Journal of the Operations Research Society of China (2020) 8:199–248
https://doi.org/10.1007/s40305-020-00295-9

A Brief Introduction to Manifold Optimization

Jiang Hu[1] · Xin Liu[2,3] · Zai-Wen Wen[1] · Ya-Xiang Yuan[2]

Received: 12 June 2019 / Revised: 10 December 2019 / Accepted: 4 February 2020 /
Published online: 4 April 2020

Abstract
Manifold optimization is ubiquitous in computational and applied mathematics, statistics, engineering, machine learning, physics, chemistry, etc. One of the main challenges usually is the non-convexity of the manifold constraints. By utilizing the geometry of manifold, a large class of constrained optimization problems can be viewed as unconstrained optimization problems on manifold. From this perspective, intrinsic structures, optimality conditions and numerical algorithms for manifold optimization are investigated. Some recent progress on the theoretical results of manifold optimization is also presented.

Keywords Convergence · First-order-type algorithms · Manifold optimization · Retraction · Second-order-type algorithms

Xin Liu's research was supported in part by the National Natural Science Foundation of China (No. 11971466), Key Research Program of Frontier Sciences, Chinese Academy of Sciences (No. ZDBS-LY-7022), the National Center for Mathematics and Interdisciplinary Sciences, Chinese Academy of Sciences and the Youth Innovation Promotion Association, CAS.
Zai-Wen Wen's research was supported in part by the the National Natural Science Foundation of China (Nos. 11421101 and 11831002), and the Beijing Academy of Artificial Intelligence.
Ya-Xiang Yuan's research was supported in part by the National Natural Science Foundation of China (Nos. 11331012 and 11461161005).

✉ Zai-Wen Wen
wenzw@pku.edu.cn

Jiang Hu
jianghu@pku.edu.cn

Xin Liu
liuxin@lsec.cc.ac.cn

Ya-Xiang Yuan
yyx@lsec.cc.ac.cn

[1] Beijing International Center for Mathematical Research, Peking University, Beijing 100871, China

[2] State Key Laboratory of Scientific and Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China

[3] University of Chinese Academy of Sciences, Beijing 100190, China

Mathematics Subject Classification 15A18 · 49Q99 · 65K05 · 90C22 · 90C26 · 90C27 · 90C30

1 Introduction

Manifold optimization is concerned with the following optimization problem:

$$\min_{x \in \mathcal{M}} \quad f(x), \tag{1.1}$$

where $\mathcal{M}$ is a Riemannian manifold and f is a real-valued function on $\mathcal{M}$, which can be non-smooth. If additional constraints other than the manifold constraint are involved, we can add in f an indicator function of the feasible set of these additional constraints. Hence, (1.1) covers a general formulation for manifold optimization. In fact, manifold optimization has been widely used in computational and applied mathematics, statistics, machine learning, data science, material science and so on. The existence of the manifold constraint is one of the main difficulties in algorithmic design and theoretical analysis.

Notations Let $\mathbb{R}$ and $\mathbb{C}$ be the sets of real and complex numbers, respectively. For a matrix $X \in \mathbb{C}^{n\times p}$, $\bar{X}$, X^*, $\Re X$ and $\Im X$ are its complex conjugate, complex conjugate transpose, real and imaginary parts, respectively. Let $\mathbb{S}^n$ be the set of all n-by-n real symmetric matrices. For a matrix $M \in \mathbb{C}^{n\times n}$, $\mathrm{diag}(M)$ is a vector in $\mathbb{C}^n$ formulated by the diagonal elements of M. For a vector $c \in \mathbb{C}^n$, $\mathrm{Diag}(c)$ is an n-by-n diagonal matrix with the elements of c on the diagonal. For a differentiable function f on $\mathcal{M}$, let $\mathrm{grad}\, f(x)$ and $\mathrm{Hess}\, f(x)$ be its Riemannian gradient and Hessian at x, respectively. If f can be extended to the ambient Euclidean space, we denote its Euclidean gradient and Hessian by $\nabla f(x)$ and $\nabla^2 f(x)$, respectively.

This paper is organized as follows. In Sect. 2, various kinds of applications of manifold optimization are presented. We review geometry on manifolds, optimality conditions as well as state-of-the-art algorithms for manifold optimization in Sect. 3. For some selected practical applications in Sect. 2, a few theoretical results based on manifold optimization are introduced in Sect. 4.

2 Applications of Manifold Optimization

In this section, we introduce applications of manifold optimization in p-harmonic flow, the maxcut problem, low-rank nearest correlation matrix estimation, phase retrieval, Bose–Einstein condensates, cryo-electron microscopy (cryo-EM), linear eigenvalue problem, nonlinear eigenvalue problem from electronic structure calculations, combinatorial optimization, deep learning, etc.

2.1 *P*-Harmonic Flow

P-harmonic flow is used in the color image recovery and medical image analysis. For instance, in medical image analysis, the human brain is often mapped to a unit sphere via a conformal mapping, see Fig. 1. By establishing a conformal mapping between

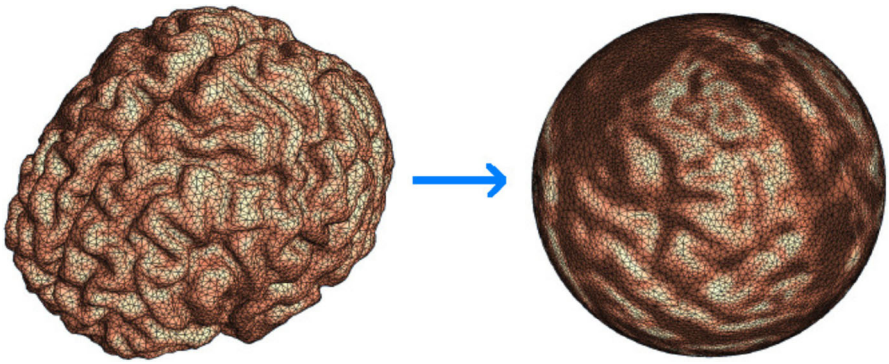

Fig. 1 Conformal mapping between the human brain and the unit sphere [1]

an irregular surface and the unit sphere, we can handle the complicated surface with the simple parameterizations of the unit sphere. Here, we focus on the conformal mapping between genus-0 surfaces. From [2], a diffeomorphic map between two genus-0 surfaces $\mathcal{N}_1$ and $\mathcal{N}_2$ is conformal if and only if it is a local minimizer of the corresponding harmonic energy. Hence, one effective way to compute the conformal mapping between two genus-0 surfaces is to minimize the harmonic energy of the map. Before introducing the harmonic energy minimization model and the diffeomorphic mapping, we review some related concepts on manifold. Let $\phi_{\mathcal{N}_1}(x^1, x^2) : \mathbb{R}^2 \to \mathcal{N}_1 \subset \mathbb{R}^3$, $\phi_{\mathcal{N}_2}(x^1, x^2) : \mathbb{R}^2 \to \mathcal{N}_2 \subset \mathbb{R}^3$ be the local coordinates on $\mathcal{N}_1$ and $\mathcal{N}_2$, respectively. The first fundamental form on $\mathcal{N}_1$ is $g = \sum_{ij} g_{ij} \mathrm{d}x^i \mathrm{d}x^j$, where $g_{ij} = \frac{\partial \phi_{\mathcal{N}_1}}{\partial x^i} \cdot \frac{\partial \phi_{\mathcal{N}_1}}{\partial x^j}$, $i, j = 1, 2$. The first fundamental form on $\mathcal{N}_2$ is $h = \sum_{ij} h_{ij} \mathrm{d}x^i \mathrm{d}x^j$, where $h_{ij} = \frac{\partial \phi_{\mathcal{N}_2}}{\partial x^i} \cdot \frac{\partial \phi_{\mathcal{N}_2}}{\partial x^j}$, $i, j = 1, 2$. Given a smooth map $f : \mathcal{N}_1 \to \mathcal{N}_2$, whose local coordinate representation is $f(x^1, x^2) = (f_1(x^1, x^2), f_2(x^1, x^2))$, the density of the harmonic energy of f is

$$e(f) = \|\mathrm{d}f\|^2 = \sum_{i,j=1,2} g^{ij} \langle f_* \partial_{x^i}, f_* \partial_{x^j} \rangle_h,$$

where (g^{ij}) is the inverse of (g_{ij}) and the inner product between $f_* \partial_{x^i}$ and $f_* \partial_{x^j}$ is defined as:

$$\left\langle f_* \partial_{x^i}, f_* \partial_{x^j} \right\rangle_h = \left\langle \sum_{m=1}^{2} \frac{\partial f_m}{\partial x^i} \partial_{y_m}, \sum_{n=1}^{2} \frac{\partial f_n}{\partial x^j} \partial_{y_n} \right\rangle_h = \sum_{m,n=1}^{2} h_{mn} \frac{\partial f_m}{\partial x^i} \frac{\partial f_n}{\partial x^j}.$$

This also defines a new Riemannian metric on $\mathcal{N}_1$, $f^*(h)(\vec{v}_1, \vec{v}_2) := \langle f_*(\vec{v}_1), f_*(\vec{v}_2) \rangle_h$, which is called the pullback metric induced by f and h. Denote by $\mathbb{S}(\mathcal{N}_1, \mathcal{N}_2)$ the set of smooth maps between $\mathcal{N}_1$ and $\mathcal{N}_2$. Then, the harmonic flow minimization problem solves

$$\min_{f\in\mathbb{S}(\mathcal{N}_1,\mathcal{N}_2)} \mathbf{E}(f) = \frac{1}{2}\int_{\mathcal{N}_1} e(f)\mathrm{d}\mathcal{N}_1,$$

where $\mathbf{E}(f)$ is called the harmonic energy of f. Stationary points of $\mathbf{E}$ are the harmonic maps from $\mathcal{N}_1$ to $\mathcal{N}_2$. In particular, if $\mathcal{N}_2 = \mathbb{R}^2$, the conformal map $f = (f_1, f_2)$ is two harmonic functions defined on $\mathcal{N}_1$. If we consider a p-harmonic map from n-dimensional manifold $\mathcal{M}$ to n-dimensional sphere $\mathrm{Sp}(n) := \{x \in \mathbb{R}^{n+1} \mid \|x\|_2 = 1\} \subset \mathbb{R}^{n+1}$, the p-harmonic energy minimization problem can be written as

$$\begin{aligned}\min_{\vec{F}(x)=(f_1(x),\cdots,f_{n+1}(x))} \quad & \mathbf{E}_p(\vec{F}) = \frac{1}{p}\int_{\mathcal{M}} \left(\sum_{k=1}^{n+1} \|\mathrm{grad}\, f_k\|^2\right)^{p/2} \mathrm{d}\mathcal{M} \\ \text{s.t.} \quad & \vec{F}(x) \in S^n, \quad \forall x \in \mathcal{M},\end{aligned}$$

where grad f_k denotes the Riemannian gradient of f_k on manifold $\mathcal{M}$.

2.2 The Maxcut Problem

Given a graph $G = (V, E)$ with a set of n vertexes V ($|V| = n$) and a set of edges E. Denote by the weight matrix $W = (w_{ij})$. The maxcut problem is to split V into two non-empty sets $(S, V\backslash S)$ such that the total weights of edges in the cut are maximized. For each vertex $i = 1, \cdots, n$, we define $x_i = 1$ if $i \in S$ and -1 otherwise. The maxcut problem can be written as

$$\max_{x\in\mathbb{R}^n} \frac{1}{2}\sum_{i<j} w_{ij}(1 - x_i x_j) \text{ s.t. } x_i^2 = 1,\ i = 1, \cdots, n. \tag{2.1}$$

It is NP-hard. By relaxing the rank-1 constraint $xx^\top$ to a positive semidefinite matrix X and further neglecting the rank-1 constraint on X, we obtain the following semidefinite program (SDP)

$$\max_{X\succeq 0} \quad \mathrm{tr}(CX) \text{ s.t. } X_{ii} = 1, \quad i = 1, \cdots, n, \tag{2.2}$$

where C is the graph Laplacian matrix divided by 4, i.e., $C = -\frac{1}{4}(\mathrm{Diag}(We) - W)$ with an n-dimensional vector e of all ones. If we decompose $X = V^\top V$ with $V := [V_1, \cdots, V_n] \in \mathbb{R}^{p\times n}$, a non-convex relaxation of (2.1) is

$$\max_{V=[V_1,\cdots,V_n]} \mathrm{tr}(CV^\top V) \text{ s.t. } \|V_i\|_2 = 1,\ i = 1, \cdots, n. \tag{2.3}$$

It is an optimization problem over multiple spheres.

2.3 Low-Rank Nearest Correlation Estimation

Given a symmetric matrix $C \in \mathbb{S}^n$ and a nonnegative symmetric weight matrix $H \in \mathbb{S}^n$, this problem is to find a correlation matrix X of low rank such that the

distance weighted by H between X and C is minimized:

$$\min_{X\succeq 0} \frac{1}{2}\|H\odot(X-C)\|_F^2 \ \text{ s.t. } X_{ii}=1,\ i=1,\cdots,n,\ \ \text{rank}\,(X)\leqslant p. \tag{2.4}$$

Algorithms for solving (2.4) can be found in [3,4]. Similar to the maxcut problem, we decompose the low-rank matrix X with $X=V^\top V$, in which $V=[V_1,\cdots,V_n]\in\mathbb{R}^{p\times n}$. Therefore, problem (2.4) is converted to a quartic polynomial optimization problem over multiple spheres:

$$\min_{V\in\mathbb{R}^{p\times n}} \frac{1}{2}\|H\odot(V^\top V-C)\|_F^2 \ \text{s.t. } \|V_i\|_2=1,\ i=1,\cdots,n.$$

2.4 Phase Retrieval

Given some modules of a complex signal $x\in\mathbb{C}^n$ under linear measurements, a classic model for phase retrieval is to solve

$$\begin{aligned}\text{find}\quad & x\in\mathbb{C}^n\\ \text{s.t.}\quad & |Ax|=b,\end{aligned} \tag{2.5}$$

where $A\in\mathbb{C}^{m\times n}$ and $b\in\mathbb{R}^m$. This problem plays an important role in X-ray, crystallography imaging, diffraction imaging and microscopy. Problem (2.5) is equivalent to the following problem, which minimizes the phase variable y and signal variable x simultaneously:

$$\begin{aligned}\min_{x\in\mathbb{C}^n,y\in\mathbb{C}^m}\quad & \|Ax-y\|_2^2\\ \text{s.t.}\quad & |y|=b.\end{aligned}$$

In [5], the problem above is rewritten as

$$\begin{aligned}\min_{x\in\mathbb{C}^n,u\in\mathbb{C}^m}\ & \frac{1}{2}\|Ax-\text{Diag}\{b\}u\|_2^2\\ \text{s.t.}\ \ & |u_i|=1, i=1,\cdots,m.\end{aligned} \tag{2.6}$$

For a fixed phase u, the signal x can be represented by $x=A^\dagger\text{Diag}\{b\}u$. Hence, problem (2.6) is converted to

$$\begin{aligned}\min_{u\in\mathbb{C}^m}\quad & u^*Mu\\ \text{s.t.}\quad & |u_i|=1, i=1,\cdots,m,\end{aligned} \tag{2.7}$$

where $M=\text{Diag}\{b\}(I-AA^\dagger)\text{Diag}\{b\}$ is positive definite. It can be regarded as a generalization of the maxcut problem to complex spheres.

If we denote $X = uu^*$, (2.7) can also be modeled as the following SDP problem [6]

$$\min \quad \mathrm{tr}(MX) \quad \text{s.t.} \quad X \succeq 0, \quad \mathrm{rank}\,(X) = 1,$$

which can be further relaxed as

$$\min \quad \mathrm{tr}(MX) \quad \text{s.t.} \quad \mathrm{rank}\,(X) = 1,$$

whose constraint is a fixed-rank manifold.

2.5 Bose–Einstein Condensates

In Bose–Einstein condensates (BEC), the total energy functional is defined as

$$E(\psi) = \int_{\mathbb{R}^d} \left[\frac{1}{2}|\nabla\psi(w)|^2 + V(w)|\psi(w)|^2 + \frac{\beta}{2}|\psi(w)|^4 - \Omega\bar{\psi}(w)L_z(w) \right] \mathrm{d}w,$$

where $w \in \mathbb{R}^d$ is the spatial coordinate vector, $\bar{\psi}$ is the complex conjugate of ψ, $L_z = -i(x\partial - y\partial x)$, $V(w)$ is an external trapping potential, and β, Ω are given constants. The ground state of BEC is defined as the minimizer of the following optimization problem

$$\min_{\phi \in S} \quad E(\phi),$$

where the spherical constraint S is

$$S = \left\{ \phi \ : \ E(\phi) \leqslant \infty, \ \int_{\mathbb{R}^d} |\phi(w)|^2 \mathrm{d}w = 1 \right\}.$$

The Euler–Lagrange equation of this problem is to find $(\mu \in \mathbb{R}, \ \phi(w))$ such that

$$\mu\phi(w) = -\frac{1}{2}\nabla^2\phi(w) + V(w)\phi(w) + \beta|\phi(w)|^2\phi(w) - \Omega L_z\phi(w), \ \xi \in \mathbb{R}^d,$$

and

$$\int_{\mathbb{R}^d} |\phi(w)|^2 \mathrm{d}w = 1.$$

Utilizing some proper discretization, such as finite difference, sine pseudospectral and Fourier pseudospectral methods, we obtain a discretized BEC problem

$$\min_{x \in \mathbb{C}^M} \ f(x) := \frac{1}{2}x^*Ax + \frac{\beta}{2}\sum_{j=1}^{M}|x_j|^4 \quad \text{s.t.} \quad \|x\|_2 = 1,$$

where $M \in \mathbb{N}$, β are given constants and $A \in \mathbb{C}^{M\times M}$ is Hermitian. Consider the case that x and A are real. Since $x^\top x = 1$, multiplying the quadratic term of the objective function by $x^\top x$, we obtain the following equivalent problem

$$\min_{x\in\mathbb{R}^M} f(x) = \frac{1}{2}x^\top Axx^\top x + \frac{\beta}{2}\sum_{i=1}^{M}|x_i|^4 \quad \text{s.t.} \quad \|x\|_2 = 1.$$

The problem above can be also regarded as the best rank-1 tensor approximation of a fourth-order tensor $\mathcal{F}$ [7], with

$$\mathcal{F}_{\pi(i,j,k,l)} = \begin{cases} a_{kl}/4, & i = j = k \neq l, \\ a_{kl}/12, & i = j, i \neq k, i \neq l, k \neq l, \\ (a_{ii} + a_{kk})/12, & i = j \neq k = l, \\ a_{ii}/2 + \beta/4, & i = j = k = l, \\ 0, & \text{otherwise.} \end{cases}$$

For the complex case, we can obtain a best rank-1 complex tensor approximation problem by a similar fashion. Therefore, BEC is a polynomial optimization problem over single sphere.

2.6 Cryo-EM

The cryo-EM problem is to reconstruct a three-dimensional object from a series of two-dimensional projected images $\{P_i\}$ of the object. A classic model formulates it into an optimization problem over multiple orthogonality constraints [8] to compute the N corresponding directions $\{\tilde{R}_i\}$ of $\{P_i\}$, see Fig. 2. Each $\tilde{R}_i \in \mathbb{R}^{3\times 3}$ is a three-dimensional rotation, i.e., $\tilde{R}_i^\top \tilde{R}_i = I_3$ and $\det(\tilde{R}_i) = 1$. Let $\tilde{c}_{ij} = (x_{ij}, y_{ij}, 0)$ be the common line of P_i and P_j (viewed in P_i). If the data are exact, it follows from the Fourier projection-slice theorem [8], the common lines coincide, i.e.,

$$\tilde{R}_i\tilde{c}_{ij} = \tilde{R}_j\tilde{c}_{ji}.$$

Since the third column of $\tilde{R}_i^3$ can be represented by the first two columns $\tilde{R}_i^1$ and $\tilde{R}_i^2$ as $\tilde{R}_i^3 = \pm\tilde{R}_i^1 \times \tilde{R}_i^2$, the rotations $\{\tilde{R}_i\}$ can be compressed as a 3-by-2 matrix. Therefore, the corresponding optimization problem is

$$\min_{R_i} \quad \sum_{i=1}^{N}\rho(R_ic_{ij}, R_jc_{ji}) \quad \text{s.t.} \quad R_i^\top R_i = I_2, R_i \in \mathbb{R}^{3\times 2}, \tag{2.8}$$

where ρ is a function to measure the distance between two vectors, R_i are the first two columns of $\tilde{R}_i$, and c_{ij} are the first two entries of $\tilde{c}_{ij}$. In [8], the distance function is set as $\rho(u, v) = \|u - v\|_2^2$. An eigenvector relaxation and SDP relaxation are also presented in [8].

