\top X = I_r, \end{array} \tag{1.3}$$

where H denotes the discretized Schrödinger operator. Note that the L_1 regularization reduces to the ℓ_1 norm of X after discretization. We refer the reader to [78] for

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

more details of this problem. Note that (1.2) and (1.3) are different in that H and $A^\top A$ have totally different structures. In particular, H is the discretized Schrödinger Hamiltonian, which is a block circulant matrix, while A in (1.2) usually comes from statistical data and thus $A^\top A$ is usually dense and unstructured. These differences may affect the performance of algorithms for solving them.

Example 3. Unsupervised Feature Selection. It is much more difficult to select the discriminative features in unsupervised learning than in supervised learning. There are some recent works that model this task as a manifold optimization problem of the form (1.1). For instance, the works [107] and [91] assume that there is a linear classifier W which assigns each data point x_i (where $i = 1, \ldots, n$) in the training data set to a class, and by denoting $G_i = W^\top x_i$, $[G_1, \ldots, G_n]$ gives a scaled label matrix which can be used to define some local discriminative scores. The target is to train a W such that the local discriminative scores are the highest for all the training data $x_1, \ldots, x_n$. It was suggested in [107] and [91] to solve the following model to find W:

$$\begin{array}{ll} \min\limits_{W \in \mathbb{R}^{n\times r}} & \mathrm{Tr}(W^\top M W) + \mu \|W\|_{2,1} \\ \text{s.t.} & W^\top W = I_r. \end{array}$$

Here, M is a given matrix computed from the input data, the $\ell_{2,1}$ norm is defined by $\|W\|_{2,1} = \sum_{i=1}^n \|W(i,:)\|_2$ with $W(i,:)$ being the ith row of W, which promotes the row sparsity of W, and the orthogonal constraint is imposed to avoid arbitrary scaling and the trivial solution of all zeros. We refer the reader to [107] and [91] for more details.

Example 4. Sparse Blind Deconvolution. Given the observations

$$y = a_0 \circledast x_0 \in \mathbb{R}^m,$$

how can one recover both the convolution kernel $a_0 \in \mathbb{R}^k$ and signal $x_0 \in \mathbb{R}^m$? Here, the signal x_0 is assumed to have a sparse and random support and $\circledast$ denotes the convolution operator. 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., [113]):

$$\begin{array}{ll} \min\limits_{a,x} & \|y - a \circledast x\|_2^2 + \mu \|x\|_1 \\ \text{s.t.} & \|a\|_2 = 1. \end{array}$$

Note that the sphere constraint here is a special case of the Stiefel manifold; i.e., $\mathrm{St}(k, 1)$.

Example 5. Nonconvex Regularizer. Problem (1.1) also allows nonconvex regularizer functions. For example, instead of using the ℓ_1 norm to promote sparsity, we can use the MCP (minimax concave penalty) function [109], which is popular in statistics. The MCP function is nonconvex and is given by

$$P(x) = \begin{cases} \lambda |x| - \frac{x^2}{2\lambda} & \text{if } |x| \le \gamma\lambda, \\ \frac{1}{2}\gamma\lambda^2 & \text{otherwise,} \end{cases}$$

where λ, γ are given parameters and $x \in \mathbb{R}$. If we replace the ℓ_1 norm in sparse PCA (1.2) by MCP, it reduces to

$$\begin{array}{ll} \min\limits_{X \in \mathbb{R}^{n\times r}} & -\mathrm{Tr}(X^\top A^\top A X) + \mu \sum\limits_{ij} P(X_{ij}) \\ \text{s.t.} & X^\top X = I_r. \end{array} \tag{1.4}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

It is easy to see that the objective function in (1.4) can be rewritten as $f_1(X)+f_2(X)$ with $f_1(X) = -\mathrm{Tr}(X^\top A^\top AX)+\mu(\sum_{ij} P(X_{ij})-\lambda\|X\|_1)$ and $f_2(X) = \mu\lambda\|X\|_1$. Note that f_1 is smooth and its gradient is Lipschitz continuous. Therefore, problem (1.4) is an instance of problem (1.1).

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 semismooth 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 ϵ-stationary point. Lastly, we present numerical results on the sparse PCA (1.2) and compressed modes (1.3) problems to show that our ManPG algorithm compares favorably with existing methods.

Notation. The following notation is adopted throughout this paper. The tangent space to the manifold $\mathcal{M}$ at the point X is denoted by $\mathrm{T}_X\mathcal{M}$. We use $\langle A, B\rangle = \mathrm{Tr}(A^\top B)$ to denote the Euclidean inner product of two matrices, A, B. We consider the Riemannian metric on $\mathcal{M}$ that is induced from the Euclidean inner product; i.e., for any $\xi, \eta \in \mathrm{T}_X\mathcal{M}$, we have $\langle \xi, \eta\rangle_X = \mathrm{Tr}(\xi^\top\eta)$. We use $\|X\|_\mathrm{F}$ to denote the Frobenius norm of X and $\|\mathcal{A}\|_\mathrm{op}$ to denote the operator norm of a linear operator $\mathcal{A}$. The Euclidean gradient of a smooth function f is denoted by ∇f, and the Riemannian gradient of f is denoted by $\mathrm{grad}\, f$. Note that by our choice of the Riemannian metric, we have $\mathrm{grad}\, f(X) = \mathrm{Proj}_{\mathrm{T}_X\mathcal{M}}\nabla f(X)$, the orthogonal projection of $\nabla f(X)$ onto the tangent space. According to [4], the projection of Y onto the tangent space at $X \in \mathrm{St}(n, r)$ is given by $\mathrm{Proj}_{T_X\,\mathrm{St}(n,r)}Y = (I_n - XX^\top)Y + \frac{1}{2}X(X^\top Y - Y^\top X)$. We use Retr to denote the retraction operation. For a convex function h, its Euclidean subgradient and Riemannian subgradient are denoted by ∂h and $\hat{\partial} h$, respectively. We use $\mathrm{vec}(X)$ to denote the vector formed by stacking the column vectors of X. The set of $r \times r$ symmetric matrices is denoted by S^r. Given an $X \in S^r$, we use $\overline{\mathrm{vec}}(X)$ to denote the $\frac{1}{2}r(r+1)$-dimensional vector obtained from $\mathrm{vec}(X)$ by eliminating all superdiagonal elements of X. We denote $Z \succeq 0$ if $(Z + Z^\top)/2$ is positive semidefinite. The proximal mapping of h at point X is defined by $\mathrm{prox}_h(X) = \mathrm{argmin}_Y \frac{1}{2}\|Y - X\|_\mathrm{F}^2 + h(Y)$.

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. We then present the main algorithm ManPG and the semismooth Newton method for solving the subproblem in section 4. In section 5, we establish the global convergence of ManPG and analyze its iteration complexity for obtaining an ϵ-stationary point. We report the numerical results of ManPG on solving compressed modes problems in physics and sparse PCA in statistics in section 6. In section 7, we discuss some recent algorithms for manifold optimization with nonsmooth objective functions that are motivated by ManPG. These include the manifold proximal point algorithm, manifold proximal linear algorithm, stochastic ManPG, zeroth-order ManPG, Riemannian proximal gradient method, and Riemannian proximal Newton method. Finally, we draw some concluding remarks in section 8.

2. Nonsmooth Optimization over Riemannian Manifold.

Unlike manifold optimization with smooth objective functions, which has been studied extensively in

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

the monographs [4, 17], the literature on manifold optimization with nonsmooth objective functions is relatively limited. Numerical algorithms for manifold optimization with nonsmooth objective functions can be roughly classified into three categories: subgradient-oriented methods, proximal point algorithms, and operator-splitting methods. We now briefly discuss the existing works in these three categories.

2.1. Subgradient-Oriented Methods. Algorithms in the first category include those proposed in [33, 15, 43, 45, 48, 46, 8, 28, 42], which are all subgradient-oriented methods. Ferreira and Oliveira [33] studied the convergence of subgradient methods for minimizing a convex function over a Riemannian manifold. The subgradient method generates the iterates via

$$X_{k+1} = \exp_{X_k}(t_k V_k),$$

where $\exp_{X_k}$ is the exponential mapping at X_k and V_k denotes a Riemannian subgradient of the objective. Like the subgradient method in Euclidean space, the stepsize t_k is chosen to be diminishing to guarantee convergence. However, the result in [33] does not apply to (1.1) because every smooth function that is convex on a compact Riemannian manifold is a constant [13]. This motivated some more advanced works on the Riemannian subgradient method. Specifically, Dirr, Helmke, and Lageman [28] and Borckmans et al. [15] proposed manifold subgradient methods for the case where the objective function is the pointwise maximum of smooth functions. In this case, a generalized gradient can be computed and a descent direction can be found by solving a quadratic program. Grohs and Hosseini [43] proposed a Riemannian ε-subgradient method. Hosseini and Uschmajew [48] proposed a Riemannian gradient sampling algorithm. Hosseini, Huang, and Yousefpour [46] generalized the Wolfe conditions and extended the BFGS algorithm to optimize nonsmooth functions on Riemannian manifolds. Grohs and Hosseini [42] proposed a generalization of a nonsmooth trust-region method for manifold optimization. Hosseini [45] studied the convergence of some subgradient-oriented descent methods based on the Kurdyka–Łojasiewicz (KŁ) inequality. Roughly speaking, all the methods studied in [28, 15, 43, 48, 46, 42, 45] require subgradient information to build a quadratic program to find a descent direction:

$$\hat{g} \longleftarrow \min_{g \in \mathrm{conv}(W)} \|g\|. \tag{2.1}$$

Here, $\mathrm{conv}(W)$ denotes the convex hull of set $W = \{G_j, j = 1, \ldots, J\}$, G_j is the Riemannian gradient of a differentiable point around the current iterate X, and J usually needs to be larger than the dimension of $\mathcal{M}$. Subsequently, the iterate X is updated by $X^+ = \mathrm{Retr}_X(\alpha\hat{g})$, where the stepsize α is found by line search. For high-dimensional problems on the Stiefel manifold $\mathrm{St}(n, r)$, problem (2.1) can be difficult to solve because n is large. Since the 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 would not be as efficient as gradient algorithms and proximal gradient algorithms on the manifold in practice.

2.2. Proximal Point Algorithms. Proximal point algorithms (PPAs) for manifold optimization are also studied in the literature. Ferreira and Oliveira [34] extended PPAs for manifold optimization, which in each iteration needs to minimize the original function plus a proximal term over the manifold. However, there are two issues that limit its applicability. The first is that the subproblem can be as difficult as

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

the original problem. For example, Bačák et al. [8] suggested using the subgradient method to solve the subproblem, but they required the subproblem to be in the form of the pointwise maximum of smooth functions tackled in [15]. The second is that the discussions in the literature mainly focus on the Hadamard manifold and heavily exploit the convexity assumption of the objective function. Thus, they do not apply to compact manifolds such as $\mathrm{St}(n,r)$. Bento, Cruz Neto, and Oliveira [11] aimed to resolve the second issue and proved the convergence of the PPA for more general Riemannian manifolds under the assumption that the KŁ inequality holds for the objective function. In [10], Bento, Cruz Neto, and Oliveira analyzed the convergence of some inexact descent methods based on the KŁ inequality, including the PPA and steepest descent method. In a more recent work [12], Bento, Ferreira, and Melo studied the iteration complexity of the PPA under the assumption that the constraint set is the Hadamard manifold and the objective function is convex. Nevertheless, the results in [34, 11, 10, 12] seem to be only of theoretical interest because no numerical results were shown. As mentioned earlier, this could be due to the difficulty in solving the PPA subproblems.

2.3. Operator-Splitting Methods. Operator-splitting methods do not require subgradient information, and existing works in the literature mainly focus on the Stiefel manifold. Note that (1.1) is challenging because of the combination of two difficult terms: the Riemannian manifold and the nonsmooth objective. If only one of them is present, then the problem is relatively easy 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, which has a long history [39, 36, 68, 35, 38, 29]. The renaissance of ADMM was initiated by several papers around 2007–2008, when it was successfully applied to solve various signal processing [26] and image processing [104, 40, 6] problems. The recent survey paper [21] popularized this method in many areas. Recently, there has been emerging interest in using ADMM to solve manifold optimization problems of the form (1.1); see, e.g., [58, 57, 112, 98]. However, the algorithms presented in these papers either lack convergence guarantee [58, 57] or their convergence needs further conditions that do not apply to (1.1) [98, 112].

Here, we briefly describe the SOC method (splitting method for orthogonality constrained problems) presented in [58]. The SOC method aims to solve

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

by introducing an auxiliary variable P and considering the following reformulation:

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

By associating a Lagrange multiplier Λ with the linear equality constraint, the augmented Lagrangian function of (2.2) can be written as

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

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

$$\begin{aligned} P^{k+1} &:= \operatorname*{argmin}_P\ \mathcal{L}_\beta(P, X^k; \Lambda^k),\\ X^{k+1} &:= \operatorname*{argmin}_X\ \mathcal{L}_\beta(P^{k+1}, X; \Lambda^k)\ \text{s.t. } X \in \mathcal{M},\\ \Lambda^{k+1} &:= \Lambda^k - \beta(P - X). \end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Note that the X-subproblem corresponds to the projection onto $\mathcal{M}$, and the P-subproblem is an unconstrained problem whose complexity depends 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 an easily computable proximal mapping, then the P-subproblem can be solved directly by computing the proximal mapping of J.

The MADMM (manifold ADMM) algorithm presented in [57] aims to solve the problem

$$\min_{X,Z} \ f(X) + g(Z) \quad \text{s.t.} \quad Z = AX,\ X \in \mathrm{St}(n,r), \tag{2.3}$$

where f is smooth and g is nonsmooth with an easily computable proximal mapping. The augmented Lagrangian function of (2.3) is

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

and the MADMM algorithm generates its iterates as follows:

$$\begin{aligned} X^{k+1} &:= \operatorname*{argmin}_X \ \mathcal{L}_\beta(X, Z^k; \Lambda^k) \ \text{ s.t. } \ X \in \mathrm{St}(n,r),\\ Z^{k+1} &:= \operatorname*{argmin}_Z \ \mathcal{L}_\beta(X^{k+1}, Z; \Lambda^k),\\ \Lambda^{k+1} &:= \Lambda^k - \beta(Z^{k+1} - AX^{k+1}). \end{aligned}$$

Note that the X-subproblem is a smooth optimization problem on the Stiefel manifold, and the authors of [57] suggested using the MANOPT toolbox [20] 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 from the literature. Though there are some recent works that analyze the convergence of ADMM for nonconvex problems [98, 112], their results need further conditions that do not apply to (1.1) or its reformulations (2.2) and (2.3).