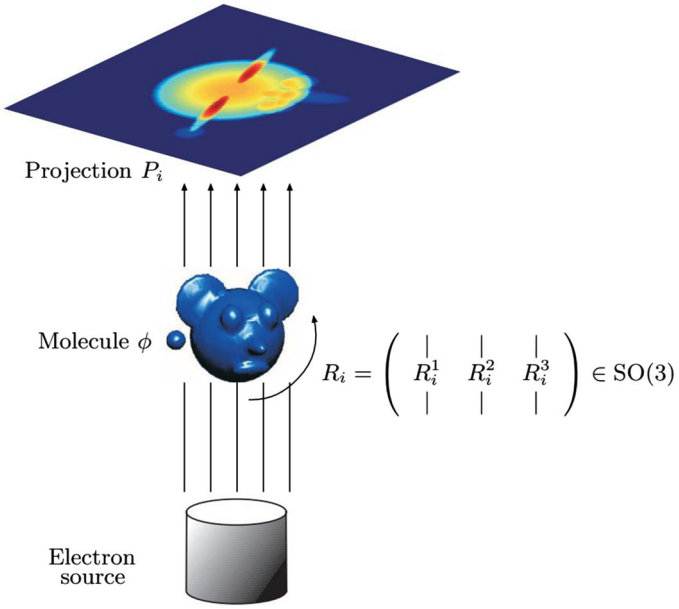

Fig. 2 Recover the 3-D structure from 2-D projections [8]

2.7 Linear Eigenvalue Problem

Linear eigenvalue decomposition and singular value decomposition are the special cases of optimization with orthogonality constraints. Linear eigenvalue problem can be written as

$$\min_{X \in \mathbb{R}^{n \times p}} \quad \mathrm{tr}(X^\top A X) \quad \text{s.t. } X^\top X = I, \tag{2.9}$$

where $A \in \mathbb{S}^n$ is given. Applications from low-rank matrix optimization, data mining, principal component analysis and high-dimensionality reduction techniques often need to deal with large-scale dense matrices or matrices with some special structures. Although modern computers are developing rapidly, most of the current eigenvalue and singular value decomposition softwares are limited by the traditional design and implementation. In particular, the efficiency may not be significantly improved when working with thousands of CPU cores. From the perspective of optimization, a series of fast algorithms for solving (2.9) were proposed in [9–12], whose essential parts can be divided into two steps, updating a subspace to approximate the eigenvector space better and extracting eigenvectors by the Rayleigh–Ritz (RR) process. The main numerical algebraic technique for updating subspaces is usually based on the Krylov subspace, which constructs a series of orthogonal bases sequentially. In [11], the authors proposed an equivalent unconstrained penalty function model

$$\min_{X \in \mathbb{R}^{n \times p}} \quad f_\mu(X) := \frac{1}{2}\mathrm{tr}(X^\top A X) + \frac{\mu}{4}\|X^\top X - I\|_F^2,$$

where μ is a parameter. By choosing an appropriate finite large μ, the authors established its equivalence with (2.9). When μ is chosen properly, the number of saddle points of this model is less than that of (2.9). More importantly, the model allows one to design an algorithm that uses only matrix–matrix multiplication. A Gauss–Newton algorithm for calculating low-rank decomposition is developed in [9]. When the matrix to be decomposed is of low rank, this algorithm can be more effective while its complexity is similar to the gradient method but with Q linear convergence. Because the bottleneck of many current iterative algorithms is the RR procedure of the eigenvalue decomposition of smaller dense matrices, the authors of [12] proposed a unified augmented subspace algorithmic framework. Each step iteratively solves a linear eigenvalue problem:

$$Y = \underset{X \in \mathbb{R}^{n \times p}}{\arg\min} \{\mathrm{tr}(X^\top A X) \ : X^\top X = I, \ X \in \mathcal{S}\},$$

where $\mathcal{S} := \mathrm{span}\{X, AX, A^2X, \cdots, A^kX\}$ with a small k (which can be far less than p). By combining with the polynomial acceleration technique and deflation in classical eigenvalue calculations, it needs only one RR procedure theoretically to reach a high accuracy.

When the problem dimension reaches the magnitude of $O(10^{42})$, the scale of data storage far exceeds the extent that traditional algorithms can handle. In [13], the authors consider to use a low-rank tensor format to express data matrices and eigenvectors. Let $N = n_1 n_2 \cdots n_d$ with positive integer $n_1, \cdots, n_d$. A vector $u \in \mathbb{R}^N$ can be reshaped as a tensor $\mathbf{u} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, whose entries $u_{i_1 i_2 \cdots i_d}$ are aligned in reverse lexicographical order, $1 \leqslant i_\mu \leqslant n_\mu$, $\mu = 1, 2, \cdots, d$. A tensor $\mathbf{u}$ can be written as the TT format if its entries can be represented by

$$u_{i_1 i_2 \cdots i_d} = U_1(i_1) U_2(i_2) \cdots U_d(i_d),$$

where $U_\mu(i_\mu) \in \mathbb{R}^{r_{\mu-1} \times r_\mu}, i_\mu = 1, 2, \cdots, n_\mu$ and fixed dimensions r_μ, $\mu = 0, 1, \cdots, d$ with $r_0 = r_d = 1$. In fact, the components r_μ, $\mu = 1, \cdots, d-1$ are often equal to a value r (r is then called the TT-rank). Hence, a vector u of dimension $\mathcal{O}(n^d)$ can be stored with $\mathcal{O}(dnr^2)$ entries if the corresponding tensor $\mathbf{u}$ has a TT format. A graphical representation of $\mathbf{u}$ can be seen in Fig. 3. The eigenvalue problem can be solved based on the subspace algorithm. By utilizing the alternating direction method with suitable truncations, the performance of the algorithm can be further improved.

The online singular value/eigenvalue decomposition appears in principal component analysis (PCA). The traditional PCA first reads the data and then performs eigenvalue decompositions on the sample covariance matrices. If the data are updated, the principal component vectors need be investigated again based on the new data. Unlike traditional PCA, the online PCA reads the samples one by one and updates the principal component vector in an iterative way, which is essentially a random iterative algorithm of the maximal trace optimization problem. As the sample grows, the online PCA algorithm leads to more accurate main components. An online PCA is proposed and analyzed in [14]. It is proved that the convergence rate is $O(1/n)$ with

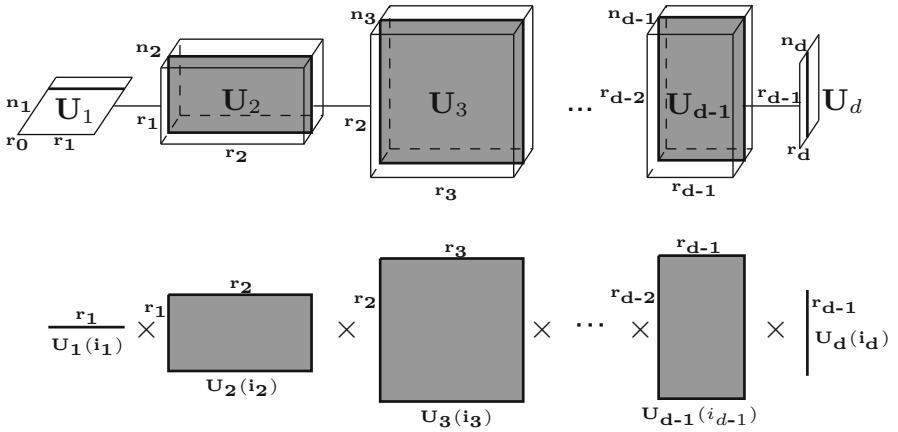

Fig. 3 Graphical representation of a TT tensor of order d with cores $\mathbf{U}_\mu$, $\mu = 1, 2, \cdots, d$. The first row is $\mathbf{u}$, and the second row are its entries $u_{i_1 i_2 \cdots i_d}$

high probability. A linear convergent VR-PCA algorithm is investigated in [15]. In [16], the scheme in [14] is further proved that under the assumption of sub-Gaussian's stochastic model, the convergence speed of the algorithm can reach the minimal bound of the information, and the convergence speed is near-global.

2.8 Nonlinear Eigenvalue Problem

The nonlinear eigenvalue problems from electronic structure calculations are another important source of problems with orthogonality constraints, such as the Kohn–Sham (KS) and Hartree–Fock (HF) energy minimization problems. By properly discretizing, the KS energy functional can be expressed as

$$E_{\mathrm{ks}}(X) := \frac{1}{4}\mathrm{tr}(X^* L X) + \frac{1}{2}\mathrm{tr}(X^* V_{\mathrm{ion}} X) + \frac{1}{2}\sum_l \sum_i \zeta_l |x_i^* w_l|^2 + \frac{1}{4}\rho^\top L^\dagger \rho + \frac{1}{2} e^\top \varepsilon_{\mathrm{xc}}(\rho),$$

where $X \in \mathbb{C}^{n\times p}$ satisfies $X^*X = I_p$, n is the spatial degrees of freedom, p is the total number of electron pairs, $\rho = \mathrm{diag}(XX^*)$ is the charge density and $\mu_{\mathrm{xc}}(\rho) := \frac{\partial \varepsilon_{\mathrm{xc}}(\rho)}{\partial \rho}$ and e is a vector in $\mathbb{R}^n$ with elements all of ones. More specifically, L is a finite-dimensional representation of the Laplacian operator, V_{ion} is a constant example, w_l represents a discrete reference projection function, ζ_l is a constant of ± 1, and $\varepsilon_{\mathrm{xc}}$ is used to characterize exchange-correlation energy. With the KS energy functional, the KS energy minimization problem is defined as

$$\min_{X\in\mathbb{C}^{n\times p}} \quad E_{\mathrm{ks}}(X) \quad \text{s.t.} \quad X^*X = I_p.$$

Compared to the KS density functional theory, the HF theory can provide a more accurate model. Specifically, it introduces a Fock exchange operator, which is a fourth-order tensor by some discretization, $\mathcal{V}(\cdot) : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$. The corresponding Fock energy can be expressed as

$$E_{\mathrm{f}} := \frac{1}{4}\left\langle \mathcal{V}(XX^*)X, X\right\rangle = \frac{1}{4}\left\langle \mathcal{V}(XX^*), XX^*\right\rangle.$$

The HF energy minimization problem is then

$$\min_{X\in\mathbb{C}^{n\times p}} \quad E_{\mathrm{hf}}(X) := E_{\mathrm{ks}}(X) + E_{\mathrm{f}}(X) \quad \text{s.t.} \quad X^*X = I_p. \tag{2.10}$$

The first-order optimality conditions of KS and HF energy minimization problems correspond to two different nonlinear eigenvalue problems. Taking KS energy minimization as an example, the first-order optimality condition is

$$H_{\mathrm{ks}}(\rho)X = X\Lambda, \quad X^*X = I_p, \tag{2.11}$$

where $H_{\mathrm{ks}}(\rho) := \frac{1}{2}L + V_{\mathrm{ion}} + \sum_l \zeta_l w_l w_l^* + \mathrm{diag}((\Re L^\dagger)\rho) + \mathrm{diag}(\mu_{\mathrm{xc}}(\rho)^* e)$ and Λ is a diagonal matrix. The equation (2.11) is also called the KS equation. The nonlinear eigenvalue problem aims to find some orthogonal eigenvectors satisfying (2.11), while the optimization problem with orthogonality constraints minimizes the objective function under the same constraints. These two problems are connected by the optimality condition and both describe the steady state of the physical system.

The most widely used algorithm for solving the KS equation is the so-called self-consistent field (SCF) iteration, which is to solve the following linear eigenvalue problems repeatedly

$$H_{\mathrm{ks}}(\rho_k)X_{k+1} = X_{k+1}\Lambda_{k+1}, \quad X_{k+1}^*X_{k+1} = I_p, \tag{2.12}$$

where $\rho_k = \mathrm{diag}(X_k X_k^*)$. In practice, to accelerate the convergence, we often replace the charge density ρ_k by a linear combination of the previously existing m charge densities

$$\rho_{\mathrm{mix}} = \sum_{j=0}^{m-1} \alpha_j \rho_{k-j}.$$

In the above expression, $\alpha = (\alpha_0, \alpha_1, \cdots, \alpha_{m-1})$ is the solution to the following minimization problem:

$$\min_{\alpha^\top e=1} \quad \|R\alpha\|^2,$$

where $R = (\Delta\rho_k, \Delta\rho_{k-1}, \cdots, \Delta\rho_{k-m+1})$, $\Delta\rho_j = \rho_j - \rho_{j-1}$ and e is an m-dimensional vector of all entries ones. After obtaining ρ_{mix}, we replace $H_{\mathrm{ks}}(\rho_k)$ in

(2.12) with $H_{\mathrm{ks}}(\rho_{\mathrm{mix}})$ and execute the iteration (2.12). This technique is called charge mixing. For more details, one can refer to [17–19].

Since SCF may not converge, many researchers have recently developed optimization algorithms for the electronic structure calculation that can guarantee convergence. In [20], the Riemannian gradient method is directly extended to solve the KS total energy minimization problem. The algorithm complexity is mainly from the calculation of the total energy and its gradient calculation, and the projection on the Stiefel manifold. Its complexity at each step is much lower than the linear eigenvalue problem, and it is easy to be parallelized. Extensive numerical experiments based on the software packages Octopus and RealSPACES show that the algorithm is often more efficient than SCF. In fact, the iteration (2.12) of SCF can be understood as an approximate Newton algorithm in the sense that the complicated part of the Hessian of the total energy is not considered:

$$\min_{X\in\mathbb{C}^{n\times p}} \quad q(X) := \frac{1}{2}\mathrm{tr}(X^* H_{\mathrm{ks}}(\rho_k)X) \quad \text{s.t.} \quad X^*X = I_p.$$

Since $q(X)$ is only a local approximation model of $E_{\mathrm{ks}}(X)$, there is no guarantee that the above model ensures a sufficient decrease of $E_{\mathrm{ks}}(X)$.

An explicit expression of the complicated part of the Hessian matrix is derived in [21]. Although this part is not suitable for an explicit storage, its operation with a vector is simple and feasible. Hence, the full Hessian matrix can be used to improve the reliability of Newton's method. By adding regularization terms, the global convergence is also guaranteed. A few other related works include [22–26].

The ensemble-based density functional theory is especially important when the spectrum of the Hamiltonian matrix has no significant gaps. The KS energy minimization model is modified by allowing the charge density to contain more wave functions. Specifically, denote by the single-particle wave functions $\psi_i(r),\ i = 1, \cdots, p'$ with $p' \geqslant p$. Then, the new charge density is defined as $\rho(r) = \sum_{i=1}^{p'} f_i|\psi_i(r)|^2$, where the fraction occupation $0 \leqslant f_i \leqslant 1$ is to ensure that the total charge density of the total orbit is p, i.e., $\sum_{i=1}^{p'} f_i = p$. To calculate the fractional occupancy, the energy functional in the ensemble model introduces a temperature T associated with an entropy $\alpha R(f)$, where $\alpha := \kappa_B T$, κ_B is the Boltzmann constant, $R(f) = \sum_{i=1}^{p'} s(f_i)$,

$$s(t) = \begin{cases} t\ln t + (1-t)\ln(1-t), & 0 < t < 1, \\ 0, & \text{otherwise.} \end{cases}$$

This method is often referred as the KS energy minimization model with temperature or the ensemble KS energy minimization model (EDFT). Similar to the KS energy minimization model, by using the appropriate discretization, the wave function can be represented with $X = [x_1, \cdots, x_{p'}] \in \mathbb{C}^{n\times p'}$. The discretized charge density in EDFT can be written as

$$\rho(X, f) := \mathrm{diag}(X\mathrm{diag}(f)X^*).$$

Obviously, $\rho(X, f)$ is real. The corresponding discretized energy functional is

$$M(X, f) = \mathrm{tr}(\mathrm{diag}(f)X^*AX) + \frac{1}{2}\rho^\top L^\dagger \rho + e^\top \varepsilon_{\mathrm{xc}}(\rho) + \alpha R(f).$$

The discretized EDFT model is

$$\begin{aligned} \min_{X\in\mathbb{C}^{n\times p'}, f\in\mathbb{R}^p} \quad & M(X, f) \\ \text{s.t.} \quad & X^*X = I_{p'}, \\ & e^\top f = p, \quad 0 \geqslant f \geqslant 1. \end{aligned} \tag{2.13}$$

Although SCF can be generalized to this model, its convergence is still not guaranteed. An equivalent simple model with only one-ball constraint is proposed in [27]. It is solved by a proximal gradient method where the terms other than the entropy function term are linearized. An explicit solution of the subproblem is then derived, and the convergence of the algorithm is established.

2.9 Approximation Models for Integer Programming

Many optimization problems arising from data analysis are NP-hard integer programmings. Spherical constraints and orthogonal constraints are often used to obtain approximate solutions with high quality. Consider optimization problem over the permutation matrices:

$$\min_{X\in\Pi_n} \quad f(X),$$

where $f(X) : \mathbb{R}^{n\times n} \to \mathbb{R}^n$ is differentiable, and Π_n is a collection of n-order permutation matrices

$$\Pi_n := \{X \in \mathbb{R}^{n\times n} \ : \ Xe = X^\top e = e, X_{ij} \in \{0, 1\}\}.$$

This constraint is equivalent to

$$\Pi_n := \{X \in \mathbb{R}^{n\times n} \ : \ X^\top X = I_n, X \geqslant 0\}.$$

It is proved in [28] that it is equivalent to an L_p-regularized optimization problem over the doubly stochastic matrices, which is much simpler than the original problem. An estimation of the lower bound of the nonzero elements at the stationary points is presented. Combining with the cutting plane method, a novel gradient-type algorithm with negative proximal terms is also proposed.

Given k communities $S_1, S_2, \cdots, S_k$ and the set of partition matrix P_n^k, where the partition matrix $X \in P_n^k$ means $X_{ij} = 1$, $i, j \in S_t$, $t \in \{1, \cdots, k\}$ and $X_{ij} = 0$ otherwise. Let A be the adjacency matrix of the network, $d_i = \sum_j A_{ij}, i \in \{1, \cdots, n\}$ and $\lambda = 1/\|d\|_2$. Define the matrix $C := -(A - \lambda dd^\top)$. The community detection

problem in social networks is to find a partition matrix to maximize the modularity function under the stochastic block model:

$$\min_{X} \quad \langle C, X \rangle \quad \text{s.t. } X \in P_n^k. \tag{2.14}$$

An SDP relaxation of (2.14) is

$$\begin{aligned} \min_{X} \quad & \langle C, X \rangle \\ \text{s.t.} \quad & X_{ii} = 1, i = 1, \cdots, n, \\ & 0 \leqslant X_{ij} \leqslant 1, \forall i, j, \\ & X \succeq 0. \end{aligned}$$

A sparse and low-rank completely positive relaxation technique is further investigated in [29] to transform the model into an optimization problem over multiple nonnegative spheres:

$$\begin{aligned} \min_{U \in \mathbb{R}^{n \times k}} \quad & \left\langle C, UU^\top \right\rangle \\ \text{s.t.} \quad & \|u_i\|_2 = 1, i = 1, \cdots, n, \\ & \|u_i\|_0 \leqslant p, i = 1, \cdots, n, \\ & U \geqslant 0, \end{aligned} \tag{2.15}$$

where u_i is the ith row of U, $1 \leqslant p \leqslant r$ is usually taken as a small number so that U can be stored for large-scale data sets. The equivalence to the original problem is proved theoretically, and an efficient row-by-row-type block coordinate descent method is proposed. In order to quickly solve network problems whose dimension is more than 10 million, an asynchronous parallel algorithm is further developed.

2.10 Deep Learning

Batch normalization is a very popular technique in deep neural networks. It avoids internal covariance translation by normalizing the input of each neuron. The space formed by its corresponding coefficient matrix can be regarded as a Riemannian manifold. For a deep neural network, batch normalization usually involves input processing before the nonlinear activation function. Define x and w as the outputs of the previous layer and the parameter vector for the current neuron, the batch normalization of $z := w^\top x$ can be written as

$$\mathrm{BN}(z) = \frac{z - \mathbf{E}(z)}{\mathrm{Var}(z)} = \frac{w^\top (x - \mathbf{E}(x))}{\sqrt{w^\top R_{xx} w}} = \frac{u^\top (x - \mathbf{E}(x))}{\sqrt{u^\top R_{xx} u}},$$

where $u := w/\|w\|$, $\mathbf{E}(z)$ is the expectation of random variable z and R_{xx} are the covariance matrices of x. From the definition, we have $\mathrm{BN}(w^\top x) = \mathrm{BN}(u^\top x)$ and

$$\frac{\partial \mathrm{BN}(w^\top x)}{\partial x} = \frac{\partial \mathrm{BN}(u^\top x)}{\partial x}, \quad \frac{\partial \mathrm{BN}(z)}{\partial w} = \frac{1}{w} \frac{\partial \mathrm{BN}(z)}{\partial u}.$$

Therefore, the use of the batch standardization ensures that the model does not explode with large learning rates and that the gradient is invariant to linear scaling during propagation.

Since $\mathrm{BN}(cw^\top x) = \mathrm{BN}(w^\top x)$ holds for any constant c , the optimization problem for deep neural networks using batch normalization can be written as

$$\min_{X \in \mathcal{M}} \quad L(X), \quad \mathcal{M} = S^{n_1-1} \times \cdots S^{n_m-1} \times \mathbb{R}^l,$$

where $L(X)$ is the loss function, S^{n-1} is a sphere in $\mathbb{R}^n$ (can also be viewed as a Grassmann manifold), $n_1, \cdots, n_m$ are the dimensions of the weight vectors, m is the number of weight vectors, and l is the number of remaining parameters to be decided, including deviations and other weight parameters. For more information, we refer to [30].

2.11 Sparse PCA

In the traditional PCA, the obtained principle eigenvectors are usually not sparse, which leads to high computational cost for computing the principle components. Spare PCA [31] wants to find principle eigenvectors with few nonzero elements. The mathematical formulation is

$$\begin{aligned} \min_{X \in \mathbb{R}^{n \times p}} \quad & -\mathrm{tr}(X^\top A^\top A X) + \rho \|X\|_1 \\ \text{s.t.} \quad & X^\top X = I_p, \end{aligned} \tag{2.16}$$

where $\|X\|_1 = \sum_{ij} |X_{ij}|$ and $\rho > 0$ is a trade-off parameter. When $\rho = 0$, this reduces to the traditional PCA problem. For $\rho > 0$, the term $\|X\|_1$ plays a role to promote sparsity. Problem (2.16) is a non-smooth optimization problem on the Stiefel manifold.

2.12 Low-Rank Matrix Completion

The low-rank matrix completion problem has important applications in computer vision, pattern recognitions, statistics, etc. It can be formulated as

$$\begin{aligned} \min_{X} \quad & \mathrm{rank}\,(X) \\ \text{s.t.} \quad & X_{ij} = A_{ij}, \ (i, j) \in \Omega, \end{aligned} \tag{2.17}$$

where X is the matrix that we want to recover (some of its entries are known) and Ω is the index set of observed entries. Due to the difficulty of the rank, a popular approach is to relax it into a convex model using the nuclear norm. The equivalence between this convex problem and the non-convex problem (2.17) is ensured under certain conditions. Another way is to use a low-rank decomposition on X and then solve the

corresponding unconstrained optimization problem [32]. If the rank of the ground-truth matrix A is known, an alternative model for a fixed-rank matrix completion is

$$\min_{X\in\mathbb{R}^{n\times p}} \quad \|\mathbf{P}_\Omega(X-A)\|_F^2 \text{ s.t. } \operatorname{rank}(X)=r, \tag{2.18}$$

where $\mathbf{P}_\Omega$ is a projection with $\mathbf{P}_\Omega(X)_{ij}=X_{ij},\ (i,j)\in\Omega$ and 0 otherwise, and $r=\operatorname{rank}(A)$. The set $\mathrm{Fr}(m,n,r):=\{X\in\mathbb{R}^{m\times n}:\ \operatorname{rank}(X)=r\}$ is a matrix manifold, called fixed-rank manifold. The related geometry is analyzed in [33]. Consequently, problem (2.18) can be solved by optimization algorithms on manifold. Problem (2.18) can deal with Gaussian noise properly. For data sets with a few outliers, the robust low-rank matrix completion problem (with the prior knowledge r) considers:

$$\min_{X\in\mathbb{R}^{n\times p}} \quad \|\mathbf{P}_\Omega(X-A)\|_1 \text{ s.t. } \operatorname{rank}(X)=r, \tag{2.19}$$

where $\|X\|_1=\sum_{i,j}|X_{ij}|$. Problem (2.19) is a non-smooth optimization problem on the fixed-rank matrix manifold. For some related algorithms for (2.18) and (2.19), the readers can refer to [34,35].

2.13 Sparse Blind Deconvolution

Blind deconvolution is to recover a convolution kernel $a_0\in\mathbb{R}^k$ and signal $x_0\in\mathbb{R}^m$ from their convolution

$$y=a_0\circledast x_0,$$

where $y\in\mathbb{R}^m$ and $\circledast$ represents some kind of convolution. Since there are infinitely many pairs (a_0,x_0) satisfying this condition, this problem is often ill conditioned. To overcome this issue, some regularization terms and extra constraints are necessary. The sphere-constrained sparse blind deconvolution reformulates the problem as

$$\min_{a,x} \quad \|y-a\circledast x\|_2^2+\mu\|x\|_1 \ \text{ s.t. } \ \|a\|_2=1,$$

where μ is a parameter to control the sparsity of the signal x. This is a non-smooth optimization problem on the product manifold of a sphere and $\mathbb{R}^m$. Some related background and the corresponding algorithms can be found in [36].

2.14 Nonnegative PCA

Since the principle eigenvectors obtained by the traditional PCA may not be sparse, one can enforce the sparsity by adding nonnegativity constraints. The problem is formulated as

$$\min_{X\in\mathbb{R}^{n\times p}} \quad \operatorname{tr}(X^\top AA^\top X) \ \text{ s.t. } \ X^\top X=I_p,\ X\geqslant 0, \tag{2.20}$$

where $A = [a_1, \cdots, a_k] \in \mathbb{R}^{n\times k}$ are given data points. Under the constraints, the variable X has at most one nonzero element in each row. This actually helps to guarantee the sparsity of the principle eigenvectors. Problem (2.20) is an optimization problem with manifold and nonnegative constraints. Some related information can be found in [37,38].

2.15 *K*-Means Clustering

K-means clustering is a fundamental problem in data mining. Given n data points $(x_1, x_2, \cdots, x_n)$ where each data point is a d-dimensional vector, k-means is to partition them into k clusters $S := \{S_1, S_2, \cdots, S_k\}$ such that the within-cluster sum of squares is minimized. Each data point belongs to the cluster with the nearest mean. The mathematical form is

$$\min_{S} \sum_{i=1}^{k} \sum_{x\in S_i} \|x - c_i\|^2, \tag{2.21}$$

where $c_i = \frac{1}{\mathrm{card}(S_i)} \sum_{x\in S_i} x$ is the center of ith cluster and $\mathrm{card}(S_i)$ is the cardinality of S_i. Equivalently, problem (2.21) can be written as [39–41]:

$$\begin{aligned} \min_{Y\in\mathbb{R}^{n\times k}} \quad & \mathrm{tr}(Y^\top D Y) \\ \text{s.t.} \quad & YY^\top \mathbf{1} = \mathbf{1}, \\ & Y^\top Y = I_k,\ Y \geqslant 0, \end{aligned} \tag{2.22}$$

where $D_{ij} := \|x_i - x_j\|^2$ is the squared Euclidean distance matrix. Problem (2.22) is a minimization over the Stiefel manifold with linear constraints and nonnegative constraints.

3 Algorithms for Manifold Optimization

In this section, we introduce a few state-of-the-art algorithms for optimization problems on Riemannian manifold. Let us start from the concepts of manifold optimization.

3.1 Preliminaries on Riemannian Manifold

A d-dimensional manifold $\mathcal{M}$ is a Hausdorff and second-countable topological space, which is homeomorphic to the d-dimensional Euclidean space locally via a family of charts. When the transition maps of intersecting charts are smooth, the manifold $\mathcal{M}$ is called a smooth manifold. Intuitively, the tangent space $T_x\mathcal{M}$ at a point x of a manifold $\mathcal{M}$ is the set of the tangent vectors of all the curves at x. Mathematically, a tangent vector ξ_x to $\mathcal{M}$ at x is a mapping such that there exists a curve γ on $\mathcal{M}$ with $\gamma(0) = x$, satisfying

$$\xi_x u := \dot{\gamma}(0)u \triangleq \left.\frac{\mathrm{d}(u(\gamma(t)))}{\mathrm{d}t}\right|_{t=0}, \quad \forall\, u \in \Im_x(\mathcal{M}),$$

where $\Im_x(\mathcal{M})$ is the set of all real-valued functions f defined in a neighborhood of x in $\mathcal{M}$. Then, the tangent space $T_x\mathcal{M}$ to $\mathcal{M}$ is defined as the set of all tangent vectors to $\mathcal{M}$ at x. If $\mathcal{M}$ is equipped with a smoothly varied inner product $g_x(\cdot,\cdot) := \langle\cdot,\cdot\rangle_x$ on the tangent space, then $(\mathcal{M}, g)$ is a Riemannian manifold. In practice, different Riemannian metrics may be investigated to design efficient algorithms. The Riemannian gradient $\operatorname{grad} f(x)$ of a function f at x is an unique vector in $T_x\mathcal{M}$ satisfying

$$\langle \operatorname{grad} f(x), \xi\rangle_x = Df(x)[\xi], \quad \forall \xi \in T_x\mathcal{M}, \tag{3.1}$$

where $Df(x)[\xi]$ is the derivative of $f(\gamma(t))$ at $t = 0$, $\gamma(t)$ is any curve on the manifold that satisfies $\gamma(0) = x$ and $\dot{\gamma}(0) = \xi$. The Riemannian Hessian $\operatorname{Hess} f(x)$ is a mapping from the tangent space $T_x\mathcal{M}$ to the tangent space $T_x\mathcal{M}$:

$$\operatorname{Hess} f(x)[\xi] := \tilde{\nabla}_\xi \operatorname{grad} f(x), \tag{3.2}$$

where $\tilde{\nabla}$ is the Riemannian connection [42]. For a function f defined on a submanifold $\mathcal{M}$ with the Euclidean metric on its tangent space, if it can be extended to the ambient Euclidean space $\mathbb{R}^{n\times p}$, we have its Riemannian gradient $\operatorname{grad} f$ and Riemannian Hessian $\operatorname{Hess} f$:

$$\begin{aligned} \operatorname{grad} f(x) &= \mathbf{P}_{T_x\mathcal{M}}(\nabla f(x)), \\ \operatorname{Hess} f(x)[u] &= \mathbf{P}_{T_x\mathcal{M}}(D\operatorname{grad} f(x)[u]),\ u \in T_x\mathcal{M}, \end{aligned} \tag{3.3}$$

where D is the Euclidean derivative and $\mathbf{P}_{T_x\mathcal{M}}(u) := \arg\min_{z\in T_x\mathcal{M}} \|x - z\|^2$ denotes the projection operator to $T_x\mathcal{M}$. When $\mathcal{M}$ is a quotient manifold whose total space is a submanifold of an Euclidean space, the tangent space in the expression (3.3) should be replaced by its horizontal space. According to (3.1) and (3.2), different Riemannian metrics will lead to different expressions of Riemannian gradient and Hessian. More detailed information on the related backgrounds can be found in [42].

We next briefly introduce some typical manifolds, where the Euclidean metric on the tangent space is considered.

- Sphere [42] $\operatorname{Sp}(n-1)$. Let $x(t)$ with $x(0) = x$ be a curve on sphere, i.e., $x(t)^\top x(t) = 1$ for all t. Taking the derivatives with respect to t, we have

$$\dot{x}(t)^\top x(t) + x(t)^\top \dot{x}(t) = 0.$$

  At $t = 0$, we have $\dot{x}(0)x + x^\top \dot{x}(0) = 0$. Hence, the tangent space is

$$T_x\operatorname{Sp}(n-1) = \{z \in \mathbb{R}^n \ :\ z^\top x = 0\}.$$

  The projection operator is defined as

$$\mathbf{P}_{T_x\operatorname{Sp}(n-1)}(z) = (I - xx^\top)z.$$

For a function defined on $\mathrm{Sp}(n-1)$ with respect to the Euclidean metric $g_x(u, v) = u^\top v,\ u, v \in T_x\mathrm{Sp}(n-1)$, its Riemannian gradient and Hessian at x can be represented by

$$\begin{aligned}\mathrm{grad}\, f(x) &= \mathbf{P}_{T_x\mathrm{Sp}(n-1)}(\nabla f(x)),\\ \mathrm{Hess}\, f(x)[u] &= \mathbf{P}_{T_x\mathrm{Sp}(n-1)}(\nabla^2 f(x)[u] - ux^\top \nabla f(x)),\ u \in T_x\mathrm{Sp}(n-1).\end{aligned}$$

- Stiefel manifold [42] $\mathrm{St}(n, p) := \{X \in \mathbb{R}^{n\times p} : X^\top X = I_p\}$. By a similar calculation as the spherical case, we have its tangent space:

$$T_X\mathrm{St}(n, p) = \{Z \in \mathbb{R}^{n\times p} : Z^\top X + X^\top Z = 0\}.$$

The projection operator onto $T_X\mathrm{St}(n, p)$ is

$$\mathbf{P}_{T_X\mathrm{St}(n,p)}(Z) = Z - X\mathrm{sym}(X^\top Z),$$

where $\mathrm{sym}(Z) := (Z+Z^\top)/2$. Given a function defined on $\mathrm{St}(n, p)$ with respect to the Euclidean metric $g_X(U, V) = \mathrm{tr}(U^\top V),\ U, V \in T_X\mathrm{St}(n, p)$, its Riemannian gradient and Hessian at X can be represented by

$$\begin{aligned}\mathrm{grad}\, f(X) &= \mathbf{P}_{T_X\mathrm{St}(n,p)}(\nabla f(X)),\\ \mathrm{Hess}\, f(X)[U] &= \mathbf{P}_{T_X\mathrm{St}(n,p)}(\nabla^2 f(X)[U] - U\mathrm{sym}(X^\top \nabla f(X))),\ U \in T_X\mathrm{St}(n, p).\end{aligned}$$

- Oblique manifold [43] $\mathrm{Ob}(n, p) := \{X \in \mathbb{R}^{n\times p} \mid \mathrm{diag}(X^\top X) = e\}$. Its tangent space is

$$T_X\mathrm{Ob}(n, p) = \{Z \in \mathbb{R}^{n\times p} : \mathrm{diag}(X^\top Z) = 0\}.$$

The projection operator onto $T_X\mathrm{Ob}(n, p)$ is

$$\mathbf{P}_{T_X\mathrm{Ob}(n,p)} = Z - X\mathrm{Diag}(\mathrm{diag}(X^\top Z)).$$

Given a function defined on $\mathrm{Ob}(n, p)$ with respect to the Euclidean metric, its Riemannian gradient and Hessian at X can be represented by

$$\begin{aligned}\mathrm{grad}\, f(X) &= \mathbf{P}_{T_X\mathrm{Ob}(n,p)}(\nabla f(X)),\\ \mathrm{Hess}\, f(X)[U] &= \mathbf{P}_{T_X\mathrm{Ob}(n,p)}(\nabla^2 f(X)[U] - U\mathrm{Diag}(\mathrm{diag}(X^\top \nabla f(X)))),\end{aligned}$$

with $U \in T_X\mathrm{Ob}(n, p)$.

- Grassmann manifold [42] $\mathrm{Grass}(n, p) := \{\mathrm{span}(X) : X \in \mathbb{R}^{n\times p}, X^\top X = I_p\}$. It denotes the set of all p-dimensional subspaces of $\mathbb{R}^n$. This manifold is different from other manifolds mentioned above. It is a quotient manifold since each element is an equivalent class of $n \times p$ matrices. From the definition of $\mathrm{Grass}(p, n)$, the equivalence relation $\sim$ is defined as

$$X \sim Y \Leftrightarrow \exists Q \in \mathbb{R}^{p\times p} \text{ with } Q^\top Q = QQ^\top = I \text{ s.t. } Y = XQ.$$

Its element is of the form

$$[X] := \{Y \in \mathbb{R}^{n\times p} : Y^\top Y = I, Y \sim X\}.$$

Then, $\mathrm{Grass}(n, p)$ is a quotient manifold of $\mathrm{St}(n, p)$, i.e., $\mathrm{St}(n, p)/\sim$. Due to this equivalence, a tangent vector ξ of $T_X\mathrm{Grass}(n, p)$ may have many different representations in its equivalence class. To find the unique representation, a horizontal space [42, Section 3.5.8] is introduced. For a given $X \in \mathbb{R}^{n\times p}$ with $X^\top X = I_p$, the horizontal space is

$$\mathcal{H}_X\mathrm{Grass}(n, p) = \{Z \in \mathbb{R}^{n\times p} \ : \ Z^\top X = 0\}.$$

Here, a function of the horizontal space is similar to the tangent space when computing the Riemannian gradient and Hessian. We have the projection onto the horizontal space

$$\mathbf{P}_{\mathcal{H}_X\mathrm{Grass}(n,p)}(Z) = Z - XX^\top Z.$$

Given a function defined on $\mathrm{Grass}(n, p)$ with respect to the Euclidean metric $g_X = \mathrm{tr}(U^\top V),\ U, V \in \mathcal{H}_X\mathrm{Grass}(n, p)$, its Riemannian gradient and Hessian at X can be represented by

$$\begin{aligned}\mathrm{grad}\, f(X) &= \mathbf{P}_{\mathcal{H}_X\mathrm{Grass}(n,p)}(\nabla f(X)),\\ \mathrm{Hess}\, f(X)[U] &= \mathbf{P}_{\mathcal{H}_X\mathrm{Grass}(n,p)}(\nabla^2 f(X)[U] - UX^\top \nabla f(X)),\ U \in T_X\mathrm{Grass}(n, p).\end{aligned}$$

- Fixed-rank manifold [33] $\mathrm{Fr}(n, p, r) := \{X \in \mathbb{R}^{n\times p} \ : \ \mathrm{rank}\,(X) = r\}$ is a set of all $n \times p$ matrices of rank r. Using the singular value decomposition (SVD), this manifold can be represented equivalently by

$$\mathrm{Fr}(n, p, r) = \{U\Sigma V^\top \ : \ U \in \mathrm{St}(n, r),\ V \in \mathrm{St}(p, r),\ \Sigma = \mathrm{diag}(\sigma_i)\},$$

where $\sigma_1 \geqslant \cdots \geqslant \sigma_r > 0$. Its tangent space at $X = U\Sigma V^\top$ is

$$\begin{aligned}T_X\mathrm{Fr}(n, p, r) &= \left\{[U, U_\perp]\begin{pmatrix}\mathbb{R}^{r\times r} & \mathbb{R}^{r\times(p-r)}\\ \mathbb{R}^{(n-r)\times r} & 0_{(n-r)\times(p-r)}\end{pmatrix}[V, V_\perp]^\top\right\}\\ &= \{UMV^\top + U_pV^\top + UV_p^\top \ : \ M \in \mathbb{R}^{r\times r},\\ &\qquad U_p \in \mathbb{R}^{n\times r}, U_p^\top U = 0, V_p \in \mathbb{R}^{p\times r},\ V_p^\top V = 0\},\end{aligned} \tag{3.4}$$

where $U_\perp$ and $V_\perp$ are the orthogonal complements of U and V, respectively. The projection operator onto the tangent space is

$$\mathbf{P}_{T_X\mathrm{Fr}(n,p,r)}(Z) = P_UZP_V + P_U^\perp ZP_V + P_UZP_V^\perp,$$

where $P_U = UU^\top$ and $P_U^\perp = I - P_U$. Comparing the representation with (3.4), we have

$$M(Z; X) := U^\top ZV,\ U_p(Z; X) = P_U^\perp ZV,\ V_p(Z; X) = P_V^\perp Z^\top U.$$