More recently, some other variants of the augmented Lagrangian method have been proposed to deal with (1.1). In [24], Chen, Ji, and You proposed a method that hybridizes an augmented Lagrangian method and the proximal alternating minimization method [7]. More specifically, this method solves the following reformulation of (1.1):

$$\min_{X,Q,P} \ f(P) + h(Q) \quad \text{s.t.} \quad Q = X,\ P = X,\ X \in \mathrm{St}(n,r). \tag{2.4}$$

By associating Lagrange multipliers Λ_1 and Λ_2 with the two linear equality constraints, the augmented Lagrangian function of (2.4) 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 (2.4) is then given by

$$\begin{aligned} (X^{k+1}, Q^{k+1}, P^{k+1}) &:= \operatorname*{argmin}_{X,Q,P} \ \mathcal{L}_\beta(X,Q,P;\Lambda_1^k,\Lambda_2^k) \ \text{ s.t. } \ X \in \mathrm{St}(n,r),\\ \Lambda_1^{k+1} &:= \Lambda_1^k - \beta(Q^{k+1} - X^{k+1}),\\ \Lambda_2^{k+1} &:= \Lambda_2^k - \beta(P^{k+1} - X^{k+1}). \end{aligned} \tag{2.5}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Note that the subproblem in (2.5) is still difficult to solve. Therefore, the authors of [24] suggested using the proximal alternating minimization method [7] to solve the subproblem in (2.5) inexactly and named their method PAMAL. They proved that under certain conditions, any limit point of the sequence generated by PAMAL is a KKT point of (2.4). It should be pointed out that the proximal alternating minimization procedure involves many parameters that need to be fine-tuned in order to solve the subproblem inexactly. Our numerical results in section 6 indicate that the performance of PAMAL depends significantly on the setting of these parameters.

In [116], Zhu et al. studied another algorithm called EPALMAL for solving (1.1), which 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 the PALM algorithm [14], while PAMAL uses PAM [7]. It is also shown in [116] 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 the elements of manifold optimization that will be needed in the study of (1.1). In fact, our discussion in this section applies to the case where $\mathcal{M}$ is any embedded submanifold of a Euclidean space. To begin, we say that a function F is 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 the Euclidean space $\mathcal{E}$, then it is also locally Lipschitz continuous when restricted to the embedded submanifold $\mathcal{M}$ of $\mathcal{E}$.

DEFINITION 3.1 (generalized Clarke subdifferential [47]). *For a locally Lipschitz function F on $\mathcal{M}$, the Riemannian generalized directional derivative of F at $X \in \mathcal{M}$ in the direction V is defined by*

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

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

$$\hat{\partial} F(X) = \{\xi \in \mathrm{T}_X\mathcal{M} : \langle \xi, V \rangle \le F^\circ(X,V) \ \forall V \in \mathrm{T}_X\mathcal{M}\}.$$

DEFINITION 3.2 ([106]). *A function f is said to be regular at $X \in \mathcal{M}$ along $\mathrm{T}_X\mathcal{M}$ if*

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

For a smooth function f, we know that $\operatorname{grad} f(X) = \mathrm{Proj}_{\mathrm{T}_X\mathcal{M}} \nabla f(X)$ by our choice of the Riemannian metric. According to Theorem 5.1 in [106], for a regular function F, we have $\hat{\partial} F(X) = \mathrm{Proj}_{\mathrm{T}_X\mathcal{M}}(\partial F(X))$. Moreover, the function $F = f + h$ in (1.1) is regular according to Lemma 5.1 in [106]. Therefore, we have $\hat{\partial} F(X) = \mathrm{Proj}_{\mathrm{T}_X\mathcal{M}}(\nabla f(X) + \partial h(X)) = \operatorname{grad} f(X) + \mathrm{Proj}_{\mathrm{T}_X\mathcal{M}}(\partial h(X))$. By Theorem 4.1 in [106], the first-order necessary condition of (1.1) is given by $0 \in \operatorname{grad} f(X) + \mathrm{Proj}_{\mathrm{T}_X\mathcal{M}}(\partial h(X))$.

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.,* $0 \in \operatorname{grad} f(X) + \mathrm{Proj}_{\mathrm{T}_X\mathcal{M}}(\partial h(X))$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

A classic geometric concept in the study of manifolds is that of an exponential mapping, which defines a geodesic curve on the manifold. However, the exponential mapping is difficult to compute in general. The concept of a retraction [4], which is a first-order approximation of the exponential mapping and can be more amenable to computation, is given as follows.

DEFINITION 3.4 ([4, Definition 4.1.1]). *A retraction on a differentiable manifold* $\mathcal{M}$ *is a smooth mapping* Retr *from the tangent bundle* $\mathrm{T}\mathcal{M}$ *onto* $\mathcal{M}$ *satisfying the following two conditions (here,* Retr_X *denotes the restriction of* Retr *onto* $\mathrm{T}_X\mathcal{M}$*):*

1. $\mathrm{Retr}_X(0) = X$ *for all* $X \in \mathcal{M}$*, where* 0 *denotes the zero element of* $\mathrm{T}_X\mathcal{M}$*.*
2. *For any* $X \in \mathcal{M}$*, it holds that*

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

Remark 3.5. Since $\mathcal{M}$ is an embedded submanifold of $\mathbb{R}^{n\times r}$, we can treat X and ξ as elements in $\mathbb{R}^{n\times r}$, and hence their sum is well defined. The second condition in Definition 3.4 ensures that $\mathrm{Retr}_X(\xi) = X + \xi + \mathcal{O}(\|\xi\|_{\mathrm{F}}^2)$ and $D\mathrm{Retr}_X(0) = \mathrm{Id}$, where $D\mathrm{Retr}_X$ is the differential of Retr_X and Id denotes the identity mapping. For more details about retraction, we refer the reader to [4, 19] and the references therein.

The retraction onto the Euclidean space is simply the identity mapping; i.e., $\mathrm{Retr}_X(\xi) = X + \xi$. For the Stiefel manifold $\mathrm{St}(n, r)$, common retractions include the exponential mapping [30]

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

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

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

the QR decomposition

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

where $\mathrm{qf}(A)$ is the Q factor of the QR factorization of A; and the Cayley transformation [101]

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

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

For any matrix $Y \in \mathbb{R}^{n\times r}$ with $r \le n$, its orthogonal projection onto the Stiefel manifold $\mathrm{St}(n, r)$ is given by $UI_rV^\top$, where U, V are the left and right singular vectors of Y, respectively. If Y has full rank, then the projection can be computed by $Y(Y^\top Y)^{-1/2}$, which is the same as the polar decomposition. The total cost of computing the projection $UI_rV^\top$ is $8nr^2 + \mathcal{O}(r^3)$ flops, where the SVD needs $6nr^2 + \mathcal{O}(r^3)$ flops [41] and the formation of $UI_rV^\top$ needs $2nr^2$ flops. By comparison, if $Y = X + \xi$ and $\xi \in T_X\mathcal{M}$, then the exponential mapping takes $8nr^2 + \mathcal{O}(r^3)$ flops and the polar decomposition takes $3nr^2 + \mathcal{O}(r^3)$ flops, where $\xi^\top\xi$ needs nr^2 flops and the remaining $2nr^2 + \mathcal{O}(r^3)$ flops come from the final assembly. Thus, polar decomposition is cheaper than the projection. Moreover, the QR decomposition of $X + \xi$ takes $2nr^2 + \mathcal{O}(r^3)$ flops. For the Cayley transformation of $X + \xi$, the total cost is $7nr^2 + \mathcal{O}(r^3)$ [101, 53]. In our algorithm, to be introduced later, we need to perform one retraction operation

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

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 to use in the algorithm is still an interesting question to investigate in practice; see also the discussion after Theorem 3 of [70].

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

FACT 3.6 ([19, 70]). *Let $\mathcal{M}$ be a compact embedded submanifold of a Euclidean space. For all $X \in \mathcal{M}$ and $\xi \in \mathrm{T}_X\mathcal{M}$, there exist constants $M_1 > 0$ and $M_2 > 0$ such that the following two inequalities hold:*

$$\|\mathrm{Retr}_X(\xi) - X\|_F \le M_1\|\xi\|_F \quad \forall X \in \mathcal{M},\ \xi \in \mathrm{T}_X\mathcal{M}, \tag{3.1}$$

$$\|\mathrm{Retr}_X(\xi) - (X+\xi)\|_F \le M_2\|\xi\|_F^2 \quad \forall X \in \mathcal{M},\ \xi \in \mathrm{T}_X\mathcal{M}. \tag{3.2}$$

4. Proximal Gradient Method on the Stiefel Manifold.

4.1. The ManPG Algorithm. For manifold optimization problems with smooth objective functions, the Riemannian gradient method [1, 4, 77] has been one of the main methods of choice. A generic update formula of the Riemannian gradient method for solving

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

is

$$X_{k+1} := \mathrm{Retr}_{X_k}(\alpha_k V_k),$$

where F is smooth, V_k is a descent direction of F in the tangent space $\mathrm{T}_{X_k}\mathcal{M}$, and α_k is a stepsize. Recently, Boumal, Absil, and Cartis [19] established the sublinear rate of the Riemannian gradient method for returning a point X_k satisfying $\|\mathrm{grad}\, F(X_k)\|_F < \epsilon$. Liu, So, and Wu [70] proved that the Riemannian gradient method converges linearly for quadratic minimization over the Stiefel manifold. Other methods for solving manifold optimization problems with smooth objective functions have also been studied in the literature, including conjugate gradient methods [4, 2], trust-region methods [4, 19], and Newton-type methods [4, 83].

We now develop our ManPG algorithm for solving (1.1). Since the objective function in (1.1) has a composite structure, a natural idea is to extend the proximal gradient method from the Euclidean setting to the manifold setting. The proximal gradient method for solving $\min_X F(X) := f(X) + h(X)$ in the Euclidean setting generates the iterates as follows:

$$X_{k+1} := \operatorname*{argmin}_Y f(X_k) + \langle \nabla f(X_k), Y - X_k\rangle + \frac{1}{2t}\|Y - X_k\|_{\mathrm{F}}^2 + h(Y). \tag{4.1}$$

In other words, one minimizes the quadratic model $Y \mapsto f(X_k) + \langle \nabla f(X_k), Y - X_k\rangle + \frac{1}{2t}\|Y - X_k\|_{\mathrm{F}}^2 + h(Y)$ of F at X_k in the kth iteration, where $t > 0$ is a parameter that can be regarded as the stepsize. It is known that the quadratic model is an upper bound of F when $t \le 1/L$, where L is the Lipschitz constant of ∇f. The subproblem (4.1) corresponds to the proximal mapping of h, and the efficiency of the proximal gradient method relies on the assumption that (4.1) is easy to solve. For (1.1), in order to deal with the manifold constraint, we need to ensure that the descent direction lies in the tangent space. This motivates the following subproblem for finding the descent

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

direction V_k in the kth iteration, with $t > 0$ being the stepsize:

$$V_k := \operatorname*{argmin}_V \quad \langle \operatorname{grad} f(X_k), V\rangle + \frac{1}{2t}\|V\|_F^2 + h(X_k + V) \quad \text{s.t.} \quad V \in \mathrm{T}_{X_k}\mathcal{M}. \tag{4.2}$$

Here and also in later discussions, we can interpret $X_k + V$ as the sum of X_k and V in the ambient Euclidean space $\mathbb{R}^{n\times r}$, as $\mathcal{M}$ is an embedded submanifold of $\mathbb{R}^{n\times r}$. Note that (4.2) is different from (4.1) in two places: (i) the Euclidean gradient ∇f is changed to the Riemannian gradient $\operatorname{grad} f$, and (ii) the descent direction V_k is restricted to the tangent space. Following the definition of $\operatorname{grad} f$, we have

$$\langle \operatorname{grad} f(X_k), V\rangle = \langle \nabla f(X_k), V\rangle \quad \forall V \in \mathrm{T}_{X_k}\mathcal{M},$$

which implies that (4.2) can be rewritten as

$$V_k := \operatorname*{argmin}_V \quad \langle \nabla f(X_k), V\rangle + \frac{1}{2t}\|V\|_F^2 + h(X_k + V) \quad \text{s.t.} \quad V \in \mathrm{T}_{X_k}\mathcal{M}. \tag{4.3}$$

As a result, we do not need to compute the Riemannian gradient $\operatorname{grad} f$. Rather, only the Euclidean gradient ∇f is needed. Note that without considering the constraint $V \in \mathrm{T}_{X_k}\mathcal{M}$, the subproblem (4.3) computes a proximal gradient step. Therefore, the subproblem (4.3) can be viewed as a proximal gradient step restricted to the tangent space $\mathrm{T}_{X_k}\mathcal{M}$. Since, for an arbitrary stepsize $\alpha_k > 0$, the point $X_k + \alpha_k V_k$ does not necessarily lie on the manifold $\mathcal{M}$, we perform a retraction to bring it back to $\mathcal{M}$.

Our ManPG algorithm for solving (1.1) is described in Algorithm 1. Note that ManPG involves an Armijo line-search procedure to determine the stepsize α. As we will show in section 5, this backtracking line-search procedure is well defined; i.e., it will terminate after a finite number of steps.

Algorithm 1 Manifold proximal gradient method (ManPG) for solving (1.1).

Input: initial point $X_0 \in \mathcal{M}$, $\gamma \in (0,1)$, stepsize $t > 0$
for $k = 0, 1, \ldots$ **do**
  obtain V_k by solving the subproblem (4.3)
  set $\alpha = 1$
  while $F(\mathrm{Retr}_{X_k}(\alpha V_k)) > F(X_k) - \dfrac{\alpha\|V_k\|_{\mathrm{F}}^2}{2t}$ **do**
    $\alpha = \gamma\alpha$
  end while
  set $X_{k+1} = \mathrm{Retr}_{X_k}(\alpha V_k)$
end for

4.2. Regularized Semismooth Newton Method for Subproblem (4.3). The main computational effort of Algorithm 1 lies in solving the convex subproblem (4.3). We have conducted extensive numerical experiments and found that the semismooth Newton (SSN) method is very suitable for this purpose. The notion of semismoothness was originally introduced by Mifflin [75] for real-valued functions and later extended to vector-valued mappings by Qi and Sun [81]. A pioneering work on the SSN method was due to Solodov and Svaiter [88], in which the authors proposed a globally convergent Newton method by exploiting the structure of monotonicity and established a

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

local superlinear convergence rate under the conditions that the generalized Jacobian is semismooth and nonsingular at the global optimal solution. The convergence rate guarantee was later extended in [115] to the setting where the generalized Jacobian is not necessarily nonsingular. Recently, the SSN method has received a significant amount of attention due to its success in solving structured convex problems to a high accuracy. In particular, it has been successfully applied to solving SDP [114, 105], LASSO [67], nearest correlation matrix estimation [80], clustering [96], sparse inverse covariance selection [103], and composite convex minimization [102].