Given a function defined on $\mathrm{Fr}(n, p, r)$ with respect to the Euclidean metric $g_X(U, V) = \mathrm{tr}(U^\top V)$, its Riemannian gradient and Hessian at $X = U\Sigma V^\top$ can be represented by

$$\begin{aligned}\mathrm{grad}\, f(X) &= \mathbf{P}_{T_X\mathrm{Fr}(n,p,r)}(\nabla f(X)),\\ \mathrm{Hess}\, f(X)[H] &= U\hat{M}V^\top + \hat{U}_p V^\top + U\hat{V}_p^\top,\ H \in T_X\mathrm{Fr}(n, p, r),\end{aligned}$$

where

$$\begin{aligned}\hat{M} &= M(\nabla^2 f(X)[H]; X),\\ \hat{U}_p &= U_p(\nabla^2 f(X)[H]; X) + P_U^\perp \nabla f(X) V_p(H; X)/\Sigma,\\ \hat{V}_p &= V_p(\nabla^2 f(X)[H]; X) + P_V^\perp \nabla f(X) U_p(H; X)/\Sigma.\end{aligned}$$

- The set of symmetric positive definite matrices [44], i.e., $\mathrm{SPD}(n) = \{X \in \mathbb{R}^{n\times n} : X^\top = X,\ X \succ 0\}$ is a manifold. Its tangent space at X is

$$T_X\mathrm{SPD}(n) = \{Z \in \mathbb{R}^{n\times n} : Z^\top = Z\}.$$

We have the projection onto $T_X\mathrm{SPD}(n)$:

$$\mathbf{P}_{T_X\mathrm{SPD}(n)}(Z) = (Z^\top + Z)/2.$$

Given a function defined on $\mathrm{SPD}(n, p)$ with respect to the Euclidean metric $g_X(U, V) = \mathrm{tr}(U^\top V)$, $U, V \in T_X\mathrm{SPD}(n)$, its Riemannian gradient and Hessian at X can be represented by

$$\begin{aligned}\mathrm{grad}\, f(X) &= \mathbf{P}_{T_X\mathrm{SPD}(n)}(\nabla f(X)),\\ \mathrm{Hess}\, f(X)[U] &= \mathbf{P}_{T_X\mathrm{SPD}(n)}(\nabla^2 f(X)[U]),\ U \in T_X\mathrm{SPD}(n).\end{aligned}$$

- The set of rank-r symmetric positive semidefinite matrices [45,46], i.e., $\mathrm{FrPSD}(n, r) = \{X \in \mathbb{R}^{n\times n} : X = X^\top,\ X \succeq 0,\ \mathrm{rank}\,(X) = r\}$. This manifold can be reformulated as

$$\mathrm{FrPSD}(n, r) = \{YY^\top : Y \in \mathbb{R}^{n\times r},\ \mathrm{rank}\,(Y) = k\},$$

which is a quotient manifold. The horizontal space at Y is

$$T_Y\mathcal{H}_{\mathrm{FrPSD}(n,r)} = \{Z \in \mathbb{R}^{n\times r} : Z^\top Y = Y^\top Z\}.$$

We have the projection operator onto $T_Y\mathcal{H}_{\mathrm{FrPSD}(n,r)}$

$$\mathbf{P}_{T_Y\mathcal{H}_{\mathrm{FrPSD}(n,r)}}(Z) = Z - Y\Omega,$$

where the skew-symmetric matrix Ω is the unique solution of the Sylvester equation $\Omega(Y^\top Y) + (Y^\top Y)\Omega = Y^\top Z - Z^\top Y$. Given a function f with respect to the Euclidean metric $g_Y(U, V) = \mathrm{tr}(U^\top V)$, $U, V \in T_Y\mathcal{H}_{\mathrm{FrPSD}(n,r)}$, its Riemannian gradient and Hessian can be represented by

$$\begin{aligned}\mathrm{grad}\, f(Y) &= \nabla f(Y),\\ \mathrm{Hess}\, f(X)[U] &= \mathbf{P}_{T_Y\mathcal{H}_{\mathrm{FrPSD}(n,r)}}(\nabla^2 f(Y)[U]),\ U \in T_Y\mathcal{H}_{\mathrm{FrPSD}(n,r)}.\end{aligned}$$

3.2 Optimality Conditions

We next present the optimality conditions for manifold optimization problem in the following form

$$\begin{aligned}\min_{x\in\mathcal{M}}\quad & f(x)\\ \text{s.t.}\quad & c_i(x) = 0,\ i \in \mathcal{E} := \{1, \cdots, \ell\},\\ & c_i(x) \geqslant 0,\ i \in \mathcal{I} := \{\ell+1, \cdots, m\},\end{aligned} \tag{3.5}$$

where $\mathcal{E}$ and $\mathcal{I}$ denote the index sets of equality constraints and inequality constraints, respectively, and $c_i : \mathcal{M} \to \mathbb{R}$, $i \in \mathcal{E} \cup \mathcal{I}$ are smooth functions on $\mathcal{M}$. We mainly adopt the notions in [47]. By keeping the manifold constraint, the Lagrangian function of (3.5) is

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i\in\mathcal{E}\cup\mathcal{I}} \lambda_i c_i(x), \quad x \in \mathcal{M},$$

where λ_i, $i \in \mathcal{E} \cup \mathcal{I}$ are the Lagrangian multipliers. Here, we notice that the domain of $\mathcal{L}$ is on the manifold $\mathcal{M}$. Let $\mathcal{A}(x) := \mathcal{E} \cup \{i \in \mathcal{I} : c_i(x) = 0\}$. Then the linear independence constraint qualifications (LICQ) for problem (3.5) holds at x if and only if

$$\mathrm{grad}\, c_i(x),\ i \in \mathcal{A}(x) \text{ is linear independent on } T_x\mathcal{M}.$$

Then, the first-order necessary conditions can be described as follows:

Theorem 3.1 (First-order necessary optimality conditions (KKT conditions)) *Suppose that x^* is a local minimum of (3.5) and that the LICQ holds at x^*, then there exist Lagrangian multipliers $\lambda_i^*, i \in \mathcal{E} \cup \mathcal{I}$ such that the following KKT conditions hold*:

$$\begin{aligned}\mathrm{grad}\, f(x^*) + \sum_{i\in\mathcal{E}\cup\mathcal{I}} \lambda_i^* \mathrm{grad}\, c_i(x^*) = 0,&\\ c_i(x^*) = 0, \quad \forall i \in \mathcal{E},&\\ c_i(x^*) \geqslant 0,\ \lambda_i^* \geqslant 0,\ \lambda_i^* c_i(x^*) = 0, \quad \forall i \in \mathcal{I}.&\end{aligned} \tag{3.6}$$

Let x^* and λ_i^*, $i \in \mathcal{E} \cup \mathcal{I}$ be one of the solution of the KKT conditions (3.6). Similar to the case without the manifold constraint, we define a critical cone $\mathcal{C}(x^*, \lambda^*)$ as

$$w \in \mathcal{C}(x^*, \lambda^*) \Leftrightarrow \begin{cases} w \in T_{x^*}\mathcal{M}, \\ \langle \operatorname{grad} c_i(x^*), w \rangle = 0, \ \forall i \in \mathcal{E}, \\ \langle \operatorname{grad} c_i(x^*), w \rangle = 0, \ \forall i \in \mathcal{A}(x^*) \cap \mathcal{I} \text{ with } \lambda_i^* > 0, \\ \langle \operatorname{grad} c_i(x^*), w \rangle \geqslant 0, \ \forall i \in \mathcal{A}(x^*) \cap \mathcal{I} \text{ with } \lambda_i^* = 0. \end{cases}$$

Then, we have the following second-order necessary and sufficient conditions.

Theorem 3.2 (Second-order optimality conditions)

- *Second-order necessary conditions*:
  Suppose that x^* is a local minimum of (3.5) and the LICQ holds at x^*. Let λ^* be the multipliers such that the KKT conditions (3.6) hold. Then, we have

$$\left\langle \operatorname{Hess} \mathcal{L}(x^*, \lambda^*)[w], w \right\rangle \geqslant 0, \ \forall w \in \mathcal{C}(x^*, \lambda^*),$$

  where $\operatorname{Hess} \mathcal{L}(x^*, \lambda^*)$ *is the Riemannian Hessian of $\mathcal{L}$ with respect to x at (x^*, λ^*).*
- *Second-order sufficient conditions*:
  Suppose that x^* and λ^* satisfy the KKT conditions (3.6). If we further have

$$\left\langle \operatorname{Hess} \mathcal{L}(x^*, \lambda^*)[w], w \right\rangle > 0, \ \forall w \in \mathcal{C}(x^*, \lambda^*), \ w \neq 0,$$

  then x^* is a strict local minimum of (3.6).

Suppose that we have only the manifold constraint, i.e., $\mathcal{E} \cup \mathcal{I}$ is empty. For a smooth function f on the manifold $\mathcal{M}$, the optimality conditions take a similar form to the Euclidean unconstrained case. Specifically, if x^* is a first-order stationary point, then it holds that

$$\operatorname{grad} f(x^*) = 0.$$

If x^* is a second-order stationary point, then

$$\operatorname{grad} f(x^*) = 0, \quad \operatorname{Hess} f(x^*) \succeq 0.$$

If x^* satisfies

$$\operatorname{grad} f(x^*) = 0, \quad \operatorname{Hess} f(x^*) \succ 0,$$

then x^* is a strict local minimum. For more details, we refer the reader to [47].

3.3 First-Order-Type Algorithms

From the perspective of Euclidean constrained optimization problems, there are many standard algorithms which can solve this optimization problem on manifold. However, since the intrinsic structure of manifolds is not considered, these algorithms may not be effective in practice. By doing curvilinear search along the geodesic, a globally convergent gradient descent method is proposed in [48]. For Riemannian conjugate gradient (CG) methods [49], the parallel translation is used to construct the conjugate directions. Due to the difficulty of calculating geodesics (exponential maps) and parallel translations, computable retraction and vector transport operators are proposed to approximate the exponential map and the parallel translation [42]. Therefore, more general Riemannian gradient descent methods and CG methods together with convergence analysis are obtained in [42]. These algorithms have been successfully applied to various applications [33,50]. Numerical experiments exhibit the advantage of using geometry of the manifold. A proximal Riemannian gradient method is proposed in [51]. Specifically, the objective function is linearized using the first-order Taylor expansion on manifold and a proximal term is added. The original problem is then transformed into a series of projection problems on the manifold. For general manifolds, the existence and uniqueness of the projection operator cannot be guaranteed. But when the given manifold satisfies certain differentiable properties, the projection operator is always locally well defined and is also a specific retraction operator [52]. Therefore, in this case, the proximal Riemannian gradient method coincides with the Riemannian gradient method. By generalizing the adaptive gradient method in [53], an adaptive gradient method on manifold is also presented in [51]. In particular, optimization over Stiefel manifold is an important special case of Riemannian optimization. Various efficient retraction operators, vector transport operators and Riemannian metric have been investigated to construct more practical gradient descent and CG methods [54–56]. The extrapolation technique is adopted to accelerate gradient-type methods on Stiefel manifold in [57]. Non-retraction-based first-order methods are also developed in [25].

We next present a brief introduction of first-order algorithms for manifold optimization. Let us start with the retraction operator R. It is a smooth mapping from the tangent bundle $T\mathcal{M} := \cup_{x\in\mathcal{M}} T_x\mathcal{M}$ to $\mathcal{M}$ and satisfies

- $R_x(0_x) = x$, 0_x is the zero element in the tangent space $T_x\mathcal{M}$,
- $DR_x(0_x)[\xi] = \xi,\ \forall \xi \in T_x\mathcal{M}$,

where R_x is the retraction operator R at x. The well-posedness of the retraction operator is shown in Section 4.1.3 of [42]. The retraction operator provides an efficient way to pull the points from the tangent space back onto the manifold. Let $\xi_k \in T_x\mathcal{M}$ be a descent direction, i.e., $\langle \text{grad } f(x_k), \xi_k\rangle_{x_k} < 0$. Another important concept on manifold is the vector transport operator $\mathcal{T}$. It is a smooth mapping from the product of tangent bundles $T\mathcal{M} \bigoplus T\mathcal{M}$ to the tangent bundle $T\mathcal{M}$ and satisfies the following properties.

- There exists a retraction R associated with $\mathcal{T}$, i.e.,

$$\mathcal{T}_{\eta_x}\xi_x \in T_{R_x(\eta_x)}\mathcal{M}.$$

- $\mathcal{T}_{0_x}\xi_x = \xi_x$ for all $x \in \mathcal{M}$ and $\xi_x \in T_x\mathcal{M}$.
- $\mathcal{T}_{\eta_x}(a\xi_x + b\zeta_x) = a\mathcal{T}_{\eta_x}\xi_x + b\mathcal{T}_{\eta_x}\zeta_x$.

The vector transport is a generalization of the parallel translation [42, Section 5.4]. The general feasible algorithm framework on the manifold can be expressed as

$$x_{k+1} = R_{x_k}(t_k\xi_k), \tag{3.7}$$

where t_k is a well-chosen step size. Similar to the line search method in Euclidean space, the step size t_k can be obtained by the curvilinear search on the manifold. Here, we take the Armijo search as an example. Given $\rho, \delta \in (0, 1)$, the monotone and non-monotone search try to find the smallest nonnegative integer h such that

$$f(R_{x_k}(t_k\xi_k)) \leqslant f(x_k) + \rho t_k \langle \operatorname{grad} f(x_k), \xi_k\rangle_{x_k}, \tag{3.8}$$

$$f(R_{x_k}(t_k\xi_k)) \leqslant C_k + \rho t_k \langle \operatorname{grad} f(x_k), \xi_k\rangle_{x_k}, \tag{3.9}$$

respectively, where $t_k = \gamma_k\delta^h$ and γ_k is an initial step size. The reference value C_{k+1} is a convex combination of C_k and $f(x_{k+1})$ and is calculated via $C_{k+1} = (\varrho Q_k C_k + f(x_{k+1}))/Q_{k+1}$, where $\varrho \in [0, 1]$, $C_0 = f(x_0)$, $Q_{k+1} = \varrho Q_k + 1$ and $Q_0 = 1$. From the Euclidean optimization, we know that the Barzilai–Borwein (BB) step size often accelerates the convergence. The BB step size can be generalized to Riemannian manifold [51] as

$$\gamma_k^{(1)} = \frac{\langle s_{k-1}, s_{k-1}\rangle_{x_k}}{|\langle s_{k-1}, v_{k-1}\rangle_{x_k}|} \quad \text{or} \quad \gamma_k^{(2)} = \frac{|\langle s_{k-1}, v_{k-1}\rangle_{x_k}|}{\langle v_{k-1}, v_{k-1}\rangle_{x_k}}, \tag{3.10}$$

where

$$s_{k-1} = -t_{k-1}\cdot\mathcal{T}_{x_{k-1}\to x_k}(\operatorname{grad} f(x_{k-1})), \quad v_{k-1} = \operatorname{grad} f(x_k) + t_{k-1}^{-1}\cdot s_{k-1},$$

and $\mathcal{T}_{x_{k-1}\to x_k} : T_{x_{k-1}}\mathcal{M} \mapsto T_{x_k}\mathcal{M}$ denotes an appropriate vector transport mapping connecting x_{k-1} and x_k; see [42,58]. When $\mathcal{M}$ is a submanifold of an Euclidean space, the Euclidean differences $s_{k-1} = x_k - x_{k-1}$ and $v_{k-1} = \operatorname{grad} f(x_k) - \operatorname{grad} f(x_{k-1})$ are an alternative choice if the Euclidean inner product is used in (3.10). This choice is often attractive since the vector transport is not needed [51,54]. We note that the differences between first- and second-order algorithms are mainly due to their specific ways of acquiring ξ_k.

In practice, the computational cost and convergence behavior of different retraction operators differ a lot. Similarly, the vector transport plays an important role in CG methods and quasi-Newton methods (we will introduce them later). There are many studies on the retraction operators and vector transports. Here, we take the Stiefel manifold $\mathrm{St}(n, p)$ as an example to introduce several different retraction operators at the current point X for a given step size τ and descent direction $-D$.

- Exponential map [59]

$$R_X^{\mathrm{geo}}(-\tau D) = \begin{bmatrix} X, & Q\end{bmatrix}\exp\left(\tau\begin{bmatrix} -X^\top D & -R^\top \\ R & 0\end{bmatrix}\right)\begin{bmatrix} I_p \\ 0\end{bmatrix},$$

where $QR = -(I_n - XX^\top)D$ is the QR decomposition of $-(I_n - XX^\top)D$. This scheme needs to calculate an exponent of a $2p$-by-$2p$ matrix and a QR decomposition of an n-by-p matrix. From [59], an explicit form of parallel translation is unknown.

- Cayley transform [21]

$$R_X^{\mathrm{wy}}(-\tau D) = X - \tau U\left(I_{2p} + \frac{\tau}{2}V^\top U\right)^{-1} V^\top X, \tag{3.11}$$

where $U = [P_X D,\ X]$, $V = [X,\ -P_X D] \in \mathbb{R}^{n\times(2p)}$ with $P_X := (I - \frac{1}{2}XX^\top)$. When $p < n/2$, this scheme is much cheaper than the exponential map. The associated vector transport is [56]

$$\mathcal{T}_{\eta_X}^{\mathrm{wy}}(\xi_X) = \left(I - \frac{1}{2}W_{\eta_X}\right)^{-1}\left(I + \frac{1}{2}W_{\eta_X}\right)\xi_X,\ W_{\eta_X} = P_X\eta_X X - X\eta_X P_X,$$

- Polar decomposition [42]

$$R_X^{\mathrm{pd}}(-\tau D) = (X - \tau D)(I_p + \tau^2 D^\top D)^{-1/2}.$$

The computational cost is lower than the Cayley transform, but the Cayley transform may give a better approximation to the exponential map [60]. The associated vector transport is then defined as [61]

$$\mathcal{T}_{\eta_X}^{\mathrm{pd}}\xi_X = Y\Omega + (I - YY^\top)\xi_X(Y^\top(X + \eta_X))^{-1},$$

where $Y = R_X\eta_X$ and $\mathrm{vec}(\Omega) = (Y^\top(X + \eta_X)) \oplus (Y^\top(X + \eta_X))^{-1}\mathrm{vec}(Y^\top\xi_X - \xi_X^\top Y)$ and $\oplus$ is the Kronecker sum, i.e., $A \oplus B = A \otimes I + I \otimes B$ with Kronecker product $\otimes$. It claims in [52] that the total number of iterations is affected by the choice of retractions. Therefore, algorithms with the polar decomposition may require more iterations than those with Cayley transform to solve the optimization problems [60].

- QR decomposition

$$R_X^{\mathrm{qr}}(-\tau D) = \mathrm{qr}(X - \tau D).$$

It can be seen as an approximation of the polar decomposition. The main cost is the QR decomposition of an n-by-p matrix. The associated vector transport is defined as [42, Example 8.1.5]

$$\mathcal{T}_{\eta_X}^{\mathrm{qr}}\xi_X = Y\rho_{\mathrm{skew}}(Y^\top\xi_X(Y^\top(X + \eta_X))^{-1}) + (I - YY^\top)\xi_X(Y^\top(X + \eta_X))^{-1},$$

where $Y = R_X(\eta_X)$ and $\rho_{\text{skew}}(A)$ is defined as

$$\rho_{\text{skew}}(A) = \begin{cases} A_{ij}, & \text{if } i > j, \\ 0, & \text{if } i = j, \\ -A_{ji}, & \text{if } i < j. \end{cases}$$

Recently, these retractions are also used to design the neural network structure and solve deep learning tasks [62,63].

The vector transport above requires an associated retraction. Removing the dependence of the retraction, a new class of vector transports is introduced in [64]. Specifically, a jointly smooth operator $\mathcal{L}(x, y) : T_x\mathcal{M} \to T_y\mathcal{M}$ is defined. In addition, $\mathcal{L}(x, x)$ is required to be an identity for all x. For a d-dimensional submanifold $\mathcal{M}$ of n-dimensional Euclidean space, two popular vector transports are defined by the projection [42, Section 8.1.3]

$$\mathcal{L}^{\text{pj}}(x, y)\xi_x = \mathbf{P}_{T_y\mathcal{M}}(\xi_x),$$

and by parallelization [64]

$$\mathcal{L}^{\text{pl}}(x, y)\xi_x = B_y B_x^{\dagger}\xi_x,$$

where $B : \mathcal{V} \to \mathbb{R}^{n\times d} : z \to B_z$ is a smooth tangent basis field defined on an open neighborhood $\mathcal{V}$ of $\mathcal{M}$ and $B_z^{\dagger}$ is the pseudo-inverse of B_z. With the tangent basis B_z, we can also represent the vector transport mentioned above intrinsically, which sometimes reduces computational cost significantly [65].