In the following, we show how to apply the SSN method to solve the subproblem (4.3) with $\mathcal{M} = \mathrm{St}(n, r)$. The tangent space to $\mathcal{M} = \mathrm{St}(n, r)$ is given by

$$\mathrm{T}_X\mathcal{M} = \{V \mid V^\top X + X^\top V = 0\}.$$

For ease of notation, we define the linear operator $\mathcal{A}_k$ by $\mathcal{A}_k(V) := V^\top X_k + X_k^\top V$ and rewrite the subproblem (4.3) as

$$V_k := \operatorname*{argmin}_{V} \quad \langle \nabla f(X_k), V\rangle + \frac{1}{2t}\|V\|_F^2 + h(X_k + V) \quad \text{s.t.} \quad \mathcal{A}_k(V) = 0. \tag{4.4}$$

By associating a Lagrange multiplier Λ with the linear equality constraint, the Lagrangian function of (4.4) can be written as

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

and the KKT system of (4.4) is given by

$$0 \in \partial_V \mathcal{L}(V; \Lambda), \quad \mathcal{A}_k(V) = 0. \tag{4.5}$$

The first condition in (4.5) implies that V can be computed by

$$V(\Lambda) = \mathrm{prox}_{th}(B(\Lambda)) - X_k \quad \text{with} \quad B(\Lambda) = X_k - t(\nabla f(X_k) - \mathcal{A}_k^*(\Lambda)), \tag{4.6}$$

where $\mathcal{A}_k^*$ denotes the adjoint operator of $\mathcal{A}_k$. By substituting (4.6) into the second condition in (4.5), we see that Λ satisfies

$$E(\Lambda) \equiv \mathcal{A}_k(V(\Lambda)) = V(\Lambda)^\top X_k + X_k^\top V(\Lambda) = 0. \tag{4.7}$$

We will use the SSN method to solve (4.7). To do so, we need to first show that the operator E is monotone and Lipschitz continuous. For any $\Lambda_1, \Lambda_2 \in S^r$, we have

$$\begin{aligned} &\|E(\Lambda_1) - E(\Lambda_2)\|_{\mathrm{F}} \\ &\le \|\mathcal{A}_k\|_{\mathrm{op}} \|\mathrm{prox}_{th}(B(\Lambda_1)) - \mathrm{prox}_{th}(B(\Lambda_2))\|_{\mathrm{F}} \\ &\le \|\mathcal{A}_k\|_{\mathrm{op}} \|B(\Lambda_1) - B(\Lambda_2)\|_{\mathrm{F}} \\ &\le t\|\mathcal{A}_k\|_{\mathrm{op}}^2 \|\Lambda_1 - \Lambda_2\|_{\mathrm{F}}, \end{aligned} \tag{4.8}$$

where the second inequality holds because the proximal mapping is nonexpansive.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Moreover,

$$\begin{aligned}
&\langle E(\Lambda_1) - E(\Lambda_2), \Lambda_1 - \Lambda_2\rangle \\
&= \langle V(\Lambda_1) - V(\Lambda_2), \mathcal{A}_k^*(\Lambda_1 - \Lambda_2)\rangle \\
&= \frac{1}{t}\langle \mathrm{prox}_{th}(B(\Lambda_1)) - \mathrm{prox}_{th}(B(\Lambda_2)), B(\Lambda_1) - B(\Lambda_2)\rangle \\
&\geq \frac{1}{t}\|\mathrm{prox}_{th}(B(\Lambda_1)) - \mathrm{prox}_{th}(B(\Lambda_2))\|_{\mathrm{F}}^2 \\
&\geq \frac{1}{t\|\mathcal{A}_k\|_{\mathrm{op}}^2}\|E(\Lambda_1) - E(\Lambda_2)\|_{\mathrm{F}}^2 \geq 0,
\end{aligned}$$

where the first inequality holds because the proximal mapping is firmly nonexpansive and the second inequality is due to (4.8). In particular, we see that E is actually $1/(t\|\mathcal{A}_k\|_{\mathrm{op}}^2)$-coercive. Therefore, the operator E is indeed monotone and Lipschitz continuous, and we can apply the SSN method to find a zero of E. In order to apply the SSN method, we need to compute the generalized Jacobian of E.[1] Toward that end, observe that the vectorization of $E(\Lambda)$ can be represented by

$$\begin{aligned}
\mathrm{vec}(E(\Lambda)) &= (X_k^\top \otimes I_p)K_{nr}\mathrm{vec}(V(\Lambda)) + (I_r \otimes X_k^\top)\mathrm{vec}(V(\Lambda)) \\
&= (K_{rr} + I_{r^2})(I_p \otimes X_k^\top)\Big[\mathrm{prox}_{th(\cdot)}(\mathrm{vec}(X_k - t\nabla f(X_k)) \\
&\qquad + 2t(I_r \otimes X_k)\mathrm{vec}(\Lambda)) - \mathrm{vec}(X_k)\Big],
\end{aligned}$$

where K_{nr} and K_{rr} denote the commutation matrices. We define the matrix

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

where $\mathcal{J}(y)$ is the generalized Jacobian of $\mathrm{prox}_{th}(y)$. From [44, Example 2.5], we know that $\mathcal{G}(\mathrm{vec}(\Lambda))\xi = \partial\mathrm{vec}(E(\mathrm{vec}(\Lambda)))\xi$ for all $\xi \in \mathbb{R}^{r^2}$. Thus, $\mathcal{G}(\mathrm{vec}(\Lambda))$ can serve as a representation of $\partial\mathrm{vec}(E(\mathrm{vec}(\Lambda)))$. Note that since Λ is a symmetric matrix, we only need to focus on the lower triangular part of Λ. It is known that there exists a unique $r^2 \times \frac{1}{2}r(r+1)$ matrix U_r, called the duplication matrix [74, Chapter 3.8], such that $U_r\overline{\mathrm{vec}}(\Lambda) = \mathrm{vec}(\Lambda)$. The Moore–Penrose inverse of U_r is $U_r^+ = (U_r^\top U_r)^{-1}U_r^\top$ and satisfies $U_r^+\mathrm{vec}(\Lambda) = \overline{\mathrm{vec}}(\Lambda)$. Note that both U_r and U_r^+ have only r^2 nonzero elements. As a result, we can represent the generalized Jacobian of $\overline{\mathrm{vec}}(E(U_r\overline{\mathrm{vec}}(\Lambda)))$ by

$$\mathcal{G}(\overline{\mathrm{vec}}(\Lambda)) = tU_r^+\mathcal{G}(\mathrm{vec}(\Lambda))U_r = 4tU_r^+(I_r \otimes X_k^\top)\mathcal{J}(y)|_{y=\mathrm{vec}(B(\Lambda))}(I_r \otimes X_k)U_r,$$

where we use the identity $K_{rr} + I_{r^2} = 2U_rU_r^+$. It should be pointed out that $\mathcal{G}(\overline{\mathrm{vec}}(\Lambda))$ can be singular. Therefore, the vanilla SSN method cannot be applied directly, and we need to resort to a regularized SSN method proposed in [88] and further studied in [115, 102]. It is known that the global convergence of the regularized SSN method is guaranteed if any element in $\mathcal{G}(\overline{\mathrm{vec}}(\Lambda))$ is positive semidefinite [102], which is the case here because it can be shown that $\mathcal{G}(\overline{\mathrm{vec}}(\Lambda)) + \mathcal{G}(\overline{\mathrm{vec}}(\Lambda))^\top$ is positive semidefinite. We find that the adaptive regularized SSN (ASSN) method proposed in [102] is very suitable for solving (4.7). The ASSN method first computes the Newton direction d_k by solving

$$(\mathcal{G}(\overline{\mathrm{vec}}(\Lambda_k)) + \eta I)d = -\overline{\mathrm{vec}}(E(\Lambda_k)), \tag{4.9}$$

[1] See Appendix A for a brief discussion of the semismoothness of operators related to the proximal mapping.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

where $\eta > 0$ is a regularization parameter. If the matrix size is large, then (4.9) can be solved inexactly by the conjugate gradient method. The authors of [102] then designed a strategy to decide whether or not to accept this d_k. Roughly speaking, if there is a sufficient decrease from $\|E(\Lambda_k)\|_2$ to $\|E(\Lambda_{k+1})\|_2$, then we accept d^k and set

$$\overline{\mathrm{vec}}(\Lambda_{k+1}) = \overline{\mathrm{vec}}(\Lambda_k) + d_k.$$

Otherwise, a safeguard step is taken. For more details on the ASSN method, we refer the reader to [102].

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). Our convergence analysis consists of three steps. First, in Lemma 5.1 we show that V_k in (4.3) is a descent direction for the objective function in (4.3). Second, in Lemma 5.2 we show that V_k is also a descent direction for the objective function in (1.1) after applying a retraction to it; i.e., there is a sufficient decrease from $F(X_k)$ to $F(\mathrm{Retr}_{X_k}(\alpha V_k))$. This is motivated by a similar result in Boumal, Absil, and Cartis [19], which states that the pullback function $\hat{F}(V) := F(\mathrm{Retr}_X(V))$ satisfies a certain Lipschitz-type property. Therefore, the results here can be seen as an extension of those for smooth problems in [19] to the nonsmooth problem (1.1). Third, we establish the global convergence of ManPG in Theorem 5.5.

Now, let us begin our analysis. The first observation is that the objective function in (4.3) is strongly convex, which implies that the subproblem (4.3) is also strongly convex. Recall that a function g is said to be α-strongly convex[2] on $\mathbb{R}^{n\times r}$ if

$$g(Y) \geq g(X) + \langle \partial g(X), Y - X\rangle + \frac{\alpha}{2}\|Y - X\|_{\mathrm{F}}^2 \quad \forall X, Y \in \mathbb{R}^{n\times r}. \tag{5.1}$$

The following lemma shows that V_k obtained by solving (4.3) is indeed a descent direction in the tangent space to $\mathcal{M}$ at X_k.

LEMMA 5.1. *Given the iterate X_k, let*

$$g(V) := \langle \nabla f(X_k), V\rangle + \frac{1}{2t}\|V\|_F^2 + h(X_k + V) \tag{5.2}$$

denote the objective function in (4.3)*. Then, the following holds for any $\alpha \in [0, 1]$:*

$$g(\alpha V_k) - g(0) \leq \frac{(\alpha - 2)\alpha}{2t}\|V_k\|_F^2. \tag{5.3}$$

Proof. Since g is $(1/t)$-strongly convex, we have

$$g(\hat{V}) \geq g(V) + \langle \partial g(V), \hat{V} - V\rangle + \frac{1}{2t}\|\hat{V} - V\|_{\mathrm{F}}^2 \quad \forall V, \hat{V} \in \mathbb{R}^{n\times r}. \tag{5.4}$$

In particular, if $V, \hat{V}$ are feasible for (4.3) (i.e., $V, \hat{V} \in \mathrm{T}_{X_k}\mathcal{M}$), then

$$\langle \partial g(V), \hat{V} - V\rangle = \langle \mathrm{Proj}_{\mathrm{T}_{X_k}\mathcal{M}} \partial g(V), \hat{V} - V\rangle.$$

[2] A function $g : \mathbb{R}^n \to \mathbb{R}$ is called α-strongly convex [82, Definition 12.58] if there exists a constant $\alpha > 0$ such that $g((1-t)x + ty) \leq (1-t)g(x) + tg(y) - \frac{1}{2}\alpha t(1-t)\|x - y\|^2$ for all x, y when $t \in (0, 1)$. This is equivalent to saying that $g - \frac{1}{2}\alpha\|\cdot\|^2$ is convex [82, Exercise 12.59]. Thus, we have the definition in (5.1).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

From the optimality condition of (4.3), we have $0 \in \mathrm{Proj}_{\mathrm{T}_{X_k}\mathcal{M}} \partial g(V_k)$. Letting $V = V_k$ and $\hat{V} = 0$ in (5.4) yields

$$g(0) \geq g(V_k) + \frac{1}{2t}\|V_k\|_{\mathrm{F}}^2,$$

which implies that

$$h(X_k) \geq \langle \nabla f(X_k), V_k \rangle + \frac{1}{2t}\|V_k\|_{\mathrm{F}}^2 + h(X_k + V_k) + \frac{1}{2t}\|V_k\|_{\mathrm{F}}^2.$$

Moreover, the convexity of h yields

$$h(X_k + \alpha V_k) - h(X_k) = h(\alpha(X_k + V_k) + (1-\alpha)X_k) - h(X_k) \leq \alpha \left(h(X_k + V_k) - h(X_k) \right).$$

Upon combining the above two inequalities, we obtain

$$\begin{aligned} g(\alpha V_k) - g(0) &= \langle \nabla f(X_k), \alpha V_k \rangle + \frac{\|\alpha V_k\|_{\mathrm{F}}^2}{2t} + h(X_k + \alpha V_k) - h(X_k) \\ &\leq \alpha \left(\langle \nabla f(X_k), V_k \rangle + \alpha \frac{\|V_k\|_{\mathrm{F}}^2}{2t} + h(X_k + V_k) - h(X_k) \right) \\ &\leq \frac{\alpha^2 - 2\alpha}{2t}\|V_k\|_{\mathrm{F}}^2, \end{aligned}$$

as desired. ∎

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

LEMMA 5.2. *For any $t > 0$, there exists a constant $\bar{\alpha} > 0$ such that for any $0 < \alpha \leq \min\{1, \bar{\alpha}\}$, the condition in step 5 of Algorithm 1 is satisfied and the sequence $\{X_k\}$ generated by Algorithm 1 satisfies*

$$F(X_{k+1}) - F(X_k) \leq -\frac{\alpha}{2t}\|V_k\|_F^2.$$

Proof. Let $X_k^+ = X_k + \alpha V_k$. Following Boumal, Absil, and Cartis [19], we first show that $f(\mathrm{Retr}_{X_k}(V))$ satisfies a certain Lipschitz smoothness condition. By the L-Lipschitz continuity of ∇f, for any $\alpha > 0$, we have