To better understand Riemannian first-order algorithms, we present a Riemannian gradient method [51] in Algorithm 1. One can easily see that the difference to the Euclidean case is an extra retraction step.

Algorithm 1: Riemannian gradient method

Step 1 Input $x_0 \in \mathcal{M}$. Set $k = 0$, $\gamma_{\min} \in [0, 1]$, $\gamma_{\max} \geqslant 1$, $C_0 = f(x_0)$, $Q_0 = 1$.
Step 2 **while** $\|\text{grad}\, f(x_k)\| \neq 0$ **do**
Step 3 Compute $\eta_k = -\text{grad}\, f(x_k)$.
Step 4 Calculate γ_k according to (3.10) and set $\gamma_k = \max(\gamma_{\min}, \min(\gamma_k, \gamma_{\max}))$. Then, compute C_k, Q_k and find a step size t_k satisfying (3.9).
Step 5 Set $x_{k+1} \leftarrow R_{x_k}(t_k\eta_k)$.
Step 6 Set $k \leftarrow k + 1$.

The convergence of Algorithm 1 [66, Theorem 1] is given as follows. Although the submanifold is considered in [66], the following theorem also holds for the quotient manifold.

Theorem 3.3 *Let $\{x_k\}$ be a sequence generated by Algorithm 1 using the non-monotone line search (3.9). Suppose that f is continuously differentiable on the manifold $\mathcal{M}$.*

Then, every accumulation point x_* of the sequence $\{x_k\}$ is a stationary point of problem (1.1), i.e., it holds $\operatorname{grad} f(x_*) = 0$.

Proof At first, by using $\langle \operatorname{grad} f(x_k), \eta_k \rangle_{x_k} = -\|\operatorname{grad} f(x_k)\|_{x_k}^2 < 0$ and applying [67, Lemma 1.1], we have $f(x_k) \leqslant C_k$ and $x_k \in \{x \in \mathcal{M} : f(x) \leqslant f(x_0)\}$ for all $k \in \mathbb{N}$. Next, due to

$$\begin{aligned}&\lim_{t \downarrow 0} \frac{(f \circ R_{x_k})(t\eta_k) - f(x_k)}{t} - \rho \langle \operatorname{grad} f(x_k), \eta_k \rangle_{x_k} \\ &= \nabla f(R_{x_k}(0))^\top DR_{x_k}(0)\eta_k + \rho \|\operatorname{grad} f(x_k)\|_{x_k}^2 = -(1-\rho)\|\operatorname{grad} f(x_k)\|_{x_k}^2 < 0,\end{aligned}$$

there always exists a positive step size $t_k \in (0, \gamma_k]$ satisfying the monotone and nonmonotone Armijo conditions (3.8) and (3.9), respectively. Now, let $x_* \in \mathcal{M}$ be an arbitrary accumulation point of $\{x_k\}$ and let $\{x_k\}_K$ be a corresponding subsequence that converges to x_*. By the definition of C_{k+1} and (3.8), we have

$$C_{k+1} = \frac{\varrho Q_k C_k + f(x_{k+1})}{Q_{k+1}} < \frac{(\varrho Q_k + 1)C_k}{Q_{k+1}} = C_k.$$

Hence, $\{C_k\}$ is monotonically decreasing and converges to some limit $\bar{C} \in \mathbb{R} \cup \{-\infty\}$. Using $f(x_k) \to f(x_*)$ for $K \ni k \to \infty$, we can infer $\bar{C} \in \mathbb{R}$ and thus, we obtain

$$\infty > C_0 - \bar{C} = \sum_{k=0}^{\infty} C_k - C_{k+1} \geqslant \sum_{k=0}^{\infty} \frac{\rho t_k \|\operatorname{grad} f(x_k)\|_{x_k}^2}{Q_{k+1}}.$$

Due to $Q_{k+1} = 1 + \varrho Q_k = 1 + \varrho + \varrho^2 Q_{k-1} = \cdots = \sum_{i=0}^{k} \varrho^i < (1-\varrho)^{-1}$, this implies $\{t_k \|\operatorname{grad} f(x_k)\|_{x_k}^2\} \to 0$. Let us now assume $\|\operatorname{grad} f(x_*)\| \neq 0$. In this case, we have $\{t_k\}_K \to 0$ and consequently, by the construction of Algorithm 1, the step size $\delta^{-1} t_k$ does not satisfy (3.9), i.e., it holds

$$-\rho(\delta^{-1}t_k)\|\operatorname{grad} f(x_k)\|_{x_k}^2 < f(R_{x_k}(\delta^{-1}t_k\eta_k)) - C_k \leqslant f(R_{x_k}(\delta^{-1}t_k\eta_k)) - f(x_k) \tag{3.12}$$

for all $k \in K$ sufficiently large. Since the sequence $\{\eta_k\}_K$ is bounded, the rest of the proof is now identical to the proof of [42, Theorem 4.3.1]. In particular, applying the mean value theorem in (3.12) and using the continuity of the Riemannian metric, we can easily derive a contradiction. We refer to [42] for more details.

3.4 Second-Order-Type Algorithms

A gradient-type algorithm usually is fast in the early iterations, but it often slows down or even stagnates when the generated iterations are close to an optimal solution. When a high accuracy is required, second-order-type algorithms may have its advantage.

By utilizing the exact Riemannian Hessian and different retraction operators, Riemannian Newton methods, trust-region methods, adaptive regularized Newton method

have been proposed in [42,51,68,69]. When the second-order information is not available, the quasi-Newton-type method becomes necessary. As in the Riemannian CG method, we need the vector transport operator to compare different tangent vectors from different tangent spaces. In addition, extra restrictions on the vector transport and the retraction are required for better convergence property or even convergence [61, 64,70–74]. Non-vector-transport-based quasi-Newton method is also explored in [75].

3.4.1 Riemannian Trust-Region Method

One of the popular second-order algorithms is a Riemannian trust-region (RTR) algorithm [42,69]. At the kth iteration x_k, by utilizing the Taylor expansion on manifold, RTR constructs the following subproblem on the Tangent space:

$$\min_{\xi \in T_{x_k}\mathcal{M}} \quad m_k(\xi) := \langle \operatorname{grad} f(x_k), \xi \rangle_{x_k} + \frac{1}{2} \langle \operatorname{Hess} f(x_k)[\xi], \xi \rangle_{x_k} \quad \text{s.t.} \quad \|\xi\|_{x_k} \leqslant \Delta_k, \tag{3.13}$$

where Δ_k is the trust-region radius. In [76], extensive methods for solving (3.13) are summarized. Among them, the Steihaug CG method, also named as truncated CG method, is most popular due to its good properties and relatively cheap computational cost. By solving this trust-region subproblem, we obtain a direction $\xi_k \in T_{x_k}\mathcal{M}$ satisfying the so-called Cauchy decrease. Then, a trial point is computed as $z_k = R_{x_k}(\xi_k)$, where the step size is chosen as 1. To determine the acceptance of z_k, we compute the ratio between the actual reduction and the predicted reduction

$$\rho_k := \frac{f(x_k) - f(R_{x_k}(\xi_k))}{m_k(0) - m_k(\xi_k)}. \tag{3.14}$$

When ρ_k is greater than some given parameter $0 < \eta_1 < 1$, z_k is accepted. Otherwise, z_k is rejected. To avoid the algorithm stagnating at some feasible point and promote the efficiency as well, the trust-region radius is also updated based on ρ_k. The full algorithm is presented in Algorithm 2.

Algorithm 2: Riemannian trust-region method

Step 1 **Input:** Initial guess $x_0 \in \mathcal{M}$ and parameters $\bar{\Delta} > 0$, $\Delta_0 \in (0, \bar{\Delta})$, $\rho' \in [0, \frac{1}{4})$.
Step 2 **Output:** Sequences of iterates $\{x_k\}$ and related information.
Step 3 **for** $k = 0, 1, 2, \cdots$ **do**
Step 4 Use the truncated CG method to obtain ξ_k by solving (3.13).
Step 5 Compute the ratio ρ_k in (3.14).
Step 6 **if** $\rho_k < \frac{1}{4}$ **then** $\Delta_{k+1} = \frac{1}{4}\Delta_k$ **else if** $\rho_k > \frac{3}{4}$ *and* $\|\xi_k\| = \Delta_k$ **then** $\Delta_{k+1} = \min(2\Delta_k, \bar{\Delta})$ **else** $\Delta_{k+1} = \Delta_k$ **if** $\rho_k > \rho'$ **then** $x_{k+1} = R_{x_k}(\xi_k)$ **else** $x_{k+1} = x_k$.

For the global convergence, the following assumptions are necessary for second-order-type algorithms on manifold.

Assumption 3.4 (a). The function f is continuous differentiable and bounded from below on the level set $\{x \in \mathcal{M} : f(x) \leqslant f(x_0)\}$.
(b). There exists a constant $\beta_{\text{Hess}} > 0$ such that

$$\|\text{Hess}\, f(x_k)\| \leqslant \beta_{\text{Hess}}, \ \forall k = 0, 1, 2, \cdots.$$

Algorithm 2 also requires a Lipschitz-type continuous property on the objective function f [42, Definition 7.4.1].

Assumption 3.5 There exists two constants $\beta_{\text{RL}} > 0$ and $\delta_{\text{RL}} > 0$ such that for all $x \in \mathcal{M}$ and $\xi \in T_x\mathcal{M}$ with $\|\xi\| = 1$,

$$\left| \frac{\mathrm{d}}{\mathrm{d}t} f \circ R_x(t\xi) \mid_{t=\tau} - \frac{\mathrm{d}}{\mathrm{d}t} f \circ R_x(t\xi) \mid_{t=0} \right| \leqslant \tau\beta_{\text{RL}}, \quad \forall \tau \leqslant \delta_{\text{RL}}.$$

Then, the global convergence to a stationary point [42, Theorem 7.4.2] is presented as follows:

Theorem 3.6 *Let $\{x_k\}$ be a sequence generated by Algorithm 2. Suppose that Assumptions 3.4 and 3.5 hold, then*

$$\liminf_{k\to\infty} \|\text{grad}\, f(x_k)\| = 0.$$

By further assuming the Lipschitz continuous property of the Riemannian gradient [42, Definition 7.4.3] and some isometric property of the retraction operator R [42, Equation (7.25)], the convergence of the whole sequence is proved [42, Theorem 7.4.4]. The locally superlinear convergence rate of RTR and its related assumptions can be found in [42, Section 7.4.2].

3.4.2 Adaptive Regularized Newton Method

From the perspective of Euclidean approximation, an adaptive regularized Newton algorithm (ARNT) is proposed for specific and general Riemannian submanifold optimization problems [21,51,77]. In the subproblem, the objective function is constructed by the second-order Taylor expansion in the Euclidean space and an extra regularization term, while the manifold constraint is kept. Specifically, the mathematical formulation is

$$\min_{x\in\mathcal{M}} \quad \hat{m}_k(x) := \langle \nabla f(x), x - x_k\rangle + \frac{1}{2}\langle H_k[x - x_k], x - x_k\rangle + \frac{\sigma_k}{2}\|x - x_k\|^2, \tag{3.15}$$

where H_k is the Euclidean Hessian or its approximation. From the definition of Riemannian gradient and Hessian, we have

$$\begin{aligned} \text{grad}\, \hat{m}_k(x_k) &= \text{grad}\, f(x_k), \\ \text{Hess}\, \hat{m}_k(x_k)[U] &= \mathbf{P}_{T_{x_k}\mathcal{M}}(H_k[U]) + \mathfrak{W}_{x_k}(U, \mathbf{P}^{\perp}_{T_{x_k}\mathcal{M}}(\nabla f(x_k))) + \sigma_k U, \end{aligned} \tag{3.16}$$

where $U \in T_{x_k}\mathcal{M}$, $\mathbf{P}^{\perp}_{T_{x_k}\mathcal{M}} := I - \mathbf{P}_{T_{x_k}\mathcal{M}}$ is the projection onto the normal space and the Weingarten map $\mathfrak{W}_x(\cdot, v)$ with $v \in T^{\perp}_{x_k}\mathcal{M}$ is a symmetric linear operator which is related to the second fundamental form of $\mathcal{M}$. To solve (3.15), a modified CG method is proposed in [51] to solve the Riemannian Newton equation at x_k,

$$\operatorname{grad} \hat{m}_k(x_k) + \operatorname{Hess} \hat{m}_k(x_k)[\xi_k] = 0.$$

Since $\operatorname{Hess} \hat{m}_k(x_k)$ may not be positive definite, CG may be terminated if a direction with negative curvature, says d_k, is encountered. Different from the truncated CG method used in RTR, a linear combination of s_k (the output of the truncated CG method) and the negative curvature direction d_k is used to construct a descent direction

$$\xi_k = \begin{cases} s_k + \tau_k d_k, & \text{if } d_k \neq 0, \\ s_k, & \text{if } d_k = 0, \end{cases} \quad \text{with} \quad \tau_k := \frac{\langle d_k, \operatorname{grad} \hat{m}_k(x_k) \rangle_{x_k}}{\langle d_k, \operatorname{Hess} \hat{m}_k(x_k)[d_k] \rangle_{x_k}}. \tag{3.17}$$

A detailed description on the modified CG method is presented in Algorithm 3. Then, Armijo search along ξ_k is adopted to obtain a trial point z_k. After obtaining z_k, we compute the following ratio between the actual reduction and the predicted reduction,

$$\hat{\rho}_k = \frac{f(z_k) - f(x_k)}{\hat{m}_k(z_k)}. \tag{3.18}$$

Algorithm 3: A modified CG method for solving subproblem (3.15)

Step 1 Set $T > 0, \theta > 1, \varepsilon \geqslant 0, \eta_0 = 0, r_0 = \operatorname{grad} m_k(x_k)$, $p_0 = -r_0$, and $i = 0$.
Step 2 **while** $i \leqslant n - 1$ **do**
Step 3 Compute $\pi_i = \langle p_i, \operatorname{Hess} \hat{m}_k(x_k)[p_i] \rangle_{x_k}$.
Step 4 **if** $\pi_i / \langle p_i, p_i \rangle_{x_k} \leqslant \varepsilon$ **then**
Step 5 **if** $i = 0$ **then** set $s_k = -p_0$, $d_k = 0$ **else** set $s_k = \eta_i$,
Step 6 **if** $\pi_i / \langle p_i, p_i \rangle_{x_k} \leqslant -\varepsilon$ **then** $d_k = p_i$, set $\sigma_{est} = |\pi_i| / \langle p_i, p_i \rangle_{x_k}$ **else** $d_k = 0$ break
Step 7 Set $\alpha_i = \langle r_i, r_i \rangle_{x_k} / \pi_i$, $\eta_{i+1} = \eta_i + \alpha_i p_i$, and $r_{i+1} = r_i + \alpha_i \operatorname{Hess} \hat{m}_k(x_k)[p_i]$.
Step 8 **if** $\|r_{i+1}\|_{x_k} \leqslant \min\{\|r_0\|^{\theta}_{x_k}, T\}$ **then**
Step 9 choose $s_k = \eta_{i+1}$, $d_k = 0$; break;
Step 10 Set $\beta_{i+1} = \langle r_{i+1}, r_{i+1} \rangle_{x_k} / \langle r_i, r_i \rangle_{x_k}$ and $p_{i+1} = -r_{i+1} + \beta_{i+1} p_i$.
Step 11 $i \leftarrow i + 1$.
Step 12 Update ξ_k according to (3.17).

If $\hat{\rho}_k \geqslant \eta_1 > 0$, then the iteration is successful and we set $x_{k+1} = z_k$; otherwise, the iteration is not successful and we set $x_{k+1} = x_k$, i.e.,

$$x_{k+1} = \begin{cases} z_k, & \text{if } \hat{\rho}_k \geqslant \eta_1, \\ x_k, & \text{otherwise.} \end{cases} \tag{3.19}$$

The regularization parameter σ_{k+1} is updated as follows:

$$\sigma_{k+1} \in \begin{cases} (0, \gamma_0 \sigma_k], & \text{if } \hat{\rho}_k \geqslant \eta_2, \\ [\gamma_0 \sigma_k, \gamma_1 \sigma_k], & \text{if } \eta_1 \leqslant \hat{\rho}_k < \eta_2, \\ [\gamma_1 \sigma_k, \gamma_2 \sigma_k], & \text{otherwise}, \end{cases} \tag{3.20}$$

where $0 < \eta_1 \leqslant \eta_2 < 1$ and $0 < \gamma_0 < 1 < \gamma_1 \leqslant \gamma_2$. These parameters determine how aggressively the regularization parameter is adjusted when an iteration is successful or unsuccessful. Putting these features together, we obtain Algorithm 4, which is dubbed as ARNT.

Algorithm 4: An Adaptive Regularized Newton Method

Step 1 Choose a feasible initial point $x_0 \in \mathcal{M}$ and an initial regularization parameter $\sigma_0 > 0$. Choose $0 < \eta_1 \leqslant \eta_2 < 1, 0 < \gamma_0 < 1 < \gamma_1 \leqslant \gamma_2$. Set $k := 0$.
Step 2 **while** *stopping conditions not met* **do**
Step 3 Compute a new trial point z_k by doing Armijo search along ξ_k obatined by Algorithm 3.
Step 4 Compute the ratio $\hat{\rho}_k$ via (3.18).
Step 5 Update x_{k+1} from the trial point z_k based on (3.19).
Step 6 Update σ_k according to (3.20).
Step 7 $k \leftarrow k + 1$.

We next present the convergence property of Algorithm 4 with the inexact Euclidean Hessian starting from a few assumptions.

Assumption 3.7 Let $\{x_k\}$ be generated by Algorithm 4 with the inexact Euclidean Hessian H_k.

(A.1) The gradient ∇f is Lipschitz continuous on the convex hull of the manifold $\mathcal{M}$ – denoted by $\mathrm{conv}(\mathcal{M})$, i.e., there exists $L_f > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leqslant L_f \|x - y\|, \quad \forall\, x, y \in \mathrm{conv}(\mathcal{M}).$$

(A.2) There exists $\kappa_g > 0$ such that $\|\nabla f(x_k)\| \leqslant \kappa_g$ for all $k \in \mathbb{N}$.
(A.3) There exists $\kappa_H > 0$ such that $\|H_k\| \leqslant \kappa_H$ for all $k \in \mathbb{N}$.
(A.4) Suppose there exists $\underline{\varpi} > 0$, $\overline{\varpi} \geqslant 1$ such that $\underline{\varpi}$ and $\overline{\varpi}$

$$\underline{\varpi} \|\xi\|^2 \leqslant \|\xi\|_{x_k}^2 \leqslant \overline{\varpi} \|\xi\|^2, \ \xi \in T_{x_k}\mathcal{M},$$

for all $k \in \mathbb{N}$.

We note that the assumptions (A.2) and (A.4) hold if f is continuous differentiable and the level set $\{x \in \mathcal{M} : f(x) \leqslant f(x_0)\}$ is compact.

The global convergence to an stationary point can be obtained.

Theorem 3.8 *Suppose that Assumptions 3.4 and 3.7 hold. Then, either*

$$\mathrm{grad}\, f(x_\ell) = 0 \ \textit{for some} \ \ell \geqslant 0 \quad \textit{or} \quad \liminf_{k \to \infty} \|\mathrm{grad}\, f(x_k)\|_{x_k} = 0.$$

For the local convergence rate, we make the following assumptions.

Assumption 3.9 Let $\{x_k\}$ be generated by Algorithm 4.

(B.1) There exists $\beta_R, \delta_R > 0$ such that

$$\left\| \frac{D}{\mathrm{d}t} \frac{\mathrm{d}}{\mathrm{d}t} R_x(t\xi) \right\|_x \leqslant \beta_R,$$

for all $x \in \mathcal{M}$, all $\xi \in T_x\mathcal{M}$ with $\|\xi\|_x = 1$ and all $t < \delta_R$.

(B.2) The sequence $\{x_k\}$ converges to x_*.

(B.3) The Euclidean Hessian $\nabla^2 f$ is continuous on $\mathrm{conv}(\mathcal{M})$.

(B.4) The Riemannian Hessian $\mathrm{Hess}\, f$ is positive definite at x_* and the constant ε in Algorithm 3 is set to zero.

(B.5) H_k is a good approximation of the Euclidean Hessian $\nabla^2 f$, i.e., it holds

$$\|H_k - \nabla^2 f(x_k)\| \to 0, \quad \text{whenever} \quad \|\mathrm{grad}\, f(x_k)\|_{x_k} \to 0.$$

Then, we have the following results on the local convergence rate.

Theorem 3.10 *Suppose that the conditions (B.1)–(B.5) in Assumption 3.9 are satisfied. Then, the sequence $\{x_k\}$ converges q-superlinearly to x_*.*

The detailed convergence analysis can be found in [51].

3.4.3 Quasi-Newton-Type Methods

When the Riemannian Hessian $\mathrm{Hess}\, f(x)$ is computationally expensive or even not available, quasi-Newton-type methods turn out to be an attractive approach. In the literature [61,64,70–74], extensive variants of quasi-Newton methods are proposed. Here, we take the Riemannian Broyden–Fletcher–Goldfarb–Shanno (BFGS) as an example to show the general idea of quasi-Newton methods on Riemannian manifold. Similar to the quasi-Newton method in the Euclidean space, an approximation $\mathcal{B}_{k+1}$ should satisfy the following secant equation

$$\mathcal{B}_{k+1} s_k = y_k,$$

where $s_k = \mathcal{T}_{S_{\alpha_k \xi_k}} \alpha_k \xi_k$ and $y_k = \beta_k^{-1} \mathrm{grad}\, f(x_{k+1}) - \mathcal{T}_{S_{\alpha_k \xi_k}} \mathrm{grad}\, f(x_k)$ with parameter β_k. Here, α_k and ξ_k is the step size and the direction used in the kth iteration. $\mathcal{T}_S$ is an isometric vector transport operator by the differentiated retraction R, i.e.,

$$\left\langle \mathcal{T}_{S_{\xi_x}} u_x, \mathcal{T}_{S_{\xi_x}} v_x \right\rangle_{R_x(\xi_x)} = \langle u_x, v_x \rangle_x .$$

Additionally, $\mathcal{T}_S$ should satisfy the following locking condition,

$$\mathcal{T}_{S_{\xi_k}} \xi_k = \beta_k \mathcal{T}_{R_{\xi_k}} \xi_k, \ \beta_k = \frac{\|\xi_k\|_{x_k}}{\|\mathcal{T}_{R_{\xi_k}} \xi_k\|_{R_{x_k}(\xi_k)}},$$

where $\mathcal{T}_{R_{\xi_k}}\xi_k = \frac{\mathrm{d}}{\mathrm{d}t}R_{x_k}(t\xi_k)\mid_{t=1}$. Then, the scheme of the Riemannian BFGS is

$$\mathcal{B}_{k+1} = \hat{\mathcal{B}}_k - \frac{\hat{\mathcal{B}}_k s_k (\hat{\mathcal{B}}_k s_k)^\flat}{(\hat{\mathcal{B}}_k s_k)^\flat s_k} + \frac{y_k y_k^\flat}{y_k^\flat s_k}, \tag{3.21}$$

where $a^\flat : T_x\mathcal{M} \to \mathbb{R} : v \to \langle a, v\rangle_x$ and $\hat{\mathcal{B}}_k = \mathcal{T}_{S_{\alpha_k\xi_k}}\alpha_k\xi_k \circ \mathcal{B}_k \circ \left(\mathcal{T}_{S_{\alpha_k\xi_k}}\alpha_k\xi_k\right)^{-1}$ is from $T_{x_{k+1}}\mathcal{M}$ to $T_{x_{k+1}}\mathcal{M}$. With this choice of β_k and the isometric property of $\mathcal{T}_S$, we can guarantee the positive definiteness of $\mathcal{B}_{k+1}$. After obtaining the new approximation $\mathcal{B}_{k+1}$, the Riemannian BFGS method solves the following linear system

$$\mathcal{B}_{k+1}\xi_{k+1} = -\mathrm{grad}\, f(x_{k+1})$$

to get ξ_{k+1}. The detailed algorithm is presented in Algorithm 5. The choice of $\beta_k = 1$ can also guarantee the convergence but with more strict assumptions. One can refer to [64] for the convergence analysis. Since the computation of differentiated retraction may be costly, authors in [74] investigate another way to preserve the positive definiteness of the BFGS scheme. Meanwhile, the Wolfe search is replaced by the Armijo search. As a result, the differentiated retraction can be avoided and the convergence analysis is presented as well.

Algorithm 5: Riemannian BFGS method

Step 1 **Input:** Initial guess $x_0 \in \mathcal{M}$, isometric vector transport $\mathcal{T}_S$ associated with the retraction R, initial Riemannian Hessian approximation $\mathcal{B}_0 : T_{x_0}\mathcal{M} \to T_{x_0}\mathcal{M}$, which is symmetric positive definite, Wolfe condition parameters $0 < c_1 < \frac{1}{2} < c_2 < 1$.

Step 2 **for** $k = 0, 1, 2, \cdots$ **do**

Step 3 Solve $\mathcal{B}_k\xi_k = -\mathrm{grad}\, f(x_k)$ to get ξ_k.

Step 4 Obtain x_{k+1} by doing a Wolfe search along ξ_k, i.e., finding $\alpha_k > 0$ such that the following two conidtions are satisfied

$$f(R_{x_k}(\alpha_k\xi_k)) \leqslant f(x_k) + c_1\alpha_k \langle \mathrm{grad}\, f(x_k), \xi_k\rangle_{x_k},$$
$$\frac{\mathrm{d}}{\mathrm{d}t} f(R_{x_k}(t\xi_k))\mid_{t=\alpha_k} \geqslant c_2 \frac{\mathrm{d}}{\mathrm{d}t} f(R_{x_k}(t\xi_k))\mid_{t=0}.$$

Step 5 Set $x_k = R_{x_k}(\alpha_k\xi_k)$.

Step 6 Update $\mathcal{B}_{k+1}$ by (3.21).

The aforementioned quasi-Newton methods rely on the vector transport operator. When the vector transport operation is computationally costly, these methods may be less competitive. Noticing the structure of the Riemannian Hessian $\mathrm{Hess}\, f(x_k)$, i.e.,

$$\mathrm{Hess}\, f(x_k)[U] = \mathbf{P}_{T_{x_k}\mathcal{M}}(\nabla^2 f(x_k)[U]) + \mathfrak{W}_{x_k}(U, \mathbf{P}^{\perp}_{T_{x_k}\mathcal{M}}(\nabla f(x_k))),\ U \in T_{x_k}\mathcal{M},$$

where the second term $\mathfrak{W}_{x_k}(U, \mathbf{P}^{\perp}_{T_{x_k}\mathcal{M}}(\nabla f(x_k)))$ is often much cheaper than the first term $\mathbf{P}_{T_{x_k}\mathcal{M}}(\nabla^2 f(x_k)[U])$. Similar to the quasi-Newton methods in unconstrained

nonlinear least square problems [78] [79, Chapter 7], we can focus on the construction of an approximation of the Euclidean Hessian $\nabla^2 f(x_k)$ and use exact formulations of remaining parts. Furthermore, if the Euclidean Hessian itself consists of cheap and expensive parts, i.e.,

$$\nabla^2 f(x_k) = \mathcal{H}^{\mathrm{c}}(x_k) + \mathcal{H}^{\mathrm{e}}(x_k), \tag{3.22}$$

where the computational cost of $\mathcal{H}^{\mathrm{e}}(x_k)$ is much more expensive than $\mathcal{H}^{\mathrm{c}}(x_k)$, an approximation of $\nabla^2 f(x_k)$ is constructed as

$$H_k = \mathcal{H}^{\mathrm{c}}(x_k) + C_k, \tag{3.23}$$

where C_k is an approximation of $\mathcal{H}^{\mathrm{e}}(x_k)$ obtained by a quasi-Newton method in the ambient Euclidean space. If an objective function f is not equipped with the structure (3.22), H_k is a quasi-Newton approximation of $\nabla^2 f(x_k)$. In the construction of the quasi-Newton approximation, a Nyström approximation technique [75, Section 2.3] is explored, which turns to be a better choice than the BB-type initialization [76, Chapter 6]. Since the quasi-Newton approximation is constructed in the ambient Euclidean space, the vector transport is not necessary. Then, subproblem (3.15) is constructed with H_k. From the expression of the Riemannian Hessian $\mathrm{Hess}\,\hat{m}_k$ in (3.16), we see that subproblem (3.15) gives us a way to approximate the Riemannian Hessian when an approximation H_k to the Euclidean Hessian is available. The same procedures of ARNT can be utilized for (3.15) with the approximate Euclidean Hessian H_k. An adaptive structured quasi-Newton method given in [75] is presented in Algorithm 6.

Algorithm 6: A structured quasi-Newton method

Step 1 Input an initial guess $X^0 \in \mathcal{M}$. Choose $\tau_0 > 0$, $0 < \eta_1 \leqslant \eta_2 < 1$, $1 < \gamma_1 \leqslant \gamma_2$. Set $k = 0$.
Step 2 **while** *stopping conditions not met* **do**
Step 3 Check the structure of $\nabla^2 f(x_k)$ to see if it can be written in a form as (3.22).
Step 4 Construct an approximation H_k by utilizing a quasi-Newton method.
Step 5 Construct and solve the subproblem (3.15) (by using the modified CG method or the Riemannian gradient-type method) to obtain a new trial point z_k.
Step 6 Compute the ratio ρ_k via (3.14).
Step 7 Update x_{k+1} from the trial point z_k based on (3.19).
Step 8 Update τ_k according to (3.20).
Step 9 $k \leftarrow k + 1$.

To explain the differences between the two quasi-Newton algorithms more straightforwardly, we take the HF total energy minimization problem (2.10) as an example. From the calculation in [75], we have the Euclidean gradients

$$\nabla E_{\mathrm{ks}}(X) = H_{\mathrm{ks}}(X)X, \quad \nabla E_{\mathrm{hf}}(X) = H_{\mathrm{hf}}(X)X,$$

where $H_{\mathrm{ks}}(X) := \frac{1}{2}L + V_{\mathrm{ion}} + \sum_l \zeta_l w_l w_l^* + \mathrm{Diag}((\Re L^\dagger)\rho) + \mathrm{Diag}(\mu_{\mathrm{xc}}(\rho)^* e)$ and $H_{\mathrm{hf}}(X) = H_{\mathrm{ks}}(X) + \mathcal{V}(XX^*)$. The Euclidean Hessian of E_{ks} and E_{f} along a matrix

$U \in \mathbb{C}^{n\times p}$ are

$$\nabla^2 E_{\rm ks}(X)[U] = H_{\rm ks}(X)U + {\rm Diag}\left(\left(\Re L^\dagger + \frac{\partial^2 \varepsilon_{\rm xc}}{\partial \rho^2} e\right)(\bar{X}\odot U + X \odot \bar{U})e\right) X,$$
$$\nabla^2 E_{\rm f}(X)[U] = \mathcal{V}(XX^*)U + \mathcal{V}(XU^* + UX^*)X.$$

Since $\nabla^2 E_{\rm f}(X)$ is significantly more expensive than $\nabla^2 E_{\rm ks}(X)$, we only need to approximate $\nabla^2 E_{\rm f}(X)$. The differences $X_k - X_{k-1}$, $\nabla E_{\rm f}(X_k) - \nabla E_{\rm f}(X_{k-1})$ are computed. Then, a quasi-Newton approximation C_k of $\nabla^2 E_{\rm f}$ is obtained without requiring vector transport. By adding the exact formulation of $\nabla^2 E_{\rm ks}(X_k)$, we have an approximation H_k, i.e.,

$$H_k = \nabla^2 E_{\rm ks} + C_k.$$

A Nyström approximation for C_k is also investigated. Note that the spectrum of $\nabla^2 E_{\rm ks}(X)$ dominates the spectrum of $\nabla^2 E_{\rm f}(X)$. The structured approximation H_k is more reliable than a direct quasi-Newton approximate to $\nabla^2 E_{\rm hf}(X)$ because the spectrum of $\nabla^2 E_{\rm ks}$ is inherited from the exact form. The remaining procedure is to solve subproblem (3.15) to update X_k.

3.5 Stochastic Algorithms

For problems arising from machine learning, the objective function f is often a summation of a finite number of functions $f_i, i = 1, \cdots, m$, namely,

$$f(x) = \sum_{i=1}^{m} f_i(x).$$

For unconstrained situations, there are many efficient algorithms, such as Adam, Adagrad, RMSProp, Adelta and SVRG. One can refer to [80]. For the case with manifold constraints, combining with retraction operators and vector transport operator, these algorithms can be well generalized. However, in the implementation, due to the considerations of the computational costs of different parts, they may have different versions. Riemannian stochastic gradient method is first developed in [81]. Later, a class of first-order methods and their accelerations are investigated for geodesically convex optimization in [82,83]. With the help of parallel translation or vector transport, Riemannian SVRG methods are generalized in [84,85]. In consideration of the computational cost of the vector transport, non-vector transport-based Riemannian SVRG is proposed in [86]. Since an intrinsic coordinate system is absent, the coordinate-wise update on manifold should be further investigated. A compromised approach for Riemannian adaptive optimization methods on product manifolds is presented in [87].

Here, the SVRG algorithm [86] is taken as an example. At the current point $X^{s,k}$, we first calculate the full gradient $\mathcal{G}(X^{s,k})$, then randomly sample a subscript from 1 to m and use this to construct a stochastic gradient with reduced variance as $\mathcal{G}(X^{s,k}, \xi_{s,k}) =$

$\nabla f(X^{s,0}) + \left(\nabla f_{i_{s,k}}(X^{s,k}) - \nabla f_{i_{s,k}}(X^{s,0})\right)$, finally move along this direction with a given step size to next iteration point

$$X^{s,k+1} = X^{s,k} - \tau_s \xi_{s,k}.$$

For Riemannian SVRG [86], after obtaining the stochastic gradient with reduced Euclidean variance, it first projects this gradient to the tangent space

$$\xi_{s,k} = \mathbf{P}_{T_{X^{s,k}}\mathcal{M}}(\mathcal{G}(X^{s,k}))$$

for a submanifold $\mathcal{M}$. We note that the tangent space should be replaced by the horizontal space when $\mathcal{M}$ is a quotient manifold. Then, the following retraction step

$$X^{s,k+1} = R_{X^{s,k}}(-\tau_s \xi_{s,k})$$

is executed to get the next feasible point. The detailed version is outlined in Algorithm 7.

Algorithm 7: Riemannian SVRG [86]

Step 1 **for** $s = 0, \cdots, S-1$ **do**

Step 2 Calculate the full gradient $\nabla f(X^{s,0})$ and sets the step size $\tau_s > 0$.

Step 3 **for** $k = 0, \cdots, K-1$ **do**

Step 4 Randomly substitute samples to get the subscript $i_{s,k} \subseteq \{1, \cdots, m\}$. Calculate a random Euclidean gradient $\mathcal{G}(X^{s,k})$

$$\mathcal{G}(X^{s,k}, \xi_{s,k}) = \nabla f(X^{s,0}) + \left(\nabla f_{i_{s,k}}(X^{s,k}) - \nabla f_{i_{s,k}}(X^{s,0})\right).$$

Calculate a random Riemann gradient

$$\xi_{s,k} = \mathbf{P}_{T_{X^{s,k}}\mathcal{M}}(\mathcal{G}(X^{s,k})).$$

Update $X^{s,k+1}$ in the following format

$$X^{s,k+1} = R_{X^{s,k}}(-\tau_s \xi_{s,k}).$$

Step 5 Take $X^{s+1,0} \leftarrow X^{s,K}$.

3.6 Algorithms for Riemannian Non-smooth Optimization

As shown in Sects. 2.11 to 2.15, many practical problems are with non-smooth objective function and manifold constraints, i.e.,

$$\min_{x \in \mathcal{M}} \quad f(x) := g(x) + h(x),$$

where g is smooth and h is non-smooth. Riemannian subgradient methods [88,89] are firstly investigated to solve this kind of problems, and their convergence analysis is exhibited in [90] with the help of Kurdyka–Łojasiewicz (KŁ) inequalities. For locally Lipschitz functions on Riemannian manifolds, a gradient sampling method and a non-smooth Riemannian trust-region method are proposed in [91,92]. Proximal point methods on manifold are presented in [93,94], where the inner subproblem is solved inexactly by subgradient-type methods. The corresponding complexity analysis is given in [95,96]. Different from the constructions of the subproblem in [93,94], a more tractable subproblem without manifold constraints is investigated in [97] for convex $h(x)$ and the Stiefel manifold. By utilizing the semi-smooth Newton method [98], the proposed proximal gradient method on manifold enjoys a faster convergence. Later, the proximal gradient method on the Stiefel manifold [97] and its accelerated version are extended to the generic manifold [99]. The accelerated proximal gradient methods are applied to solve sparse PCA and sparse canonical correlation analysis problems [100,101]. Another class of methods is based on operator-splitting techniques. Some variants of the alternating direction method of multipliers (ADMM) are studied in [102–107].

We briefly introduce the proximal gradient method on the Stiefel manifold [97] here. Assume that the convex function h is Lipschitz continuous. At each iteration x_k, the following subproblem is constructed

$$\min_d \quad \langle \operatorname{grad} g(x_k), d\rangle + \frac{1}{2t}\|d\|_F^2 + h(x_k + d) \quad \text{s.t.} \quad d \in T_{x_k}\mathcal{M}, \tag{3.24}$$

where $t > 0$ is a step size and $\mathcal{M}$ denotes the Stiefel manifold. Given a retraction R, problem (3.24) can be seen as a first-order approximation of $f(R_{x_k}(d))$ near the zero element 0_{x_k} on $T_{x_k}\mathcal{M}$. From the Lipschitz continuous property of h and the definition of R, we have

$$|h(R_{x_k}(d)) - h(x_k + d)| \leqslant L_h \|R_{x_k}(d) - (x_k + d)\|_F = O(\|d\|_F^2),$$

where L_h is the Lipschitz constant of h. Therefore, we conclude

$$f(R_{x_k}(d)) = \langle \operatorname{grad} g(x_k), d\rangle + h(x_k + d) + O(\|d\|_F^2), \quad d \to 0.$$

Then, the next step is to solve (3.24). Since (3.24) is convex and with linear constraints, the KKT conditions are sufficient and necessary for the global optimality. Specifically, we have

$$d(\lambda) = \operatorname{prox}_{th}(b(\lambda)) - x_k, \quad b(\lambda) = x_k - t(\operatorname{grad} f(x_k) - \mathcal{A}_k^*(\lambda)), \quad \mathcal{A}_k(d(\lambda)) = 0,$$

where $d \in T_{x_k}\mathcal{M}$ is represented by $\mathcal{A}_k(d) = 0$ with a linear operator $\mathcal{A}_k$, $\mathcal{A}_k^*$ is the adjoint operator of $\mathcal{A}_k$. Define $E(\lambda) := \mathcal{A}_k(d(\lambda))$, it is proved in [97] that E is monotone and then the semi-smooth Newton method in [98] is utilized to solve the nonlinear equation $E(\lambda) = 0$ to obtain a direction d_k. Combining with a curvilinear

search along d_k with R_{x_k}, the decrease on f is guaranteed and the global convergence is established.

3.7 Complexity Analysis

The complexity analysis of the Riemannian gradient method and the Riemannian trust-region method has been studied in [108]. Similar to the Euclidean unconstrained optimization, the Riemannian gradient method (using a fixed step size or Armijo curvilinear search) converges to $\|\text{grad}\, f(x)\|_x \leqslant \varepsilon$ up to $O(1/\varepsilon^2)$ steps. Under mild assumptions, a modified Riemannian trust-region method converges to $\|\text{grad}\, f(x)\|_x \leqslant \varepsilon$, $\text{Hess}\, f(x) \succeq -\sqrt{\varepsilon} I$ at most $O(\max\{1/\varepsilon^{1.5}, 1/\varepsilon^{2.5}\})$ iterations. For objective functions with multi-block convex but non-smooth terms, an ADMM of complexity of $O(1/\varepsilon^4)$ is proposed in [105]. For the cubic regularization methods on the Riemannian manifold, recent studies [109,110] show a convergence to $\|\text{grad}\, f(x)\|_x \leqslant \varepsilon$, $\text{Hess}\, f(x) \succeq -\sqrt{\varepsilon} I$ with complexity of $O(1/\varepsilon^{1.5})$.