(5.5)
$$\begin{aligned} f(\mathrm{Retr}_{X_k}(\alpha V_k)) - f(X_k) &\leq \langle \nabla f(X_k), \mathrm{Retr}_{X_k}(\alpha V_k) - X_k \rangle + \frac{L}{2}\|\mathrm{Retr}_{X_k}(\alpha V_k) - X_k\|_{\mathrm{F}}^2 \\ &= \langle \nabla f(X_k), \mathrm{Retr}_{X_k}(\alpha V_k) - X_k^+ + X_k^+ - X_k \rangle + \frac{L}{2}\|\mathrm{Retr}_{X_k}(\alpha V_k) - X_k\|_{\mathrm{F}}^2 \\ &\leq M_2 \|\nabla f(X_k)\|_{\mathrm{F}} \|\alpha V_k\|_{\mathrm{F}}^2 + \alpha \langle \nabla f(X_k), V_k \rangle + \frac{L M_1^2}{2}\|\alpha V_k\|_{\mathrm{F}}^2, \end{aligned}$$

where the last inequality follows from (3.1) and (3.2). Since ∇f is continuous on the compact manifold $\mathcal{M}$, there exists a constant $G > 0$ such that $\|\nabla f(X)\|_{\mathrm{F}} \leq G$ for all $X \in \mathcal{M}$. It then follows from (5.5) that

$$f(\mathrm{Retr}_{X_k}(\alpha V_k)) - f(X_k) \leq \alpha \langle \nabla f(X_k), V_k \rangle + c_0 \alpha^2 \|V_k\|_{\mathrm{F}}^2, \tag{5.6}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

where $c_0 = M_2 G + LM_1^2/2$. This implies that

$$\begin{aligned}
&F(\mathrm{Retr}_{X_k}(\alpha V_k)) - F(X_k) \\
&\overset{(5.6)}{\leq} \alpha\langle \nabla f(X_k), V_k\rangle + c_0\alpha^2\|V_k\|_{\mathrm{F}}^2 + h(\mathrm{Retr}_{X_k}(\alpha V_k)) - h(X_k^+) + h(X_k^+) - h(X_k) \\
&\leq \alpha\langle \nabla f(X_k), V_k\rangle + c_0\alpha^2\|V_k\|_F^2 + L_h\|\mathrm{Retr}_{X_k}(\alpha V_k) - X_k^+\|_{\mathrm{F}} + h(X_k^+) - h(X_k) \\
&\overset{(3.2)}{\leq} (c_0 + L_h M_2)\|\alpha V_k\|_F^2 + g(\alpha V_k) - \frac{1}{2t}\|\alpha V_k\|_{\mathrm{F}}^2 - g(0) \\
&\overset{(5.3)}{\leq} \left(c_0 + L_h M_2 - \frac{1}{\alpha t}\right)\|\alpha V_k\|_{\mathrm{F}}^2,
\end{aligned}$$

where g is defined in (5.2) and the second inequality follows from the Lipschitz continuity of h. Upon setting $\bar{\alpha} = 1/(2(c_0 + L_h M_2)t)$, we conclude that for any $0 < \alpha \leq \min\{\bar{\alpha}, 1\}$,

$$F(\mathrm{Retr}_{X_k}(\alpha V_k)) - F(X_k) \leq -\frac{1}{2\alpha t}\|\alpha V_k\|_{\mathrm{F}}^2 = -\frac{\alpha}{2t}\|V_k\|_{\mathrm{F}}^2.$$

This completes the proof. ∎

The following lemma shows that if one cannot make any progress by solving (4.3) (i.e., $V_k = 0$), then a stationary point is found.

LEMMA 5.3. *If $V_k = 0$, then X_k is a stationary point of problem* (1.1).

Proof. By Theorem 4.1 in [106], the optimality conditions of the subproblem (4.2) are given by

$$0 \in \frac{1}{t}V_k + \mathrm{grad}\, f(X_k) + \mathrm{Proj}_{\mathrm{T}_{X_k}\mathcal{M}}\partial h(X_k + V_k), \qquad V_k \in \mathrm{T}_{X_k}\mathcal{M}.$$

If $V_k = 0$, then $0 \in \mathrm{grad}\, f(X_k) + \mathrm{Proj}_{\mathrm{T}_{X_k}\mathcal{M}}\partial h(X_k)$, which is exactly the first-order necessary condition of problem (1.1) since $X_k \in \mathcal{M}$. ∎

From Lemma 5.3, we know that $V_k = 0$ implies the stationarity of X_k with respect to (1.1). This motivates the following definition of an ϵ-stationary point of (1.1).

DEFINITION 5.4. *We say that $X_k \in \mathcal{M}$ is an ϵ-stationary point of* (1.1) *if the solution V_k to* (4.4) *with $t = 1/L$ satisfies $\|V_k\|_F \leq \epsilon/L$.*

We use $\|V_k\|_{\mathrm{F}} \leq \epsilon/L$ as the stopping criterion of Algorithm 1 with $t = 1/L$. From Lemma 5.2, we obtain the following result, which is similar to that in [19, Theorem 2] for manifold optimization with smooth objective functions.

THEOREM 5.5. *Under Assumption* 1.1, *every limit point of the sequence $\{X_k\}$ generated by Algorithm* 1 *is a stationary point of problem* (1.1). *Moreover, Algorithm* 1 *with $t = 1/L$ will return an ϵ-stationary point of* (1.1) *in at most $\lceil 2L(F(X_0) - F^*)/(\gamma\bar{\alpha}\epsilon^2)\rceil$ iterations, where $\bar{\alpha}$ is defined in Lemma* 5.2 *and F^* is the optimal value of* (1.1).

Proof. Since F is bounded below on $\mathcal{M}$, by Lemma 5.2 we have

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

Combining this with Lemma 5.3, it follows that every limit point of $\{X_k\}$ is a stationary point of (1.1). Moreover, since $\mathcal{M}$ is compact, the sequence $\{X_k\}$ has at least one

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

limit point. Furthermore, suppose that Algorithm 1 with $t = 1/L$ does not terminate after K iterations; i.e., $\|V_k\|_{\mathrm{F}} > \epsilon/L$ for $k = 0, 1, \ldots, K-1$. Let α_k be the stepsize in the kth iteration; i.e., $X_{k+1} = \mathrm{Retr}_{X_k}(\alpha_k V_k)$. From Lemma 5.2, we know that $\alpha_k \geq \gamma\bar{\alpha}$. Thus, we have

$$F(X_0) - F^* \geq F(X_0) - F(X_K) \geq \frac{t}{2}\sum_{k=0}^{K-1}\alpha_k\|V_k/t\|_{\mathrm{F}}^2 > \frac{t\epsilon^2}{2}\sum_{k=0}^{K-1}\alpha_k \geq \frac{tK\epsilon^2}{2}\gamma\bar{\alpha}.$$

Therefore, Algorithm 1 finds an ϵ-stationary point in at most $\lceil 2L(F(X_0)-F^*)/(\gamma\bar{\alpha}\epsilon^2)\rceil$ iterations. ∎

Remark 5.6. When the objective function F in (1.1) is smooth (i.e., the nonsmooth function h vanishes), the iteration complexity in Theorem 5.5 matches the result given by Boumal, Absil, and Cartis in [19]. Zhang and Sra [111] analyzed the iteration complexity of some first-order methods, but they assumed that the objective function is geodesically convex. Such an assumption is rather restrictive, as every smooth function that is geodesically convex on a compact Riemannian manifold is constant [13]. Bento, Ferreira, and Melo [12] also established some iteration complexity results for gradient, subgradient, and proximal point methods. However, their results for gradient and subgradient methods require the objective function to be convex and the manifold to be of nonnegative curvature, while those for proximal point methods only apply to a convex objective function over the Hadamard manifold.

6. Numerical Experiments.

In this section, we apply our ManPG algorithm[3] (Algorithm 1) to solve the sparse PCA (1.2) and compressed modes (CM) (1.3) problems. We compare ManPG with two existing methods: SOC [58] and PAMAL [24]. For both problems, we set the parameter $\gamma = 0.5$ and use the polar decomposition as the retraction mapping in ManPG. The latter is because it was found that the MATLAB implementation of QR factorization is slower than the polar decomposition; see [5]. Moreover, we implement a more practical version of ManPG, named ManPG-Ada and described in Algorithm 2, which incorporates a few tricks, including adaptively updating the stepsize t. We set the parameters $\gamma = 0.5$ and $\tau = 1.01$ in ManPG-Ada. All the codes used in this section were written in MATLAB and run on a standard PC with 3.70 GHz I7 Intel microprocessor and 16GB of memory.

6.1. A More Practical ManPG: ManPG-Ada.

In this subsection, we introduce some tricks used to further improve the performance of ManPG in practice. First, a warm-start strategy is adopted for SSN; i.e., the initial point Λ_0 in SSN is set as the solution of the previous subproblem. For the ASSN algorithm, we always take the SSN step as suggested by [102]. Second, we adaptively update t in ManPG. When t is large, we may need a smaller total number of iterations to reach an ϵ-stationary point. However, it increases the number of line-search steps and SSN steps. For sparse PCA and CM problems, we found that setting $t = 1/L$ leads to fewer line-search steps. We can then increase t slightly if no line-search step was needed in the previous iteration. This new version of ManPG, named ManPG-Ada, is described in Algorithm 2. We also applied ManPG-Ada to solve sparse PCA and CM problems and compared its performance with those of ManPG, SOC, and PAMAL.

[3]Our MATLAB code is available at https://github.com/chenshixiang/ManPG.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Algorithm 2 ManPG-Ada for solving (1.1).

Input: initial point $X_0 \in \mathcal{M}$, $\gamma \in (0,1)$, $\tau > 1$, Lipschitz constant L
set $t = 1/L$
for $k = 0, 1, \ldots$ **do**
  obtain V_k by solving the subproblem (4.3)
  set $\alpha = 1$ and linesearchflag $= 0$
  while $F(\mathrm{Retr}_{X_k}(\alpha V_k)) > F(X_k) - \dfrac{\alpha \|V_k\|_{\mathrm{F}}^2}{2t}$ **do**
    $\alpha = \gamma\alpha$
    linesearchflag $= 1$
  end while
  set $X_{k+1} = \mathrm{Retr}_{X_k}(\alpha V_k)$
  if linesearchflag $= 1$ **then**
    $t = \tau t$
  else
    $t = \max\{1/L, t/\tau\}$
  end if
end for

6.2. Numerical Results on CM. For the CM problem (1.3), both SOC [58] and PAMAL [24] rewrite the problem as

$$\begin{array}{ll} \min\limits_{X,Q,P\in\mathbb{R}^{n\times r}} & \mathrm{Tr}(P^\top HP) + \mu\|Q\|_1 \\ \text{s.t.} & Q = P, X = P, X^\top X = I_r. \end{array} \tag{6.1}$$

SOC employs a three-block ADMM to solve (6.1), which updates the iterates as follows:

$$\begin{aligned} P_{k+1} &:= \operatorname*{argmin}_P \ \mathrm{Tr}(P^\top HP) + \frac{\beta}{2}\|P - Q_k + \Lambda_k\|_F^2 + \frac{\beta}{2}\|P - X_k + \Gamma_k\|_F^2, \\ Q_{k+1} &:= \operatorname*{argmin}_Q \ \mu\|Q\|_1 + \frac{\beta}{2}\|P_{k+1} - Q + \Lambda_k\|_F^2, \\ X_{k+1} &:= \operatorname*{argmin}_X \ \frac{\beta}{2}\|P_{k+1} - X + \Gamma_k\|_F^2 \ \text{ s.t. } X^\top X = I_r, \\ \Lambda_{k+1} &:= \Lambda_k + P_{k+1} - Q_{k+1}, \\ \Gamma_{k+1} &:= \Gamma_k + P_{k+1} - X_{k+1}. \end{aligned} \tag{6.2}$$

PAMAL uses an inexact augmented Lagrangian method to solve (6.1), with the augmented Lagrangian function being minimized by the proximal alternating minimization algorithm proposed in [7]. Both SOC and PAMAL need to solve a linear system $(H + \beta I)X = B$, where B is a given matrix.

In our numerical experiments, we tested the same problems as in [78] and [24]. In particular, we considered the time-independent Schrödinger equation

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

where $\hat{H} = -\frac{1}{2}\Delta$ denotes the Hamiltonian, Δ denotes the Laplacian operator, and H is a symmetric matrix formed by discretizing the Hamiltonian $\hat{H}$. We focused 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 considered

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

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 $\hat{H}$ and is given by $2n^2/50^2$ in this case.

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 $\|V_k/t\|_{\mathrm{F}}^2 \le \epsilon := 10^{-8}nr$ or the maximum iteration number 30000 was reached. For the inner iteration of ManPG (i.e., using SSN to solve (4.3)), we terminated it when $\|E(\Lambda)\|_{\mathrm{F}}^2 \le \max\{10^{-13}, \min\{10^{-11}, 10^{-3}t^2\epsilon\}\}$ or the maximum iteration number 100 was reached. In all the tests of the CM problem, we ran ManPG first and let F_M denote the returned objective value. We then ran SOC and PAMAL and terminated them when $F(X_k) \le F_M + 10^{-7}$ and

$$\frac{\|Q_k - P_k\|_F}{\max\{1, \|Q_k\|_F, \|P_k\|_F\}} + \frac{\|X_k - P_k\|_F}{\max\{1, \|X_k\|_F, \|P_k\|_F\}} \le 10^{-4}. \tag{6.3}$$

Note that (6.3) measures the constraint violation of the reformulation (6.1). If (6.3) was not satisfied in 30000 iterations, then we terminated SOC and PAMAL. We also ran ManPG-Ada (Algorithm 2) and terminated it if $F(X_k) \le F_M + 10^{-7}$.

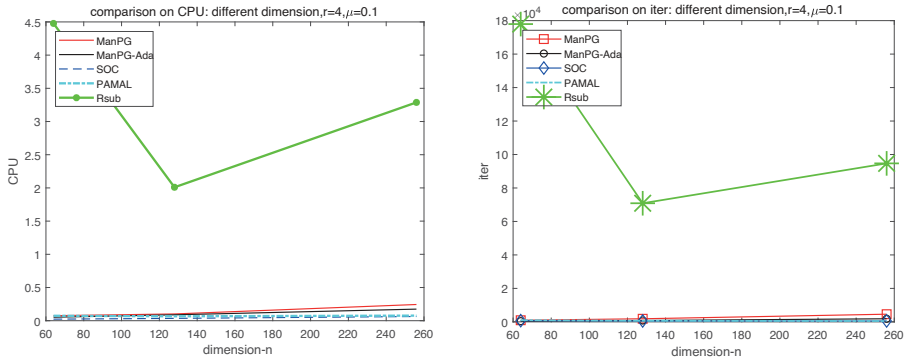
In our experiments, we found that SOC and PAMAL are very sensitive to the choice of parameters. The default setting of the parameters of SOC and PAMAL suggested in [78] and [24] usually cannot achieve our desired accuracy. Unfortunately, there is no systematic study on how to tune these parameters. We spent a significant amount of effort on tuning these parameters, and the ones we used are given as follows. For SOC (6.2), we set the penalty parameter $\beta = nr\mu/25 + 1$. For PAMAL, we found that the setting of the parameters given on page B587 of [24] did not work well for the problems we tested. Instead, we found that the following settings of these parameters worked best and were thus adopted in our tests: $\tau = 0.99$, $\gamma = 1.001$, $\rho^1 = 2\,|\lambda_{\min}(H)| + r/10 + 2$, $\overline{\Lambda}_{p,\min} = -100$, $\overline{\Lambda}_{p,\max} = 100$, $\Lambda_p^1 = 0_{nr}$, $p = 1, 2$, and $\epsilon^k = (0.995)^k$, $k \in \mathbb{N}$. For the meaning of these parameters, we refer the reader to page B587 of [24]. We used the same parameters of PAM in PAMAL as recommended by [24]. For different settings of (n, r, μ), we ran the four algorithms on 50 instances whose initial points were obtained by projecting randomly generated points onto $\mathrm{St}(n, r)$. Since problem (1.3) is nonconvex, it is possible that ManPG, ManPG-Ada, SOC, and PAMAL return different solutions from random initializations. To increase the chance that all four solvers found the same solution, we ran the Riemannian subgradient method for 500 iterations and used the resulting iterate as the initial point. The Riemannian subgradient method is described as follows:

$$\begin{aligned} \hat{\partial} F(X_k) &:= \mathrm{Proj}_{\mathrm{T}_{X_k}\mathrm{St}(n,r)}(2HX_k + \mu\,\mathrm{sign}(X_k)), \\ X_{k+1} &:= \mathrm{Retr}_{X_k}\left(-\frac{1}{k^{3/4}}\hat{\partial} F(X_k)\right), \end{aligned} \tag{6.4}$$

where $\mathrm{sign}(\cdot)$ denotes the elementwise sign function. Moreover, we tried to run the Riemannian subgradient method (6.4) until it solved the CM problem. However, this method is extremely slow and we only report one case in Figure 1. We report the average CPU time, iteration number, and sparsity in Figures 1 to 4, where sparsity is the percentage of zeros; when computing sparsity, X is truncated by zeroing out its entries with magnitude smaller than 10^{-5}. For SOC and PAMAL, we only took into account the solutions that were close to the one generated by ManPG.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



(a) CPU (b) Iteration

Fig. 1 *Comparison for CM problem* (1.3)*, different* $n = \{64, 128, 256\}$ *with* $r = 4$ *and* $\mu = 0.1$.

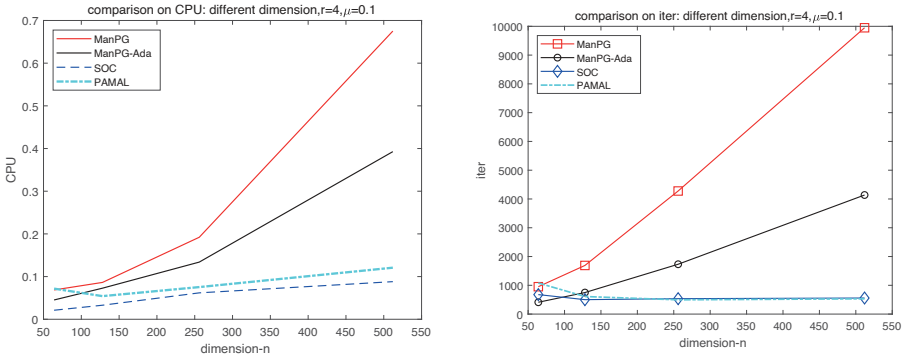
Here, the closeness of the solutions is measured by the distance between their column spaces. More specifically, let X_M, X_S, and X_P denote the solutions generated by ManPG, SOC, and PAMAL, respectively. Then their distances are computed by $\text{dist}(X_M, X_S) = \|X_M X_M^\top - X_S X_S^\top\|_{\text{F}}$ and $\text{dist}(X_M, X_P) = \|X_M X_M^\top - X_P X_P^\top\|_{\text{F}}$. We only counted the results if $\text{dist}^2(X_M, X_S) \leq 0.1$ and $\text{dist}^2(X_M, X_P) \leq 0.1$.

In Figure 1, we report the results of the Riemannian subgradient method with respect to different n's. We terminated the Riemannian subgradient method (6.4) if $F(X_k) < F_M + 10^{-3}$. We see that this accuracy tolerance 10^{-3} is too large to yield a good solution with reasonable sparsity level, yet it is already very time consuming. As a result, we do not report more results on the Riemannian subgradient method. In Figures 2, 3, and 4, we see that the solutions returned by ManPG and ManPG-Ada have better sparsity than SOC and PAMAL. We also see that ManPG-Ada outperforms ManPG in terms of CPU time and iteration number. In Figure 2, the iteration number of ManPG increases with the dimension n, because the Lipschitz constant $L = 2\lambda_{\max}(H) = 4n^2/50^2$ increases quadratically, which is consistent with our complexity result. In Figure 3, we see that the CPU times of ManPG and ManPG-Ada are comparable to those of SOC and PAMAL when r is small, but are slightly higher when r becomes large. In Figure 4, we see that the performance of the algorithms is also affected by μ. In terms of CPU time, ManPG and ManPG-Ada are comparable to SOC and PAMAL when μ becomes large.

The first five CM of the 1D FE model computed by the ManPG-Ada, SOC, and PAMAL methods are shown in Figure 5. We found that the CM generated by ManPG and ManPG-Ada were the same, so we only report the results of ManPG-Ada. We flip the CM if necessary so that most values on the support of the CM are positive, as sign ambiguities do not affect the minimal values of the objective function in (1.3). It can be seen that the CM obtained from the three methods are compactly supported functions, and their localization degree is almost the same. We next examine the approximation behavior of the unitary transformations derived from the CM to the eigenmodes of the Schrödinger operator. The approximation accuracy is measured by comparing the first r eigenvalues $(\sigma_1, \ldots, \sigma_r)$ of the matrix $X^\top \hat{H} X$ with the first r eigenvalues $(\lambda_1, \ldots, \lambda_r)$ of the corresponding Schrödinger operator $\hat{H}$. Figure 6 reports the results for different values of r. We can see that the approximation errors of the ManPG-Ada, SOC, and PAMAL methods are similar, and that $(\sigma_1, \ldots, \sigma_r)$ converges to $(\lambda_1, \ldots, \lambda_r)$ as r increases.

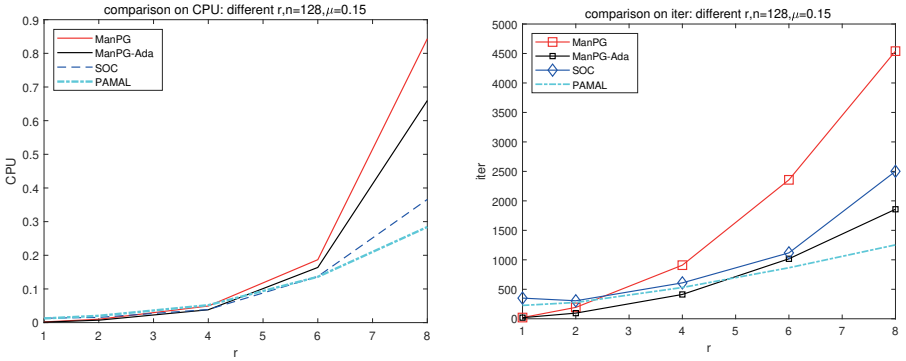
Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.



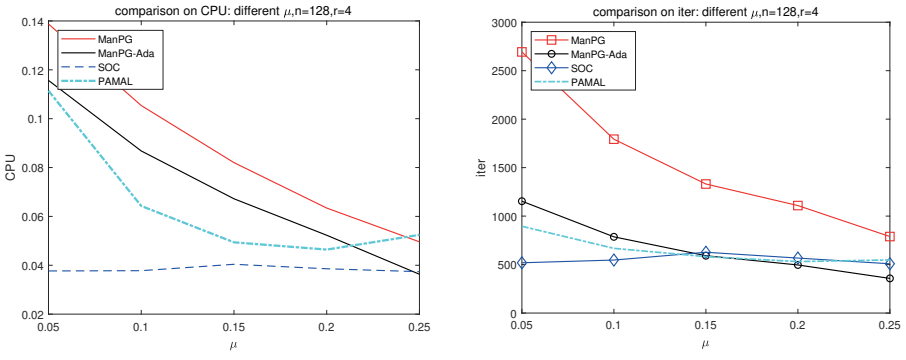
(a) CPU (b) Iteration

Fig. 2 *Comparison for CM problem* (1.3)*, different* $n = \{64, 128, 256, 512\}$ *with* $r = 4$ *and* $\mu = 0.1$.



(a) CPU (b) Iteration

Fig. 3 *Comparison for CM problem* (1.3)*, different* $r = \{1, 2, 4, 6, 8\}$ *with* $n = 128$ *and* $\mu = 0.15$.



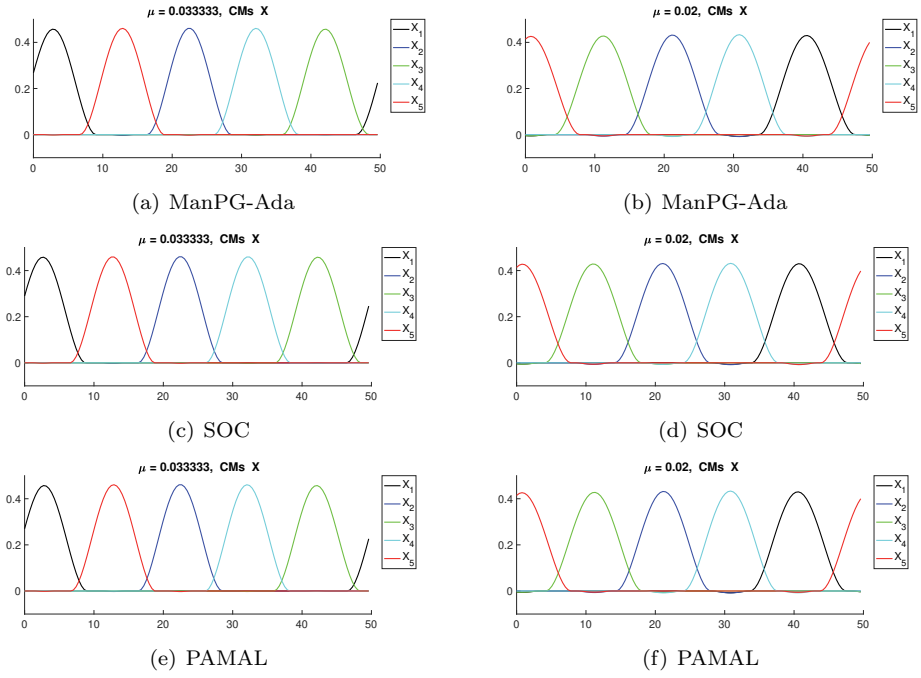
(a) CPU (b) Iteration

Fig. 4 *Comparison for CM problem* (1.3)*, different* $\mu = \{0.05, 0.1, 0.15, 0.2, 0.25\}$ *with* $n = 128$ *and* $r = 4$.

We also report the total number of line-search steps and the averaged iteration number of SSN in ManPG and ManPG-Ada in Table 1. We see that ManPG-Ada needs more line-search steps and SSN iterations, but as we show in Figures 2, 3, and 4, ManPG-Ada is faster than ManPG in terms of CPU time. This is mainly because the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

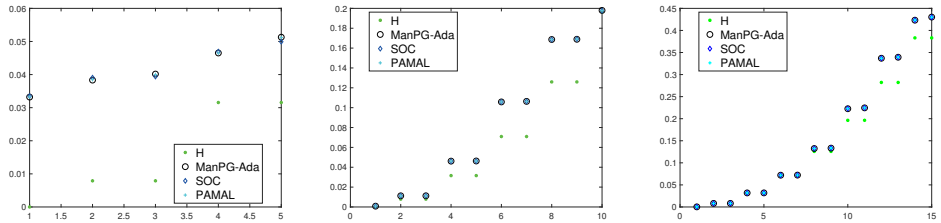


(a) ManPG-Ada (b) ManPG-Ada

(c) SOC (d) SOC

(e) PAMAL (f) PAMAL

Fig. 5 *Comparison of the first five modes obtained for the 1D FE model with different values of μ. Left column: $\mu = 1/30$. Right column: $\mu = 1/50$.*



(a) $r = 5$ (b) $r = 10$ (c) $r = 15$

Fig. 6 *Comparisons of the first r eigenvalues of the 1D FE model. $*$: the first r eigenvalues of the matrix $\hat{H}$. $\circ$: the first r eigenvalues of the matrix $X^\top \hat{H} X$, where X is the solution obtained by ManPG-Ada. $\diamond$: the first r eigenvalues of the matrix $X^\top \hat{H} X$, where X is the solution obtained by SOC. $+$: the first r eigenvalues of the matrix $X^\top \hat{H} X$, where X is the solution obtained by PAMAL.*

computational costs of retraction and SSN steps in this problem are both nearly the same as computing the gradient. In the last two columns of Table 1, "#s|d" denotes the number of instances for which SOC and PAMAL generate the same/ different solutions as ManPG with the closeness measurement discussed above; "# f" denotes the number of instances in which SOC and PAMAL fail to converge. We see that for the tested instances of the CM problem, all algorithms converged thanks to the parameters that we chose, although sometimes the solutions generated by PAMAL are different from those generated by ManPG and SOC.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Table 1 *Number of line-search steps and average number of SSN iterations for different* (n, r, μ).