4 Analysis for Manifold Optimization

4.1 Geodesic Convexity

For a convex function in the Euclidean space, any local minimum is also a global minimum. An interesting extension is the geodesic convexity of functions. Specifically, a function defined on manifold is said to be geodesically convex if it is convex along any geodesic. Similarly, a local minimum of a geodesically convex function on manifold is also a global minimum. Naturally, a question is how to distinguish the geodesically convex function.

Definition 4.1 Given a Riemannian manifold $(\mathcal{M}, g)$, a set $\mathcal{K} \subset \mathcal{M}$ is called g-fully geodesic, if for any $p, q \in \mathcal{K}$, any geodesic γ_{pq} is located entirely in $\mathcal{K}$.

For example, revise the set $\{P \in \mathbb{S}^n_{++} \mid \det(P) = c\}$ with a positive constant c is not convex in $\mathbb{R}^{n\times n}$, but is a fully geodesic set [111] of Riemannian manifolds $(\mathbb{S}^n_{++}, g)$, where the Riemannian metric g at P is $g_P(U, V) := \text{tr}(P^{-1}UP^{-1}V)$. Now we present the definition of the g-geodesically convex function.

Definition 4.2 Given a Riemannian manifold $(\mathcal{M}, g)$ and a g-fully geodesic set $\mathcal{K} \subset \mathcal{M}$, a function $f : \mathcal{K} \to \mathbb{R}$ is g-geodesically convex if for any $p, q \in \mathcal{K}$ and any geodesic $\gamma_{pq} : [0, 1] \to \mathcal{K}$ connecting p, q, it holds:

$$f(\gamma_{pq}(t)) \leqslant (1-t) f(p) + t f(q), \ \forall t \in [0, 1].$$

A g-fully geodesically convex function may not be convex. For example, $f(x) := (\log x)^2$, $x \in \mathbb{R}_+$ is not convex in the Euclidean space, but is convex with respect to the manifold $(\mathbb{R}_+, g)$, where $g_x(u, v) := ux^{-1}v$.

Therefore, for a specific function, it is of significant importance to define a proper Riemannian metric to recognize the geodesic convexity. A natural problem is, given

a manifold $\mathcal{M}$ and a smooth function $f : \mathcal{M} \to \mathbb{R}$, whether there is a metric g such that f is geodesic convex with respective to g? It is generally not easy to prove the existence of such a metric. From the definition of the geodesic convexity, we know that if a function has a non-global local minimum, then this function is not geodesically convex for any metric. For more information on geodesic convexity, we refer to [111].

4.2 Convergence of Self-Consistent Field Iterations

In [112,113], several classical theoretical problems from KSDFT are studied. Under certain conditions, the equivalence between KS energy minimization problems and KS equations are established. In addition, a lower bound of nonzero elements of the charge density is also analyzed. By treating the KS equation as a fixed point equation with respect to a potential function, the Jacobian matrix is explicitly derived using the spectral operator theory and the theoretical properties of the SCF method are analyzed. It is proved that the second-order derivatives of the exchange-correlation energy are uniformly bounded if the Hamiltonian has a sufficiently large eigenvalue gap. Moreover, SCF converges from any initial point and enjoys a local linear convergence rate. Related results can be found in [22–24,56,114,115].

Specifically, consider the real case of KS equation (2.11), we define the potential function

$$V := \mathbb{V}(\rho) = L^{\dagger}\rho + \mu_{\mathrm{xc}}(\rho)^{\top} e \tag{4.1}$$

and

$$H(V) := \frac{1}{2}L + \sum_{l} \zeta_l w_l w_l^{\top} + V_{\mathrm{ion}} + \mathrm{Diag}(V). \tag{4.2}$$

Then, we have $H_{\mathrm{ks}}(\rho) = H(V)$. From (2.11), X are the eigenvectors corresponding to the p-smallest eigenvalues of $H(V)$, which is dependent on V. Then, a fixed point mapping for V can be written as

$$V = \mathbb{V}(F_{\phi}(V)), \tag{4.3}$$

where $F_{\phi}(V) = \mathrm{diag}(X(V)X(V)^{\top})$. Therefore, each iteration of SCF is to update V_k as

$$V_{k+1} = \mathbb{V}(F_{\phi}(V_k)). \tag{4.4}$$

For SCF with a simple charge-mixing strategy, the update scheme can be written as

$$V_{k+1} = V_k - \alpha(V_k - \mathbb{V}(F_{\phi}(V_k))), \tag{4.5}$$

where α is an appropriate step size. Under some mild assumptions, SCF converges with a local linear convergence rate.

Theorem 4.3 *Suppose that* $\lambda_{p+1}(H(V)) - \lambda_p(H(V)) > \delta,\ \forall V$, *the second-order derivatives of* $\varepsilon_{\mathrm{xc}}$ *are upper bounded and there is a constant* θ *such that* $\|L^{\dagger} + \frac{\partial \mu_{\mathrm{xc}}(\rho)}{\partial \rho} e\|_2 \leqslant \theta,\ \forall \rho \in \mathbb{R}^n$. *Let* $b_1 := 1 - \frac{\theta}{\delta} > 0$, $\{V_k\}$ *be a sequence generated by*

(4.5) *with a step size of* α *satisfying*

$$0 < \alpha < \frac{2}{2 - b_1}.$$

Then, $\{V_k\}$ *converges to a solution of the KS equation* (2.11)*, and its convergence rate is not worse than* $|1 - \alpha| + \alpha(1 - b_1)$.

4.3 Pursuing Global Optimality

In the Euclidean space, a common way to escape the local minimum is to add white noise to the gradient flow, which leads to a stochastic differential equation

$$\mathrm{d}X(t) = -\nabla f(X(t))\mathrm{d}t + \sigma(t)\mathrm{d}B(t),$$

where $B(t)$ is the standard n-by-p Brownian motion. A generalized noisy gradient flow on the Stiefel manifold is investigated in [116]

$$\mathrm{d}X(t) = -\mathrm{grad}\, f(X(t))\mathrm{d}t + \sigma(t) \circ \mathrm{d}B_{\mathcal{M}}(t),$$

where $B_{\mathcal{M}}(t)$ is the Brownian motion on the manifold $\mathcal{M} := \mathrm{St}(n, p)$. The construction of a Brownian motion is then given in an extrinsic form. Theoretically, it can converge to the global minima by assuming second-order continuity.

4.4 Community Detection

For community detection problems, a commonly used model is called the degree-correlated stochastic block model (DCSBM). It assumes that there are no overlaps between nodes in different communities. Specifically, the hypothesis node set $[n] = \{1, \cdots, n\}$ contains k communities, $\{C_1^*, \cdots, C_k^*\}$ satisfying

$$C_a^* \cap C_b^* = \varnothing, \forall a \neq b \text{ and } \cup_{a=1}^k C_a^* = [n].$$

In DCSBM, the network is a random graph, which can be represented by a matrix with all elements 0 to 1 represented by $B \in \mathbb{S}^k$. Let $A \in \{0, 1\}^{n \times n}$ be the adjacency matrix of this network and $A_{ii} = 0, \forall i \in [n]$. Then, for $i \in C_a^*$, $j \in C_b^*$, $i \neq j$,

$$A_{ij} = \begin{cases} 1, & \text{with probability } B_{ab}\theta_i\theta_j, \\ 0, & \text{with probability } 1 - B_{ab}\theta_i\theta_j, \end{cases}$$

where the heterogeneity of nodes is characterized by the vector θ. More specifically, larger θ_i corresponds to i with more edges connecting other nodes. For DCSBM, the aforementioned relaxation model (2.15) is proposed in [29]. By solving (2.15), an approximation of the global optimal solution can be obtained with high probability.

Theorem 4.4 *Define* $G_a = \sum_{i\in C_a^*}\theta_i$, $H_a = \sum_{b=1}^k B_{ab}G_b$, $f_i = H_a\theta_i$. *Let* U^* *and* Φ^* *be global optimal solutions for* (2.15) *and* (2.14), *respectively, and define* $\Delta = U^*(U^*)^\top - \Phi^*(\Phi^*)^\top$. *Suppose that* $\max_{1\leqslant a<b\leqslant k}\frac{B_{ab}+\delta}{H_aH_b} < \lambda < \min_{1\leqslant a\leqslant k}\frac{B_{aa}-\delta}{H_a^2}$ *for some* $\delta > 0$. *Then, with high probability, we have*

$$\|\Delta\|_{1,\theta} \leqslant \frac{C_0}{\delta}\left(1+\left(\max_{1\leqslant a\leqslant k}\frac{B_{aa}}{H_a^2}\|f\|_1\right)\right)(\sqrt{n\|f\|_1}+n),$$

where the constant $C_0 > 0$ *is independent with problem scale and parameter selections.*

4.5 The Maxcut Problem

Consider the SDP relaxation (2.2) and the non-convex relaxation problem with low-rank constraints (2.3). If $p \geqslant \sqrt{2n}$, the composition of a solution V_* of (2.3), i.e., $V_*^\top V_*$, is always an optimal solution of SDP (2.2) [117–119]. If $p \geqslant \sqrt{2n}$, for almost all matrices C, problem (2.3) has a unique local minimum and this minimum is also a global minimum of the original problem (2.1) [120]. The relationship between solutions of the two problems (2.2) and (2.3) is presented in [121]. Define $\mathrm{SDP}(C) = \max\{\langle C, X\rangle : X \succeq 0, X_{ii} = 1, i \in [n]\}$. A point $V \in \mathrm{Ob}(p,n)$ is called an ε-approximate concave point of (2.3), if

$$\langle U, \mathrm{Hess}\, f(V)[U]\rangle \leqslant \varepsilon\|U\|_V^2, \quad \forall U \in T_V\mathrm{Ob}(p,n),$$

where $f(V) = \langle C, V^\top V\rangle$. The following theorem [121, Theorem 1] tells the approximation quality of an ε-approximate concave point of (2.3).

Theorem 4.5 *For any* ε*-approximate concave point* V *of* (2.3), *we have*

$$\mathrm{tr}(CV^\top V) \geqslant \mathrm{SDP}(C) - \frac{1}{p-1}(\mathrm{SDP}(C)+\mathrm{SDP}(-C)) - \frac{n}{2}\varepsilon. \tag{4.6}$$

Another problem with similar applications is the $\mathbb{Z}_2$ synchronization problem [122]. Specifically, given noisy observations $Y_{ij} = z_iz_j + \sigma W_{ij}$, where $W_{ij} \sim \mathcal{N}(0,1)$ for $i > j$, $W_{ij} = W_{ji}$ for $i < j$ and $W_{ii} = 0$, we want to estimate the unknown labels $z_i \in \{\pm 1\}$. It can be seen as a special case of the maxcut problem with $p = 2$. The following results are presented in [122].

Theorem 4.6 *If* $\sigma < \frac{1}{8}\sqrt{n}$, *then, with a high probability, all second-order stable points* Q *of problem* (2.3) ($p = 2$) *have the following non-trivial relationship with the true label* z, *i.e., for each such* σ, *there is* ε *such that*

$$\frac{1}{n}\|Q^\top z\|_2 \geqslant \varepsilon.$$

4.6 Burer–Monteiro Factorizations of Smooth Semidefinite Programs

Consider the following SDP

$$\min_{X\in\mathbb{S}^n} \operatorname{tr}(CX) \quad \text{s.t. } \mathcal{A}(X) = b,\ X \succeq 0, \tag{4.7}$$

where $C \in \mathbb{S}^n$ is a cost matrix, $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$ is a linear operator and $\mathcal{A}(X) = b$ leads to m equality constraints on X, i.e., $\operatorname{tr}(A_i X) = b_i$ with $A_i \in \mathbb{S}^n$, $b \in \mathbb{R}^m$, $i = 1, \cdots, m$. Define $\mathcal{C}$ as the constraint set

$$\mathcal{C} = \{X \in \mathbb{S}^n : \mathcal{A}(X) = b,\ X \succeq 0\}.$$

If $\mathcal{C}$ is compact, it is proved in [117,118] that (4.7) has a global minimum of rank r with $\frac{r(r+1)}{2} \leqslant m$. This allows to use the Burer–Monteiro factorizations [119] (i.e., let $X = YY^\top$ with $Y \in \mathbb{R}^{n\times p}$, $\frac{p(p+1)}{2} \geqslant m$) to solve the following non-convex optimization problem

$$\min_{Y\in\mathbb{R}^{n\times p}} \operatorname{tr}(CYY^\top) \quad \text{s.t. } \mathcal{A}(YY^\top) = b. \tag{4.8}$$

Here, we define the constraint set

$$\mathcal{M} = \mathcal{M}_p := \{Y \in \mathbb{R}^{n\times p} : \mathcal{A}(YY^\top) = b\}. \tag{4.9}$$

Since $\mathcal{M}$ is non-convex, there may exist many non-global local minima of (4.8). It is claimed in [123] that each local minimum of (4.8) maps to a global minimum of (4.7) if $\frac{p(p+1)}{2} > m$. By utilizing the optimality theory of manifold optimization, any second-order stationary point can be mapped to a global minimum of (4.7) under mild assumptions [124]. Note that (4.9) is generally not a manifold. When the dimension of the space spanned by $\{A_1Y, \cdots, A_mY\}$, denoted by rank $\mathcal{A}$, is fixed for all Y, $\mathcal{M}_p$ defines a Riemannian manifold. Hence, we need the following assumptions.

Assumption 4.7 For a given p such that $\mathcal{M}_p$ is not empty, assume at least one of the following conditions are satisfied.

(SDP.1) $\{A_1Y, \cdots, A_mY\}$ are linearly independent in $\mathbb{R}^{n\times p}$ for all $Y \in M_p$
(SDP.2) $\{A_1Y, \cdots, A_mY\}$ span a subspace of constant dimension in $\mathbb{R}^{n\times p}$ for all Y in an open neighborhood of $\mathcal{M}_p \in \mathbb{R}^{n\times p}$.

By comparing the optimality conditions of (4.8) and the KKT conditions of (4.7), the following equivalence between (4.7) and (4.8) is established in [124, Theorem 1.4].

Theorem 4.8 *Let p satisfy $\frac{p(p+1)}{2} >$ rank $\mathcal{A}$. Suppose that Assumption 4.7 holds. For almost any cost matrix $C \in \mathbb{S}^n$, if $Y \in \mathcal{M}_p$ satisfies first- and second-order necessary optimality conditions for (4.8), then Y is globally optimal and $X = YY^\top$ is globally optimal for (4.7).*

4.7 Little Grothendieck Problem with Orthogonality Constraints

Given a positive semidefinite matrix $C \in \mathbb{R}^{dn\times dn}$, the little Grothendieck problem with orthogonality constraints can be expressed as

$$\max_{O_1,\cdots,O_d\in\mathcal{O}_d} \sum_{i=1}^{n}\sum_{j=1}^{n} \mathrm{tr}\left(C_{ij}^{\top} O_i O_j^{\top}\right), \tag{4.10}$$

where C_{ij} represents the (i, j)th $d \times d$ block of C, $\mathcal{O}_d$ is a group of $d \times d$ orthogonal matrices (i.e., $O \in \mathcal{O}_d$ if and only if $O^{\top} O = OO^{\top} = I$.) A SDP relaxation of (4.10) is as follows: [125]

$$\max_{\substack{G\in\mathbb{R}^{dn\times dn}\\ G_{ii}=I_{d\times d},\ G\succeq 0}} \mathrm{tr}(CG). \tag{4.11}$$

For the original problem (4.10), a randomized approximation algorithm is presented in [125]. Specifically, it consists of the following two procedures.

- Let G be a solution to problem (4.11). Denote by the Cholesky decomposition $G = LL^{\top}$. Let X_i be a $d \times (nd)$ matrix such that $L = (X_1^{\top}, X_2^{\top}, \cdots, X_n^{\top})^{\top}$.
- Let $R \in \mathbb{R}^{(nd)\times d}$ be a real-valued Gaussian random matrix whose entries are i.i.d. $\mathcal{N}(0, \frac{1}{d})$. The approximate solution of the problem (4.10) can be calculated as follows:

$$V_i = \mathcal{P}(X_i R),$$

where $\mathcal{P}(Y) = \arg\min_{Z\in\mathcal{O}_d} \|Z - Y\|_F$ with $Y \in \mathbb{R}^{d\times d}$.

For the solution obtained in the above way, a constant approximation ratio on the objective function value is shown, which recovers the known $\frac{2}{\pi}$ approximation guarantee for the classical little Grothendieck problem.

Theorem 4.9 *Let $V_1, \cdots, V_n \in \mathcal{O}_d$ be obtained as above. For being given a symmetric matrix $C \geqslant 0$, then*

$$\mathbf{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} \mathrm{tr}\left(C_{ij}^{\top} V_i V_j^{\top}\right)\right] \geqslant \alpha(d)^2 \max_{O_1,\cdots,O_n\in\mathcal{O}_d} \sum_{i=1}^{n}\sum_{j=1}^{n} \mathrm{tr}\left(C_{ij}^{\top} O_i O_j^{\top}\right),$$

where

$$\alpha(d) := \mathbf{E}\left[\frac{1}{d}\sum_{j=1}^{d}\sigma_j(Z)\right],$$

$Z \in \mathbb{R}^{d\times d}$ is a Gaussian random matrix whose components i.i.d. $\mathcal{N}(0, \frac{1}{d})$ and $\sigma_j(Z)$ is the jth singular value of Z.

5 Conclusions

Manifold optimization has been extensively studied in the literature. We review the definition of manifold optimization, a few related applications, algorithms and analysis. However, there are still many issues and challenges. Many manifold optimization problems that can be effectively solved are still limited to relatively simple structures such as orthogonal constraints and rank constraints. For other manifolds with complicated structures, what are the most efficient choices of Riemannian metrics and retraction operators are not obvious. Another interesting topic is to combine the manifold structure with the characteristics of specific problems and applications, such as graph-based data analysis, real-time data flow analysis and biomedical image analysis. Non-smooth problems appear to be more and more attractive.

Acknowledgements The authors are grateful to the associate editor and two anonymous referees for their detailed and valuable comments and suggestions.