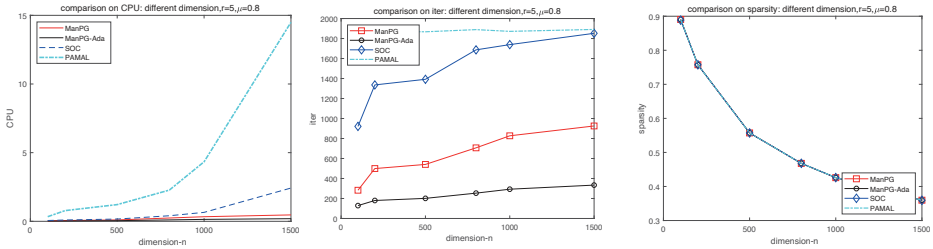
	ManPG		ManPG-Ada		SOC	PAMAL
	# line-search	SSN iter	# line-search	SSN iter	# s| d| f	# s| d| f
n	$r = 4, \mu = 0.1$					
64	85.94	1.0005	165.98	1.3307	50|0|0	48|2|0
128	70.5	0.64414	540.76	1.2237	50|0|0	50|0|0
256	84.06	0.39686	1191.5	0.60652	50|0|0	50|0|0
512	55.1	0.16622	2720.6	0.2417	50|0|0	49|1|0
μ	$n = 128, r = 4$					
0.05	49.2	0.30933	695.6	0.83637	50|0|0	50|0|0
0.1	74.38	0.54915	572.42	1.1514	50|0|0	50|0|0
0.15	102.62	0.82093	439.6	1.2899	50|0|0	50|0|0
0.2	82.52	0.81565	350.86	1.2114	50|0|0	50|0|0
0.25	93.3	0.57232	209.12	1.0122	50|0|0	48|2|0
r	$n = 128, \mu = 0.15$					
1	0	0.8971	0	0.98694	50|0|0	50|0|0
2	3.48	1.0001	61.02	1.1135	50|0|0	50|0|0
4	86.92	0.91814	311	1.2812	50|0|0	50|0|0
6	169.8	0.60206	719.42	1.5195	50|0|0	49|1|0
8	216.54	1.2011	1198.8	2.8667	50|0|0	42|8|0

6.3. Numerical Results on Sparse PCA. In this section, we compare the performance of ManPG, ManPG-Ada, SOC, and PAMAL for solving the sparse PCA problem (1.2). Note that there are other algorithms for sparse PCA such as those proposed in [55, 27], but these methods work only for the special case when $r = 1$, i.e., the constraint set is a sphere. The algorithm proposed in [37] needs to smooth the ℓ_1 norm in order to apply existing gradient-type methods, and thus the sparsity of the solution is no longer guaranteed. Algorithms proposed in [118, 86, 56] 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.

The random data matrices $A \in \mathbb{R}^{m\times n}$ considered in this section were generated in the following manner. We first generate a random matrix using the MATLAB function $A = \text{randn}(m, n)$, then shift the columns of A so that their mean is equal to 0, and lastly normalize the columns so that their Euclidean norms are equal to one. In all tests, m is equal to 50. The Lipschitz constant L is $2\sigma^2_{\max}(A)$, so we use $t = 1/(2\sigma^2_{\max}(A))$ in Algorithms 1 and 2, where $\sigma_{\max}(A)$ is the largest singular value of A. Again, we spent a lot of effort in fine-tuning the parameters for SOC and PAMAL and found that the following settings of the parameters worked best for our tested problems. For SOC, we set the penalty parameter $\beta = 2\sigma^2_{\max}(A)$. For PAMAL, we set $\tau = 0.99$, $\gamma = 1.001$, $\rho^1 = 5\sigma^2_{\max}(A)$, $\overline{\Lambda}_{p,\min} = -100$, $\overline{\Lambda}_{p,\max} = 100$, $\Lambda^1_p = 0_{nr}, p = 1, 2$, and $\epsilon^k = (0.996)^k$, $k \in \mathbb{N}$. We again refer the reader to page B587 of [24] for the meanings of these parameters. We used the same parameters of PAM in PAMAL as suggested in [24]. We used the same stopping criterion for ManPG, ManPG-Ada, SOC, and PAMAL as for the CM problems. For different settings of (n, r, μ), we ran the four algorithms with 50 instances whose initial points were obtained by projecting randomly generated points onto $\text{St}(n, r)$. We then ran the Riemannian subgradient method (6.4) for 500 iterations and used the returned solution as the initial point of the compared solvers.

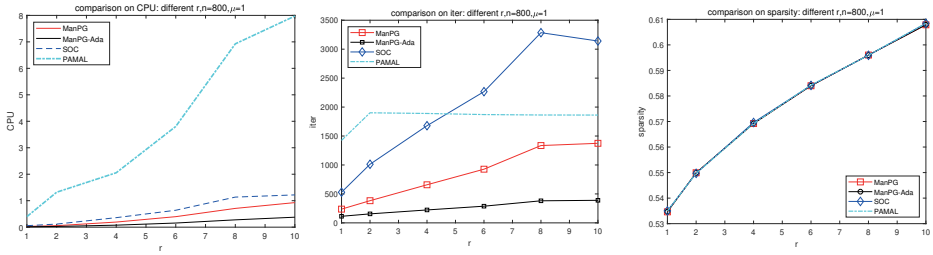
Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



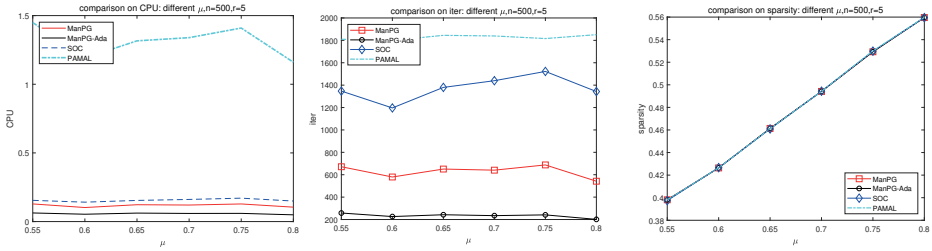
(a) CPU (b) Iteration (c) Sparsity

Fig. 7 *Comparison for sparse PCA problem* (1.2)*, different* $n = \{100, 200, 500, 800, 1000, 1500\}$ *with* $r = 5$ *and* $\mu = 0.8$.



(a) CPU (b) Iteration (c) Sparsity

Fig. 8 *Comparison for sparse PCA problem* (1.2)*, different* $r = \{1, 2, 4, 6, 8, 10\}$ *with* $n = 800$ *and* $\mu = 1$.



(a) CPU (b) Iteration (c) Sparsity

Fig. 9 *Comparison for sparse PCA problem* (1.2)*, different* $\mu = \{0.55, 0.6, 0.65, 0.7, 0.75, 0.8\}$ *with* $n = 500$ *and* $r = 5$.

The CPU time, iteration number, and sparsity are reported in Figures 7, 8, and 9. As with the CM problem, all the values were averaged over those instances that yielded solutions that were close to those given by ManPG. In Figures 7, 8, and 9, we see that ManPG and ManPG-Ada significantly outperformed SOC and PAMAL in terms of the CPU time required to obtain the same solutions. We also see that ManPG-Ada greatly improved the performance of ManPG. We also report the total number of line-search steps and the average iteration number of SSN in ManPG and ManPG-Ada in Table 2. We observe from Table 2 that SOC failed to converge on one instance, and for several instances SOC and PAMAL generated different solutions when compared to those generated by ManPG.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Table 2 *Sparse PCA: Number of line-search steps and average number of SSN iterations for different* (n, r, μ).

	ManPG		ManPG-Ada		SOC	PAMAL
	# line-search	SSN iter	# line-search	SSN iter	# s| d| f	# s| d| f
n	$r = 5, \mu = 0.8$					
100	0.8	1.1881	0.08	1.5221	46|3|1	50|0|0
200	2.98	1.0722	15.1	1.3705	48|2|0	48|2|0
500	0.4	1.025	29.4	1.2066	50|0|0	50|0|0
800	0	1.0167	59.36	1.1847	49|1|0	50|0|0
1000	3.08	1.016	82.04	1.1712	49|1|0	49|1|0
15	11	1.0121	108.94	1.1035	48|2|0	49|1|0
μ	$n = 500, r = 5$					
0.55	0	1.0155	68.7	1.1463	48|2|0	50|0|0
0.60	0	1.0197	48.82	1.1431	50|0|0	49|1|0
0.65	0	1.019	57.96	1.1841	48|2|0	48|2|0
0.70	0	1.0246	52.5	1.2098	49|1|0	50|0|0
0.75	0.36	1.0238	55.88	1.2252	48|2|0	49|1|0
0.80	0	1.0286	28.98	1.1966	49|1|0	49|1|0
r	$n = 800, \mu = 0.6$					
1	0	0.90182	4.12	1.0335	50|0|0	50|0|0
2	82.06	1.0041	10.74	1.0767	49|1|0	50|0|0
4	8.52	1.0229	39.04	1.1453	48|2|0	50|0|0
6	0	1.0243	72.22	1.3198	46|4|0	49|1|0
8	0.34	1.0309	125.64	1.5325	46|4|0	50|0|0
10	0.76	1.0579	132.58	1.6894	42|8|0	47|3|0

7. Subsequent Developments. In this section, we discuss some recent advances in algorithms for Riemannian optimization with nonsmooth objective functions that are mostly inspired by ManPG.

7.1. Manifold Proximal Point Algorithm. An immediate extension of ManPG is the manifold proximal point algorithm (ManPPA), which is studied by Chen et al. in [22]. In particular, the authors focused on two representative applications of Riemannian optimization with nonsmooth objective: orthogonal dictionary learning and robust subspace recovery. Both problems take the following form, which minimizes a nonsmooth function over the Stiefel manifold [61, 60, 63, 62, 59, 92, 117]:

$$\min_X \ h(X) := \|Y^\top X\|_1 \quad \text{s.t.} \quad X \in \mathrm{St}(n, r). \tag{7.1}$$

Note that (7.1) is a special case of (1.1), where the smooth function f vanishes. Therefore, ManPG can be naturally applied to solve (7.1) and, interestingly, ManPG becomes a Riemannian counterpart of the PPA in this case. The authors thus named it ManPPA in [22]. It is proved in [22] that if the problem instance has the sharpness property, then the local convergence rate of ManPPA is at least quadratic. Key to their proof of the quadratic rate result is a new Riemannian subgradient inequality (see also [66]), which can be of independent interest. A stochastic ManPPA is also proposed in [22] to tackle problems with larger size.

7.2. Manifold Proximal Linear Algorithm. In a similar vein as ManPG and ManPPA, a manifold proximal linear algorithm (ManPL) is proposed in [100] to solve

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

the following problem:

$$\min_X \ f(X) + h(c(X)) \quad \text{s.t.} \quad X \in \mathcal{M}. \tag{7.2}$$

Here, f and h satisfy Assumption 1.1, and c is a smooth mapping. A typical iteration of the ManPL algorithm for solving (7.2) is given by

$$V_k := \operatorname*{argmin}_V \ \langle \nabla f(X_k), V\rangle + h(c(X_k) + J(X_k)V) + \frac{1}{2t}\|V\|_F^2 \ \text{ s.t. } \ V \in \mathrm{T}_{X_k}\mathcal{M},$$
$$X_{k+1} := \mathrm{Retr}_{X_k}(\alpha_k V_k),$$

where $J(X) = \nabla c(X)$ is the Jacobian of c and $t > 0$ is a stepsize. Note that the subproblem for updating V_k is convex and thus can be solved by the SSN method in a similar manner as the ManPG subproblem (4.3). The iteration complexity of ManPL is also established in [100].

7.3. Stochastic ManPG. Wang, Ma, and Xue [95] proposed the stochastic counterpart of ManPG, which solves (1.1) when the smooth function f takes one of the following forms:

$$f(X) = \mathbb{E}_\pi[f(X;\pi)], \qquad \text{(Online Case)}$$
$$f(X) = \frac{1}{m}\sum_{i=1}^m f_i(X). \qquad \text{(Finite-Sum Case)}$$

Here, π is a random variable and $\mathbb{E}_\pi$ is the expectation with respect to the distribution of π. The stochastic ManPG generates the iterates via

$$V_k := \operatorname*{argmin}_V \langle g_k, V\rangle + \frac{1}{2t}\|V\|_F^2 + h(X_k + V) \ \text{ s.t. } \ V \in \mathrm{T}_{X_k}\mathcal{M},$$
$$X_{k+1} := \mathrm{Retr}_{X_k}(\alpha_k V_k),$$

where g_k is a stochastic gradient of f. The authors of [95] discussed two different choices of g_k. One is a minibatch stochastic gradient estimator, which gives rise to the R-ProxSGD algorithm. The other is the SpiderBoost [32, 99] gradient estimator, which gives rise to the R-ProxSPB algorithm. They then established the first-order oracle complexities of these two algorithms.

7.4. Zeroth-Order ManPG. Using the Gaussian smoothing technique [76], the work [64] develops the zeroth-order ManPG (ZO-ManPG) algorithm, which estimates the gradient of the objective function from its zeroth-order information. Specifically, suppose that the smooth function f in (1.1) takes the form

$$f(x) := \int_\xi \bar{F}(x,\xi)dP(\xi),$$

where P is a probability distribution. In the ZO-ManPG algorithm, the gradient $\nabla f(X_k)$ in (4.3) is replaced by the zeroth-order gradient estimator

$$\bar{g}_{\mu,\xi}(x) = \frac{1}{m}\sum_{i=1}^m g_{\mu,\xi_i}(x),$$

where

$$g_{\mu,\xi_i}(x) = \frac{\bar{F}(\mathrm{Retr}_x(\mu u_i),\xi_i) - \bar{F}(x,\xi_i)}{\mu}u_i,$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

$\mu > 0$ is the smoothing parameter, and u_i is a standard normal random vector on $\mathrm{T}_x\mathcal{M}$. The zeroth-order oracle complexity of the proposed ZO-ManPG algorithm is established in [64].

7.5. Riemannian Proximal Gradient Method. It is not hard to see that most results in this paper directly apply to (1.1) when the Stiefel manifold is replaced by an embedded submanifold. A natural question then is whether ManPG can be extended to solve (1.1) when $\mathcal{M}$ is a general Riemannian manifold. This question was studied by Huang and Wei in [52]. In particular, the Riemannian proximal gradient (RPG) method proposed in [52] replaces (4.2) in ManPG by the following update: Find $V_k^* \in \mathrm{T}_{X_k}\mathcal{M}$ such that V_k^* is a stationary point of $\ell_{X_k}(V)$ on $\mathrm{T}_{X_k}\mathcal{M}$ and $\ell_{X_k}(0) \geq \ell_{X_k}(V_k^*)$, where

$$\ell_{X_k}(V) := \langle \mathrm{grad}\, f(X_k), V\rangle_{X_k} + \frac{1}{2t}\|V\|_{X_k}^2 + h(\mathrm{Retr}_{X_k}(V)).$$

The reason that one can only find a stationary point of $\ell_{X_k}(V)$ is because the term $h(\mathrm{Retr}_{X_k}(V))$ is nonsmooth and nonconvex. Consequently, the convergence rate analysis in [52] requires a rather strong assumption (see [52, Assumption 4]), which is only known to hold when the retraction is the exponential mapping.

7.6. Riemannian Proximal Newton Method. A Riemannian proximal Newton (RPN) method for solving (1.1) with $h(X) = \|X\|_1$ is proposed in [87]. The method proceeds as follows. First, the RPG direction $V_k \in \mathrm{T}_{X_k}\mathcal{M}$ is obtained by solving (4.3). Then, the Riemannian Newton direction $U_k \in \mathrm{T}_{X_k}\mathcal{M}$ is obtained by solving the linear system $J_k(U_k) = -V_k$, where J_k is a certain linear operator related to the generalized Jacobian of V_k. Finally, a retraction step is performed to update X_{k+1}; i.e., $X_{k+1} = \mathrm{Retr}_{X_k}(U_k)$. It is proved in [87] that the proposed RPN method has a local superlinear convergence rate under certain assumptions.