References

[1] Lai, R., Wen, Z., Yin, W., Gu, X., Lui, L.M.: Folding-free global conformal mapping for genus-0 surfaces by harmonic energy minimization. J. Sci. Comput. **58**, 705–725 (2014)
[2] Schoen, R.M., Yau, S.-T.: Lectures on Harmonic Maps, vol. 2. American Mathematical Society, Providence (1997)
[3] Simon, D., Abell, J.: A majorization algorithm for constrained correlation matrix approximation. Linear Algebra Appl. **432**, 1152–1164 (2010)
[4] Gao, Y., Sun, D.: A majorized penalty approach for calibrating rank constrained correlation matrix problems, tech. report, National University of Singapore (2010)
[5] Waldspurger, I., d'Aspremont, A., Mallat, S.: Phase recovery, maxcut and complex semidefinite programming. Math. Program. **149**, 47–81 (2015)
[6] Cai, J.-F., Liu, H., Wang, Y.: Fast rank-one alternating minimization algorithm for phase retrieval. J. Sci. Comput. **79**, 128–147 (2019)
[7] Hu, J., Jiang, B., Liu, X., Wen, Z.: A note on semidefinite programming relaxations for polynomial optimization over a single sphere. Sci. China Math. **59**, 1543–1560 (2016)
[8] Singer, A., Shkolnisky, Y.: Three-dimensional structure determination from common lines in cryo-em by eigenvectors and semidefinite programming. SIAM J. Imaging Sci. **4**, 543–572 (2011)
[9] Liu, X., Wen, Z., Zhang, Y.: An efficient Gauss–Newton algorithm for symmetric low-rank product matrix approximations. SIAM J. Optim. **25**, 1571–1608 (2015)
[10] Liu, X., Wen, Z., Zhang, Y.: Limited memory block Krylov subspace optimization for computing dominant singular value decompositions. SIAM J. Sci. Comput. **35**, A1641–A1668 (2013)
[11] Wen, Z., Yang, C., Liu, X., Zhang, Y.: Trace-penalty minimization for large-scale eigenspace computation. J. Sci. Comput. **66**, 1175–1203 (2016)
[12] Wen, Z., Zhang, Y.: Accelerating convergence by augmented Rayleigh–Ritz projections for large-scale eigenpair computation. SIAM J. Matrix Anal. Appl. **38**, 273–296 (2017)

[13] Zhang, J., Wen, Z., Zhang, Y.: Subspace methods with local refinements for eigenvalue computation using low-rank tensor-train format. J. Sci. Comput. **70**, 478–499 (2017)
[14] Oja, E., Karhunen, J.: On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. J. Math. Anal. Appl. **106**, 69–84 (1985)
[15] Shamir, O.: A stochastic PCA and SVD algorithm with an exponential convergence rate. Int. Conf. Mach. Learn. 144–152 (2015)
[16] Li, C.J., Wang, M., Liu, H., Zhang, T.: Near-optimal stochastic approximation for online principal component estimation. Math. Program. **167**, 75–97 (2018)
[17] Pulay, P.: Convergence acceleration of iterative sequences. The case of SCF iteration. Chem. Phys. Lett. **73**, 393–398 (1980)
[18] Pulay, P.: Improved SCF convergence acceleration. J. Comput. Chem. **3**, 556–560 (1982)
[19] Toth, A., Ellis, J.A., Evans, T., Hamilton, S., Kelley, C., Pawlowski, R., Slattery, S.: Local improvement results for Anderson acceleration with inaccurate function evaluations. SIAM J. Sci. Comput. **39**, S47–S65 (2017)
[20] Zhang, X., Zhu, J., Wen, Z., Zhou, A.: Gradient type optimization methods for electronic structure calculations. SIAM J. Sci. Comput. **36**, C265–C289 (2014)
[21] Wen, Z., Milzarek, A., Ulbrich, M., Zhang, H.: Adaptive regularized self-consistent field iteration with exact Hessian for electronic structure calculation. SIAM J. Sci. Comput. **35**, A1299–A1324 (2013)
[22] Dai, X., Liu, Z., Zhang, L., Zhou, A.: A conjugate gradient method for electronic structure calculations. SIAM J. Sci. Comput. **39**, A2702–A2740 (2017)
[23] Zhao, Z., Bai, Z.-J., Jin, X.-Q.: A Riemannian Newton algorithm for nonlinear eigenvalue problems. SIAM J. Matrix Anal. Appl. **36**, 752–774 (2015)
[24] Zhang, L., Li, R.: Maximization of the sum of the trace ratio on the Stiefel manifold, II: computation. Sci. China Math. **58**, 1549–1566 (2015)
[25] Gao, B., Liu, X., Chen, X., Yuan, Y.: A new first-order algorithmic framework for optimization problems with orthogonality constraints. SIAM J. Optim. **28**, 302–332 (2018)
[26] Lai, R., Lu, J.: Localized density matrix minimization and linear-scaling algorithms. J. Comput. Phys. **315**, 194–210 (2016)
[27] Ulbrich, M., Wen, Z., Yang, C., Klockner, D., Lu, Z.: A proximal gradient method for ensemble density functional theory. SIAM J. Sci. Comput. **37**, A1975–A2002 (2015)
[28] Jiang, B., Liu, Y.-F., Wen, Z.: L_p-norm regularization algorithms for optimization over permutation matrices. SIAM J. Optim. **26**, 2284–2313 (2016)
[29] Zhang, J., Liu, H., Wen, Z., Zhang, S.: A sparse completely positive relaxation of the modularity maximization for community detection. SIAM J. Sci. Comput. **40**, A3091–A3120 (2018)
[30] Cho, M., Lee, J.: Riemannian approach to batch normalization. Adv. Neural Inf. Process. Syst. 5225–5235 (2017). https://papers.nips.cc/paper/7107-riemannian-approach-to-batch-normalization.pdf
[31] Jolliffe, I.T., Trendafilov, N.T., Uddin, M.: A modified principal component technique based on the lasso. J. Comput. Graph. Stat. **12**, 531–547 (2003)
[32] Wen, Z., Yin, W., Zhang, Y.: Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. Math. Program. Comput. **4**, 333–361 (2012)
[33] Vandereycken, B.: Low-rank matrix completion by Riemannian optimization. SIAM J. Optim. **23**, 1214–1236 (2013)
[34] Wei, K., Cai, J.-F., Chan, T.F., Leung, S.: Guarantees of Riemannian optimization for low rank matrix recovery. SIAM J. Matrix Anal. Appl. **37**, 1198–1222 (2016)
[35] Cambier, L., Absil, P.-A.: Robust low-rank matrix completion by Riemannian optimization. SIAM J. Sci. Comput. **38**, S440–S460 (2016)
[36] Zhang, Y., Lau, Y., Kuo, H.-w., Cheung, S., Pasupathy, A., Wright, J.: On the global geometry of sphere-constrained sparse blind deconvolution. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 4894–4902 (2017)
[37] Zass, R., Shashua, A.: Nonnegative sparse PCA. Adv. Neural Inf. Process. Syst. 1561–1568 (2007). https://papers.nips.cc/paper/3104-nonnegative-sparse-pca
[38] Montanari, A., Richard, E.: Non-negative principal component analysis: message passing algorithms and sharp asymptotics. IEEE Trans. Inf. Theory **62**, 1458–1484 (2016)
[39] Carson, T., Mixon, D.G., Villar, S.: Manifold optimization for K-means clustering. Int. Conf. Sampl. Theory. Appl. SampTA 73–77. IEEE (2017)

[40] Liu, H., Cai, J.-F., Wang, Y.: Subspace clustering by (k, k)-sparse matrix factorization. Inverse Probl. Imaging **11**, 539–551 (2017)
[41] Xie, T., Chen, F.: Non-convex clustering via proximal alternating linearized minimization method. Int. J. Wavelets Multiresolut. Inf. Process. **16**, 1840013 (2018)
[42] Absil, P.-A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton, NJ (2008)
[43] Absil, P.-A., Gallivan, K.A.: Joint diagonalization on the oblique manifold for independent component analysis. Proc. IEEE Int. Conf. Acoust. Speech Signal Process 5, 945–958 (2006)
[44] Bhatia, R.: Positive Definite Matrices, vol. 24. Princeton University Press, Princeton (2009)
[45] Journée, M., Bach, F., Absil, P.-A., Sepulchre, R.: Low-rank optimization on the cone of positive semidefinite matrices. SIAM J. Optim. **20**, 2327–2351 (2010)
[46] Massart, E., Absil, P.-A.: Quotient geometry with simple geodesics for the manifold of fixed-rank positive-semidefinite matrices. SIAM J. Matrix Anal. Appl. **41**, 171–198 (2020)
[47] Yang, W.H., Zhang, L.-H., Song, R.: Optimality conditions for the nonlinear programming problems on Riemannian manifolds. Pac. J. Optim. **10**, 415–434 (2014)
[48] Gabay, D.: Minimizing a differentiable function over a differential manifold. J. Optim. Theory Appl. **37**, 177–219 (1982)
[49] Smith, S.T.: Optimization techniques on Riemannian manifolds. Fields Institute Communications **3** (1994)
[50] Kressner, D., Steinlechner, M., Vandereycken, B.: Low-rank tensor completion by Riemannian optimization. BIT Numer. Math. **54**, 447–468 (2014)
[51] Hu, J., Milzarek, A., Wen, Z., Yuan, Y.: Adaptive quadratically regularized Newton method for Riemannian optimization. SIAM J. Matrix Anal. Appl. **39**, 1181–1207 (2018)
[52] Absil, P.-A., Malick, J.: Projection-like retractions on matrix manifolds. SIAM J. Optim. **22**, 135–158 (2012)
[53] Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
[54] Wen, Z., Yin, W.: A feasible method for optimization with orthogonality constraints. Math. Program. **142**, 397–434 (2013)
[55] Jiang, B., Dai, Y.-H.: A framework of constraint preserving update schemes for optimization on Stiefel manifold. Math. Program. **153**, 535–575 (2015)
[56] Zhu, X.: A Riemannian conjugate gradient method for optimization on the Stiefel manifold. Comput. Optim. Appl. **67**, 73–110 (2017)
[57] Siegel, J.W.: Accelerated optimization with orthogonality constraints, arXiv:1903.05204 (2019)
[58] Iannazzo, B., Porcelli, M.: The Riemannian Barzilai–Borwein method with nonmonotone line search and the matrix geometric mean computation. IMA J. Numer. Anal. **00**, 1–23 (2017)
[59] Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl. **20**, 303–353 (1999)
[60] Nishimori, Y., Akaho, S.: Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold. Neurocomputing **67**, 106–135 (2005)
[61] Huang, W.: Optimization algorithms on Riemannian manifolds with applications, Ph.D. thesis, The Florida State University (2013)
[62] Lezcano-Casado, M., Martínez-Rubio, D.: Cheap orthogonal constraints in neural networks: a simple parametrization of the orthogonal and unitary group, arXiv:1901.08428 (2019)
[63] Li, J., Fuxin, L., Todorovic, S.: Efficient Riemannian optimization on the Stiefel manifold via the Cayley transform, Conference arXiv:2002.01113 (2020)
[64] Huang, W., Gallivan, K.A., Absil, P.-A.: A Broyden class of quasi-Newton methods for Riemannian optimization. SIAM J. Optim. **25**, 1660–1685 (2015)
[65] Huang, W., Absil, P.-A., Gallivan, K.A.: Intrinsic representation of tangent vectors and vector transports on matrix manifolds. Numer. Math. **136**, 523–543 (2017)
[66] Hu, J., Milzarek, A., Wen, Z., Yuan, Y.: Adaptive regularized Newton method for Riemannian optimization, arXiv:1708.02016 (2017)
[67] Zhang, H., Hager, W.W.: A nonmonotone line search technique and its application to unconstrained optimization. SIAM J. Optim. **14**, 1043–1056 (2004)
[68] Udriste, C.: Convex Functions and Optimization Methods on Riemannian Manifolds, vol. 297. Springer, Berlin (1994)

[69] Absil, P.-A., Baker, C.G., Gallivan, K.A.: Trust-region methods on Riemannian manifolds. Found. Comput. Math. **7**, 303–330 (2007)
[70] Qi, C.: Numerical optimization methods on Riemannian manifolds, Ph.D. thesis, Florida State University (2011)
[71] Ring, W., Wirth, B.: Optimization methods on Riemannian manifolds and their application to shape space. SIAM J. Optim. **22**, 596–627 (2012)
[72] Seibert, M., Kleinsteuber, M., Hüper, K.: Properties of the BFGS method on Riemannian manifolds. Mathematical System Theory C Festschrift in Honor of Uwe Helmke on the Occasion of his Sixtieth Birthday, pp. 395–412 (2013)
[73] Huang, W., Absil, P.-A., Gallivan, K.A.: A Riemannian symmetric rank-one trust-region method. Math. Program. **150**, 179–216 (2015)
[74] Huang, W., Absil, P.-A., Gallivan, K.: A Riemannian BFGS method without differentiated retraction for nonconvex optimization problems. SIAM J. Optim. **28**, 470–495 (2018)
[75] Hu, J., Jiang, B., Lin, L., Wen, Z., Yuan, Y.-X.: Structured quasi-Newton methods for optimization with orthogonality constraints. SIAM J. Sci. Comput. **41**, A2239–A2269 (2019)
[76] Nocedal, J., Wright, S.J.: Numerical Optimization. Springer Series in Operations Research and Financial Engineering, 2nd edn. Springer, New York (2006)
[77] Wu, X., Wen, Z., Bao, W.: A regularized Newton method for computing ground states of Bose–Einstein condensates. J. Sci. Comput. **73**, 303–329 (2017)
[78] Kass, R.E.: Nonlinear regression analysis and its applications. J. Am. Stat. Assoc. **85**, 594–596 (1990)
[79] Sun, W., Yuan, Y.: Optimization Theory and Methods: Nonlinear Programming, vol. 1. Springer, Berlin (2006)
[80] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436 (2015)
[81] Bonnabel, S.: Stochastic gradient descent on Riemannian manifolds. IEEE Trans. Autom. Control. **58**, 2217–2229 (2013)
[82] Zhang, H., Sra, S.: First-order methods for geodesically convex optimization, In: Conference on Learning Theory, pp. 1617–1638 (2016)
[83] Liu, Y., Shang, F., Cheng, J., Cheng, H., Jiao, L.: Accelerated first-order methods for geodesically convex optimization on Riemannian manifolds. Adv. Neural Inf. Process. Syst. 4868–4877 (2017)
[84] Zhang, H., Reddi, S.J., Sra, S.: Riemannian SVRG: fast stochastic optimization on Riemannian manifolds. Adv. Neural Inf. Process. Syst. 4592–4600 (2016)
[85] Sato, H., Kasai, H., Mishra, B.: Riemannian stochastic variance reduced gradient algorithm with retraction and vector transport. SIAM J. Optim. **29**, 1444–1472 (2019)
[86] Jiang, B., Ma, S., So, A.M.-C., Zhang, S.: Vector transport-free svrg with general retraction for Riemannian optimization: Complexity analysis and practical implementation, arXiv:1705.09059 (2017)
[87] Bécigneul, G., Ganea, O.-E.: Riemannian adaptive optimization methods, arXiv:1810.00760 (2018)
[88] Dirr, G., Helmke, U., Lageman, C.: Nonsmooth Riemannian optimization with applications to sphere packing and grasping. In: Lagrangian and Hamiltonian Methods for Nonlinear Control 2006, pp. 29–45. Springer, Berlin (2007)
[89] Borckmans, P.B., Selvan, S.E., Boumal, N., Absil, P.-A.: A Riemannian subgradient algorithm for economic dispatch with valve-point effect. J Comput. Appl. Math. **255**, 848–866 (2014)
[90] Hosseini, S.: Convergence of nonsmooth descent methods via Kurdyka–Lojasiewicz inequality on Riemannian manifolds, Hausdorff Center for Mathematics and Institute for Numerical Simulation, University of Bonn (2015). https://ins.uni-bonn.de/media/public/publication-media/8_INS1523.pdf
[91] Grohs, P., Hosseini, S.: Nonsmooth trust region algorithms for locally Lipschitz functions on Riemannian manifolds. IMA J. Numer. Anal. **36**, 1167–1192 (2015)
[92] Hosseini, S., Uschmajew, A.: A Riemannian gradient sampling algorithm for nonsmooth optimization on manifolds. SIAM J. Optim. **27**, 173–189 (2017)
[93] Bacák, M., Bergmann, R., Steidl, G., Weinmann, A.: A second order nonsmooth variational model for restoring manifold-valued images. SIAM J. Sci. Comput. **38**, A567–A597 (2016)
[94] de Carvalho Bento, G., da Cruz Neto, J.X., Oliveira, P.R.: A new approach to the proximal point method: convergence on general Riemannian manifolds. J Optim. Theory Appl. **168**, 743–755 (2016)
[95] Bento, G., Neto, J., Oliveira, P.: Convergence of inexact descent methods for nonconvex optimization on Riemannian manifolds, arXiv:1103.4828 (2011)

[96] Bento, G.C., Ferreira, O.P., Melo, J.G.: Iteration-complexity of gradient, subgradient and proximal point methods on Riemannian manifolds. J Optim. Theory Appl. **173**, 548–562 (2017)
[97] Chen, S., Ma, S., So, A.M.-C., Zhang, T.: Proximal gradient method for nonsmooth optimization over the Stiefel manifold. SIAM J. Optim. **30**, 210–239 (2019)
[98] Xiao, X., Li, Y., Wen, Z., Zhang, L.: A regularized semi-smooth Newton method with projection steps for composite convex programs. J. Sci. Comput. **76**, 1–26 (2018)
[99] Huang, W., Wei, K.: Riemannian proximal gradient methods, arXiv:1909.06065 (2019)
[100] Chen, S., Ma, S., Xue, L., Zou, H.: An alternating manifold proximal gradient method for sparse PCA and sparse cca, arXiv:1903.11576 (2019)
[101] Huang, W., Wei, K.: Extending FISTA to Riemannian optimization for sparse PCA, arXiv:1909.05485 (2019)
[102] Lai, R., Osher, S.: A splitting method for orthogonality constrained problems. J. Sci. Comput. **58**, 431–449 (2014)
[103] Kovnatsky, A., Glashoff, K., Bronstein, M.M.: Madmm: a generic algorithm for non-smooth optimization on manifolds. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision ECCV, pp. 680–696. Springer, Berlin (2016)
[104] Wang, Y., Yin, W., Zeng, J.: Global convergence of admm in nonconvex nonsmooth optimization. J. Sci. Comput. **78**, 29–63 (2019)
[105] Zhang, J., Ma, S., Zhang, S.: Primal-dual optimization algorithms over Riemannian manifolds: an iteration complexity analysis, arXiv:1710.02236 (2017)
[106] Birgin, E.G., Haeser, G., Ramos, A.: Augmented lagrangians with constrained subproblems and convergence to second-order stationary points. Comput. Optim. Appl. **69**, 51–75 (2018)
[107] Liu, C., Boumal, N.: Simple algorithms for optimization on Riemannian manifolds with constraints, arXiv:1901.10000 (2019)
[108] Boumal, N., Absil, P.-A., Cartis, C.: Global rates of convergence for nonconvex optimization on manifolds. IMA J. Numer. Anal. **39**, 1–33 (2018)
[109] Zhang, J., Zhang, S.: A cubic regularized Newton's method over Riemannian manifolds, arXiv:1805.05565 (2018)
[110] Agarwal, N., Boumal, N., Bullins, B., Cartis, C.: Adaptive regularization with cubics on manifolds with a first-order analysis, arXiv:1806.00065 (2018)
[111] Vishnoi, N.K.: Geodesic convex optimization: differentiation on manifolds, geodesics, and convexity, arXiv:1806.06373 (2018)
[112] Liu, X., Wang, X., Wen, Z., Yuan, Y.: On the convergence of the self-consistent field iteration in Kohn–Sham density functional theory. SIAM J. Matrix Anal. Appl. **35**, 546–558 (2014)
[113] Liu, X., Wen, Z., Wang, X., Ulbrich, M., Yuan, Y.: On the analysis of the discretized Kohn-Sham density functional theory. SIAM J. Numer. Anal. **53**, 1758–1785 (2015)
[114] Cai, Y., Zhang, L.-H., Bai, Z., Li, R.-C.: On an eigenvector-dependent nonlinear eigenvalue problem. SIAM J. Matrix Anal. Appl. **39**, 1360–1382 (2018)
[115] Bai, Z., Lu, D., Vandereycken, B.: Robust Rayleigh quotient minimization and nonlinear eigenvalue problems. SIAM J. Sci. Comput. **40**, A3495–A3522 (2018)
[116] Yuan, H., Gu, X., Lai, R., Wen, Z.: Global optimization with orthogonality constraints via stochastic diffusion on manifold. J. Sci. Comput. **80**, 1139–1170 (2019)
[117] Barvinok, A.I.: Problems of distance geometry and convex properties of quadratic maps. Discrete Comput. Geom. **13**, 189–202 (1995)
[118] Pataki, G.: On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. Math. Oper. Res. **23**, 339–358 (1998)
[119] Burer, S., Monteiro, R.D.: A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. Math. Program. **95**, 329–357 (2003)
[120] Boumal, N., Voroninski, V., Bandeira, A.: The non-convex Burer–Monteiro approach works on smooth semidefinite programs. In: Advances in Neural Information Processing Systems, pp. 2757–2765 (2016). https://papers.nips.cc/paper/6517-the-non-convex-burer-monteiro-approach-works-on-smooth-semidefinite-programs.pdf
[121] Mei, S., Misiakiewicz, T., Montanari, A., Oliveira, R.I.: Solving SDPs for synchronization and maxcut problems via the Grothendieck inequality, arXiv:1703.08729 (2017)
[122] Bandeira, A.S., Boumal, N., Voroninski, V.: On the low-rank approach for semidefinite programs arising in synchronization and community detection. Conf. Learn. Theor. 361–382 (2016)

[123] Burer, S., Monteiro, R.D.: Local minima and convergence in low-rank semidefinite programming. Math. Program. **103**, 427–444 (2005)
[124] Boumal, N., Voroninski, V., Bandeira, A.S.: Deterministic guarantees for Burer–Monteiro factorizations of smooth semidefinite programs, arXiv:1804.02008 (2018)
[125] Bandeira, A.S., Kennedy, C., Singer, A.: Approximating the little Grothendieck problem over the orthogonal and unitary groups. Math. Program. **160**, 433–475 (2016)