8. Discussion and Concluding Remarks. Manifold optimization has attracted a lot of attention recently. In this paper, we have discussed our ManPG algorithm for solving (1.1), which involves minimizing a structured nonsmooth function over the Stiefel manifold. Unlike existing methods, our ManPG algorithm relies on proximal gradient information on the tangent space rather than subgradient information. Under the assumption that the smooth part of the objective function has a Lipschitz continuous gradient, we proved that ManPG converges globally to a stationary point of (1.1). Moreover, we analyzed the iteration complexity of ManPG for obtaining an ϵ-stationary solution. Our numerical experiments suggested that when combined with a regularized SSN method for finding the descent direction, ManPG performs efficiently and robustly. In particular, ManPG is more robust than SOC and PAMAL for solving the compressed modes and sparse PCA problems, as it is less sensitive to the choice of parameters. Moreover, ManPG significantly outperforms SOC and PAMAL for solving the sparse PCA problem in terms of CPU time needed to obtain the same solution.

It is worth noting that the convergence and iteration complexity analyses in section 5 also hold for other, not necessarily bounded, embedded submanifolds of a Euclidean space, provided that the objective function F satisfies some additional assumptions (e.g., F is coercive and lower bounded on $\mathcal{M}$). We focused on the Stiefel manifold because it is easier to discuss the SSN method in section 4.2 for finding the descent direction. As demonstrated in our tests on the compressed modes and sparse PCA problems, the efficiency of ManPG relies highly on that of solving the convex subproblem to find the descent direction. For general Riemannian submani-

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

folds, it remains an interesting question whether the operator $\mathcal{A}_k$ in (4.4) can be easily computed and the resulting subproblem solved efficiently.

Our ManPG algorithm has motivated many follow-up works, some of which were discussed in section 7. Riemannian optimization with nonsmooth objective functions is an active research area, and there are many developments that we are not able to cover in this paper. For example, various applications involving manifold optimization with nonsmooth objective functions can be found in [3, 97, 69]. Moreover, variants of ManPG have been applied to tackle applications such as sparse CCA [23] and robust matrix completion [50]. There are also some exciting recent developments in Riemannian ADMM [65] and Riemannian optimization with non-Lipschitz objective functions [108]. One important research goal is to design algorithms for nonsmooth optimization over Riemannian manifold that enjoy strong convergence guarantees and are numerically efficient.

Appendix A. Semismoothness of Proximal Mapping.

DEFINITION A.1. *Let $E : \Omega \to \mathbb{R}^q$ be locally Lipschitz continuous at $X \in \Omega \subset \mathbb{R}^p$. The B-subdifferential of E at X is defined by*

$$\partial_B E(X) := \left\{ \lim_{k\to\infty} E'(X_k) \,\middle|\, X^k \in D_E, X_k \to X \right\},$$

where D_E is the set of differentiable points of E in Ω. The set $\partial E(X) = \mathrm{conv}(\partial_B E(X))$ is called Clarke's generalized Jacobian, where conv *denotes the convex hull.*

Note that if $q = 1$ and E is convex, then the definition is the same as that of the standard convex subdifferential. Thus, we use the notation ∂ in Definition A.1.

DEFINITION A.2 ([75, 81]). *Let $E : \Omega \to \mathbb{R}^q$ be locally Lipschitz continuous at $X \in \Omega \subset \mathbb{R}^p$. We say that E is semismooth at $X \in \Omega$ if E is directionally differentiable at X and for any $J \in \partial E(X + \Delta X)$ with $\Delta X \to 0$,*

$$E(X + \Delta X) - E(X) - J\Delta X = o(\|\Delta X\|).$$

We say that E is strongly semismooth at X if E is semismooth at X and

$$E(X + \Delta X) - E(X) - J\Delta X = O(\|\Delta X\|^2).$$

We say that E is semismooth on Ω if it is semismooth at every $X \in \Omega$.

The proximal mapping of the ℓ_p ($p \geq 1$) norm is strongly semismooth [31, 93]. From [93, Proposition 2.26], if $E : \Omega \to \mathbb{R}^m$ is a piecewise $\mathcal{C}^1$ (piecewise smooth) function, then E is semismooth. If E is a piecewise $\mathcal{C}^2$ function, then E is strongly semismooth. It is known that proximal mappings of many interesting functions are piecewise linear or piecewise smooth.

REFERENCES

[1] T. E. ABRUDAN, J. ERIKSSON, AND V. KOIVUNEN, *Steepest descent algorithms for optimization under unitary matrix constraint*, IEEE Trans. Signal Process., 56 (2008), pp. 1134–1147. (Cited on p. 329)

[2] T. E. ABRUDAN, J. ERIKSSON, AND V. KOIVUNEN, *Conjugate gradient algorithm for optimization under unitary matrix constraint*, Signal Process., 89 (2009), pp. 1704–1714. (Cited on p. 329)

[3] P.-A. ABSIL AND S. HOSSEINI, *A collection of nonsmooth Riemannian optimization problems*, in Nonsmooth Optimization and Its Applications, Internat. Ser. Numer. Math. 170, Birkhäuser/Springer, Cham, 2019, pp. 1–15. (Cited on p. 347)

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[4] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, 2009. (Cited on pp. 320, 323, 324, 328, 329)
[5] P.-A. Absil and I. V. Oseledets, *Low-rank retractions: A survey and new results*, Comput. Optim. Appl., 62 (2015), pp. 5–29. (Cited on p. 336)
[6] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, *Fast image recovery using variable splitting and constrained optimization*, IEEE Trans. Image Process., 19 (2010), pp. 2345–2356. (Cited on p. 325)
[7] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, *Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality*, Math. Oper. Res., 35 (2010), pp. 438–457. (Cited on pp. 326, 327, 337)
[8] M. Bačák, R. Bergmann, G. Steidl, and A. Weinmann, *A second order nonsmooth variational model for restoring manifold-valued images*, SIAM J. Sci. Comput., 38 (2016), pp. A567–A597, https://doi.org/10.1137/15M101988X. (Cited on pp. 324, 325)
[9] T. Bendory, Y. C. Eldar, and N. Boumal, *Non-convex phase retrieval from STFT measurements*, IEEE Trans. Inform. Theory, 64 (2018), pp. 467–484. (Cited on p. 320)
[10] G. C. Bento, J. X. Cruz Neto, and P. R. Oliveira, *Convergence of Inexact Descent Methods for Nonconvex Optimization on Riemannian Manifolds*, preprint, https://arxiv.org/abs/1103.4828v1, 2011. (Cited on p. 325)
[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 (2016), pp. 743–755. (Cited on p. 325)
[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 (2017), pp. 548–562. (Cited on pp. 325, 336)
[13] R. L. Bishop and B. O'Neill, *Manifolds of negative curvature*, Trans. Amer. Math. Soc., 145 (1969), pp. 1–49. (Cited on pp. 324, 336)
[14] J. Bolte, S. Sabach, and M. Teboulle, *Proximal alternating linearized minimization for nonconvex and nonsmooth problems*, Math. Program., 146 (2014), pp. 459–494. (Cited on p. 327)
[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 (2014), pp. 848–866. (Cited on pp. 324, 325)
[16] N. Boumal, *Nonconvex phase synchronization*, SIAM J. Optim., 26 (2016), pp. 2355–2377, https://doi.org/10.1137/16M105808X. (Cited on p. 320)
[17] N. Boumal, *An Introduction to Optimization on Smooth Manifolds*, Cambridge University Press, 2023. (Cited on pp. 320, 324)
[18] N. Boumal and P.-A. Absil, *RTRMC: A Riemannian trust-region method for low-rank matrix completion*, in Advances in Neural Information Processing Systems 24, 2011, pp. 406–414. (Cited on p. 320)
[19] N. Boumal, P.-A. Absil, and C. Cartis, *Global rates of convergence for nonconvex optimization on manifolds*, IMA J. Numer. Anal., 39 (2019), pp. 1–33. (Cited on pp. 328, 329, 333, 334, 335, 336)
[20] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, *MANOPT, a MATLAB toolbox for optimization on manifolds*, J. Mach. Learn. Res., 15 (2014), pp. 1455–1459. (Cited on p. 326)
[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn., 3 (2011), pp. 1–122. (Cited on p. 325)
[22] S. Chen, Z. Deng, S. Ma, and A. M.-C. So, *Manifold proximal point algorithms for dual principal component pursuit and orthogonal dictionary learning*, IEEE Trans. Signal Process., 69 (2021), pp. 4759–4773. (Cited on p. 344)
[23] S. Chen, S. Ma, L. Xue, and H. Zou, *An alternating manifold proximal gradient method for sparse principal component analysis and sparse canonical correlation analysis*, INFORMS J. Optim., 2 (2020), pp. 192–208. (Cited on p. 347)
[24] W. Chen, H. Ji, and Y. You, *An augmented Lagrangian method for ℓ_1-regularized optimization problems with orthogonality constraints*, SIAM J. Sci. Comput., 38 (2016), pp. B570–B592, https://doi.org/10.1137/140988875. (Cited on pp. 326, 327, 336, 337, 338, 342)
[25] A. Cherian and S. Sra, *Riemannian dictionary learning and sparse coding for positive definite matrices*, IEEE Trans. Neural Networks and Learning Syst., 28 (2017), pp. 2859–2871. (Cited on p. 320)
[26] P. L. Combettes and J.-C. Pesquet, *A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery*, IEEE J. Selected Topics Signal Process., 1 (2007), pp. 564–574. (Cited on p. 325)

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

[27] A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. G. Lanckriet, *A direct formulation for sparse PCA using semidefinite programming*, SIAM Rev., 49 (2007), pp. 434–448, https://doi.org/10.1137/050645506. (Cited on pp. 321, 342)

[28] G. Dirr, U. Helmke, and C. Lageman, *Nonsmooth Riemannian optimization with applications to sphere packing and grasping*, in Lagrangian and Hamiltonian Methods for Nonlinear Control, Lect. Notes Control Inf. Sci. 366, Springer, Berlin, 2007, pp. 28–45. (Cited on p. 324)

[29] J. Eckstein, *Splitting Methods for Monotone Operators with Applications to Parallel Optimization*, Ph.D. thesis, Massachusetts Institute of Technology, Boston, 1989. (Cited on p. 325)

[30] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353, https://doi.org/10.1137/S0895479895290954. (Cited on p. 328)

[31] F. Facchinei and J. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Springer Science & Business Media, 2007. (Cited on p. 347)

[32] C. Fang, C. J. Li, Z. Lin, and T. Zhang, *SPIDER: Near-optimal nonconvex optimization via stochastic path-integrated differential estimator*, in Advances in Neural Information Processing Systems 31, 2018, pp. 689–699. (Cited on p. 345)

[33] O. P. Ferreira and P. R. Oliveira, *Subgradient algorithm on Riemannian manifolds*, J. Optim. Theory Appl., 97 (1998), pp. 93–104. (Cited on p. 324)

[34] O. P. Ferreira and P. R. Oliveira, *Proximal point algorithm on Riemannian manifold*, Optimization, 51 (2002), pp. 257–270. (Cited on pp. 324, 325)

[35] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*, North-Holland, Amsterdam, 1983. (Cited on p. 325)

[36] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite-element approximations*, Comput. Math. Appl., 2 (1976), pp. 17–40. (Cited on p. 325)

[37] M. Genicot, W. Huang, and N. T. Trendafilov, *Weakly correlated sparse components with nearly orthonormal loadings*, in Lecture Notes in Comput. Sci. 9389, Springer, Cham, 2015, pp. 484–490. (Cited on p. 342)

[38] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, SIAM, Philadelphia, 1989, https://doi.org/10.1137/1.9781611970838. (Cited on p. 325)

[39] R. Glowinski and A. Marrocco, *Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires*, Rev. Française Automat. Informat. Recherche Opérationnelle Sér. Rouge Anal. Numér., 9 (1975), pp. 41–76. (Cited on p. 325)

[40] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM J. Imaging Sci., 2 (2009), pp. 323–343, https://doi.org/10.1137/080725891. (Cited on p. 325)

[41] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 2012. (Cited on p. 328)

[42] P. Grohs and S. Hosseini, *Nonsmooth trust region algorithms for locally Lipschitz functions on Riemannian manifolds*, IMA J. Numer. Anal., 36 (2016), pp. 1167–1192. (Cited on p. 324)

[43] P. Grohs and S. Hosseini, *ε-subgradient algorithms for locally Lipschitz functions on Riemannian manifolds*, Adv. Comput. Math., 42 (2016), pp. 333–360. (Cited on p. 324)

[44] J.-B. Hiriart-Urruty, J.-J. Strodiot, and V. H. Nguyen, *Generalized Hessian matrix and second-order optimality conditions for problems with $C^{1,1}$ data*, Appl. Math. Optim., 11 (1984), pp. 43–56. (Cited on p. 332)

[45] S. Hosseini, *Convergence of Nonsmooth Descent Methods via Kurdyka-Łojasiewicz Inequality on Riemannian Manifolds*, Technical report, Hausdorff Center for Mathematics and Institute for Numerical Simulation, University of Bonn, Bonn, Germany, 2015. (Cited on p. 324)

[46] S. Hosseini, W. Huang, and R. Yousefpour, *Line search algorithms for locally Lipschitz functions on Riemannian manifolds*, SIAM J. Optim., 28 (2018), pp. 596–619, https://doi.org/10.1137/16M1108145. (Cited on p. 324)

[47] S. Hosseini and M. R. Pouryayevali, *Generalized gradients and characterization of epi-Lipschitz sets in Riemannian manifolds*, Nonlinear Anal., 72 (2011), pp. 3884–3895. (Cited on p. 327)

[48] S. Hosseini and A. Uschmajew, *A Riemannian gradient sampling algorithm for nonsmooth optimization on manifolds*, SIAM J. Optim., 27 (2017), pp. 173–189, https://doi.org/10.1137/16M1069298. (Cited on p. 324)

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[49] H. Hotelling, *Analysis of a complex of statistical variables into principal components*, J. Educ. Psych., 24 (1933), pp. 417–441. (Cited on p. 321)

[50] M. Huang, S. Ma, and L. Lai, *Robust low-rank matrix completion via an alternating manifold proximal gradient continuation method*, IEEE Trans. Signal Process., 69 (2021), pp. 2639–2652. (Cited on p. 347)

[51] W. Huang and P. Hand, *Blind deconvolution by a steepest descent algorithm on a quotient manifold*, SIAM J. Imaging Sci., 11 (2018), pp. 2757–2785, https://doi.org/10.1137/17M1151390. (Cited on p. 320)

[52] W. Huang and K. Wei, *Riemannian proximal gradient methods*, Math. Prog., 194 (2022), pp. 371–413. (Cited on p. 346)

[53] B. Jiang and Y.-H. Dai, *A framework of constraint preserving update schemes for optimization on Stiefel manifold*, Math. Program., 153 (2015), pp. 535–575. (Cited on p. 328)

[54] 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*, preprint, https://arxiv.org/abs/1705.09059v1, 2017. (Cited on p. 320)

[55] I. Jolliffe, N. Trendafilov, and M. Uddin, *A modified principal component technique based on the LASSO*, J. Comput. Graph. Statist., 12 (2003), pp. 531–547. (Cited on pp. 321, 342)

[56] M. Journee, Yu. Nesterov, P. Richtarik, and R. Sepulchre, *Generalized power method for sparse principal component analysis*, J. Mach. Learn. Res., 11 (2010), pp. 517–553. (Cited on pp. 321, 342)

[57] A. Kovnatsky, K. Glashoff, and M. M. Bronstein, *MADMM: A generic algorithm for non-smooth optimization on manifolds*, in European Conference on Computer Vision, Springer, 2016, pp. 680–696. (Cited on pp. 325, 326)

[58] R. Lai and S. Osher, *A splitting method for orthogonality constrained problems*, J. Sci. Comput., 58 (2014), pp. 431–449. (Cited on pp. 325, 336, 337)

[59] G. Lerman and T. Maunu, *An overview of robust subspace recovery*, Proc. IEEE, 106 (2018), pp. 1380–1410. (Cited on p. 344)

[60] G. Lerman and T. Maunu, *Fast, robust and non-convex subspace recovery*, Inf. Inference, 7 (2018), pp. 277–336. (Cited on p. 344)

[61] G. Lerman, M. B. McCoy, J. A. Tropp, and T. Zhang, *Robust computation of linear models by convex relaxation*, Found. Comput. Math., 15 (2015), pp. 363–410. (Cited on p. 344)

[62] T. Maunu, C. Yu, and G. Lerman, *Stochastic and private nonconvex outlier-robust PCA*, J. Mach. Learn. Res., 190 (2022), pp. 173–188. (Cited on p. 344)

[63] T. Maunu, T. Zhang, and G. Lerman, *A well-tempered landscape for non-convex robust subspace recovery*, J. Mach. Learn. Res., 20 (2019), pp. 1–59. (Cited on p. 344)

[64] J. Li, K. Balasubramanian, and S. Ma, *Stochastic zeroth-order Riemannian derivative estimation and optimization*, Math. Oper. Res., 48 (2023), pp. 1183–1211. (Cited on pp. 345, 346)

[65] J. Li, S. Ma, and T. Srivastava, *A Riemannian ADMM*, preprint, https://arxiv.org/abs/2211.02163, 2022. (Cited on p. 347)

[66] X. Li, S. Chen, Z. Deng, Q. Qu, Z. Zhu, and A. M.-C. So, *Weakly convex optimization over Stiefel manifold using Riemannian subgradient-type methods*, SIAM J. Optim., 31 (2021), pp. 1605–1634, https://doi.org/10.1137/20M1321000. (Cited on p. 344)

[67] X. Li, D. Sun, and K.-C. Toh, *A highly efficient semismooth Newton augmented Lagrangian method for solving lasso problems*, SIAM J. Optim., 28 (2018), pp. 433–458, https://doi.org/10.1137/16M1097572. (Cited on p. 331)

[68] P. L. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM J. Numer. Anal., 16 (1979), pp. 964–979, https://doi.org/10.1137/0716071. (Cited on p. 325)

[69] H. Liu, X. Li, and A. M.-C. So, *ReSync: Riemannian subgradient-based robust rotation synchronization*, in Advances in Neural Information Processing Systems 36, 2023. (Cited on p. 347)

[70] H. Liu, A. M.-C. So, and W. Wu, *Quadratic optimization with orthogonality constraint: Explicit Łojasiewicz exponent and linear convergence of retraction-based line-search and stochastic variance-reduced gradient methods*, Math. Program., 178 (2019), pp. 215–262, https://doi.org/10.1007/s10107-018-1285-1. (Cited on pp. 320, 329)

[71] 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 (2017), pp. 2426–2446, https://doi.org/10.1137/16M110109X. (Cited on p. 320)

[72] H. Liu, M.-C. Yue, and A. M.-C. So, *A unified approach to synchronization problems over subgroups of the orthogonal group*, Appl. Comput. Harmon. Anal., 66 (2023), pp. 320–372,

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

https://doi.org/10.1016/j.acha.2023.05.002. (Cited on p. 320)

[73] S. Ma, *Alternating direction method of multipliers for sparse principal component analysis*, J. Oper. Res. Soc. China, 1 (2013), pp. 253–274. (Cited on p. 321)

[74] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, Wiley Ser. Probab. Math. Statist. Appl. Probab. Statist., Wiley, 1988. (Cited on p. 332)

[75] R. Mifflin, *Semismooth and semiconvex functions in constrained optimization*, SIAM J. Control Optim., 15 (1977), pp. 959–972, https://doi.org/10.1137/0315061. (Cited on pp. 330, 347)

[76] Yu. Nesterov and V. Spokoiny, *Random gradient-free minimization of convex functions*, Found. Comput. Math., 17 (2017), pp. 527–566. (Cited on p. 345)

[77] Y. Nishimori and S. Akaho, *Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold*, Neurocomput., 67 (2005), pp. 106–135. (Cited on p. 329)

[78] V. Ozoliņš, R. Lai, R. Caflisch, and S. Osher, *Compressed modes for variational problems in mathematics and physics*, Proc. Natl. Acad. Sci. USA, 110 (2013), pp. 18368–18373. (Cited on pp. 321, 337, 338)

[79] K. Pearson, *LIII. On lines and planes of closest fit to systems of points in space*, London Edinburgh Dublin Philos. Mag. J. Sci., 2 (1901), pp. 559–572. (Cited on p. 321)

[80] H. Qi and D. Sun, *An augmented Lagrangian dual approach for the H-weighted nearest correlation matrix problem*, IMA J. Numer. Anal., 31 (2011), pp. 491–511. (Cited on p. 331)

[81] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Program., 58 (1993), pp. 353–367. (Cited on pp. 330, 347)

[82] R. Rockafellar and R. J.-B. Wets, *Variational Analysis*, Springer Science & Business Media, 2009. (Cited on p. 333)

[83] B. Savas and L.-H. Lim, *Quasi-Newton methods on Grassmannians and multilinear approximations of tensors*, SIAM J. Sci. Comput., 32 (2010), pp. 3352–3393, https://doi.org/10.1137/090763172. (Cited on p. 329)

[84] O. Shamir, *A stochastic PCA and SVD algorithm with an exponential convergence rate*, in Proceedings of the 32nd International Conference on Machine Learning, 2015, pp. 144–152. (Cited on p. 320)

[85] O. Shamir, *Fast stochastic algorithms for SVD and PCA: Convergence properties and convexity*, in Proceedings of the 33rd International Conference on Machine Learning, 2016, pp. 248–256. (Cited on p. 320)

[86] H. Shen and J. Z. Huang, *Sparse principal component analysis via regularized low rank matrix approximation*, J. Multivariate Anal., 99 (2008), pp. 1015–1034. (Cited on pp. 321, 342)

[87] W. Si, P.-A. Absil, W. Huang, R. Jiang, and S. Vary, *A Riemannian proximal Newton method*, SIAM J. Optim., 34 (2024), pp. 654–681, https://doi.org/10.1137/23M1565097. (Cited on p. 346)

[88] 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, M. Fukushima and L. Qi, eds., Springer, New York, 1998, pp. 355–369. (Cited on pp. 330, 332)

[89] J. Sun, Q. Qu, and J. Wright, *Complete dictionary recovery over the sphere* I*: Overview and the geometric picture*, IEEE Trans. Inform. Theory, 63 (2017), pp. 853–884. (Cited on p. 320)

[90] J. Sun, Q. Qu, and J. Wright, *A geometrical analysis of phase retrieval*, Found. Comput. Math., 18 (2018), pp. 1131–1198. (Cited on p. 320)

[91] J. Tang and H. Liu, *Unsupervised feature selection for linked social media data*, in Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2012, pp. 904–912. (Cited on p. 322)

[92] M. C. Tsakiris and R. Vidal, *Dual principal component pursuit*, J. Mach. Learn. Res., 19 (2018), pp. 1–50. (Cited on p. 344)

[93] M. Ulbrich, *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces*, MOS-SIAM Ser. Optim. 11, SIAM, Philadelphia, 2011, https://doi.org/10.1137/1.9781611970692. (Cited on p. 347)

[94] B. Vandereycken, *Low-rank matrix completion by Riemannian optimization*, SIAM J. Optim., 23 (2013), pp. 1214–1236, https://doi.org/10.1137/110845768. (Cited on p. 320)

[95] B. Wang, S. Ma, and L. Xue, *Riemannian stochastic proximal gradient methods for nonsmooth optimization over the Stiefel manifold*, J. Mach. Learn. Res., 23 (2022), pp. 1–33. (Cited on p. 345)

[96] C. Wang, D. Sun, and K.-C. Toh, *Solving log-determinant optimization problems by a Newton-CG primal proximal point algorithm*, SIAM J. Optim., 20 (2010), pp. 2994–3013, https://doi.org/10.1137/090772514. (Cited on p. 331)

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[97] P. Wang, H. Liu, and A. M.-C. So, *Linear convergence of a proximal alternating minimization method with extrapolation for ℓ_1-norm principal component analysis*, SIAM J. Optim., 33 (2023), pp. 684–712, https://doi.org/10.1137/21M1434507. (Cited on p. 347)

[98] Y. Wang, W. Yin, and J. Zeng, *Global convergence of ADMM in nonconvex nonsmooth optimization*, J. Sci. Comput., 78 (2019), pp. 29–63. (Cited on pp. 325, 326)

[99] Z. Wang, K. Ji, Y. Zhou, Y. Liang, and V. Tarokh, *SpiderBoost and momentum: Faster stochastic variance reduction algorithms*, in Advances in Neural Information Processing Systems 32, 2019, pp. 2406–2416. (Cited on p. 345)

[100] Z. Wang, B. Liu, S. Chen, S. Ma, L. Xue, and H. Zhao, *A manifold proximal linear method for sparse spectral clustering with application to single-cell RNA sequencing data analysis*, INFORMS J. Optim., 4 (2021), pp. 200–214. (Cited on pp. 344, 345)

[101] Z. Wen and W. Yin, *A feasible method for optimization with orthogonality constraints*, Math. Program., 142 (2013), pp. 397–434. (Cited on pp. 320, 328)

[102] 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. (Cited on pp. 331, 332, 333, 336)

[103] J. Yang, D. Sun, and K.-C. Toh, *A proximal point algorithm for log-determinant optimization with group Lasso regularization*, SIAM J. Optim., 23 (2013), pp. 857–893, https://doi.org/10.1137/120864192. (Cited on p. 331)

[104] J. Yang, W. Yin, Y. Zhang, and Y. Wang, *A fast algorithm for edge-preserving variational multichannel image restoration*, SIAM J. Imaging Sci., 2 (2009), pp. 569–592, https://doi.org/10.1137/080730421. (Cited on p. 325)

[105] L. Yang, D. Sun, and K.-C. Toh, *SDPNAL+: A majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints*, Math. Program. Comput., 7 (2015), pp. 331–366. (Cited on p. 331)

[106] W. H. Yang, L.-H. Zhang, and R. Song, *Optimality conditions for the nonlinear programming problems on Riemannian manifolds*, Pacific J. Optim., 10 (2014), pp. 415–434. (Cited on pp. 327, 335)

[107] Y. Yang, H. Shen, Z. Ma, Z. Huang, and X. Zhou, *$\ell_{2,1}$-norm regularized discriminative feature selection for unsupervised learning*, in Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Vol. 2, 2011, pp. 1589–1594. (Cited on p. 322)

[108] C. Zhang, X. Chen, and S. Ma, *A Riemannian smoothing steepest descent method for non-Lipschitz optimization on embedded submanifolds of $\mathbb{R}^n$*, Math. Oper. Res., https://doi.org/10.1287/moor.2022.0286. (Cited on p. 347)

[109] C.-H. Zhang, *Nearly unbiased variable selection under minimax concave penalty*, Ann. Statist., 38 (2010), pp. 894–942. (Cited on p. 322)

[110] H. Zhang, S. Reddi, and S. Sra, *Fast stochastic optimization on Riemannian manifolds*, in Advances in Neural Information Processing Systems 29, 2016, pp. 4599–4607. (Cited on p. 320)

[111] H. Zhang and S. Sra, *First-order methods for geodesically convex optimization*, Proc. Mach. Learn. Res., 49 (2016), pp. 1617–1638. (Cited on p. 336)

[112] J. Zhang, S. Ma, and S. Zhang, *Primal-dual optimization algorithms over Riemannian manifolds: An iteration complexity analysis*, Math. Program., 184 (2020), pp. 445–490, https://doi.org/10.1007/s10107-019-01418-8. (Cited on pp. 325, 326)

[113] 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 Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2017, pp. 4894–4902. (Cited on p. 322)

[114] X.-Y. Zhao, D. Sun, and K.-C. Toh, *A Newton-CG augmented Lagrangian method for semidefinite programming*, SIAM J. Optim., 20 (2010), pp. 1737–1765, https://doi.org/10.1137/080718206. (Cited on p. 331)

[115] G. Zhou and K.-C. Toh, *Superlinear convergence of a Newton-type algorithm for monotone equations*, J. Optim. Theory Appl., 125 (2005), pp. 205–221. (Cited on pp. 331, 332)

[116] H. Zhu, X. Zhang, D. Chu, and L. Liao, *Nonconvex and nonsmooth optimization with generalized orthogonality constraints: An approximate augmented Lagrangian method*, J. Sci. Comput., 72 (2017), pp. 331–372. (Cited on p. 327)

[117] Z. Zhu, Y. Wang, D. Robinson, D. Naiman, R. Vidal, and M. Tsakiris, *Dual principal component pursuit: Improved analysis and efficient algorithms*, in Advances in Neural Information Processing Systems 31, 2018, pp. 2175–2185. (Cited on p. 344)

[118] H. Zou, T. Hastie, and R. Tibshirani, *Sparse principal component analysis*, J. Comput. Graph. Statist., 15 (2006), pp. 265–286. (Cited on pp. 321, 342)

[119] H. Zou and L. Xue, *A selective overview of sparse principal component analysis*, Proc. IEEE, 106 (2018), pp. 1311–1320. (Cited on p. 321)

Downloaded 05/10/24 to 73.232.71.252 by Shiqian Ma (sqma@rice.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.