2013

Optimization Algorithms on Riemannian Manifolds with Applications

Wen Huang

FLORIDA STATE UNIVERSITY

COLLEGE OF ARTS AND SCIENCES

OPTIMIZATION ALGORITHMS ON RIEMANNIAN MANIFOLDS WITH APPLICATIONS

By

WEN HUANG

A Dissertation submitted to the
Department of Mathematics
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Degree Awarded:
Fall Semester, 2013

Wen Huang defended this dissertation on November 5th, 2013.
The members of the supervisory committee were:

Kyle A. Gallivan
Professor Co-Directing Dissertation

Pierre-Antoine Absil
Professor Co-Directing Dissertation

Dennis Duke
University Representative

Giray Okten
Committee Member

Eric P. Klassen
Committee Member

The Graduate School has verified and approved the above-named committee members, and certifies that the dissertation has been approved in accordance with university requirements.

This work is dedicated to my parents and my family, for their constant encouragement.

ACKNOWLEDGMENTS

Thank my advisor, my co-advisor, my committee members, my parents, my sister, my friends...

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

ABSTRACT

This dissertation generalizes three well-known unconstrained optimization approaches for $\mathbb{R}^n$ to solve optimization problems with constraints that can be viewed as a d-dimensional Riemannian manifold to obtain the Riemannian Broyden family of methods, the Riemannian symmetric rank-one trust region method, and Riemannian gradient sampling method. The generalization relies on basic differential geometric concepts, such as tangent spaces, Riemannian metrics, and the Riemannian gradient, as well as on the more recent notions of (first-order) retraction and vector transport. The effectiveness of the methods and techniques for their efficient implementation are derived and evaluated. Basic experiments and applications are used to illustrate the value of the proposed methods.

Both the Riemannian symmetric rank-one trust region method and the RBroyden family of methods are generalized from Euclidean quasi-Newton optimization methods, in which a Hessian approximation exploits the well-known secant condition. The generalization of the secant condition and the associated update formulas that define quasi-Newton methods to the Riemannian setting is a key result of this dissertation.

The dissertation also contains convergence theory for these methods. The Riemannian symmetric rank-one trust region method is shown to converge globally to a stationary point and $d+1$-step q-superlinearly to a minimizer of the objective function. The RBroyden family of methods is shown to converge globally and q-superlinearly to a minimizer of a retraction-convex objective function. A condition, called the locking condition, on vector transport and retraction that guarantees convergence for the RBroyden family method and facilitates efficient computation is derived and analyzed. The Dennis Moré sufficient and necessary conditions for superlinear convergence, can be generalized to the Riemannian setting in multiple ways. This dissertation generalizes them in a novel manner that is applicable to both Riemannian optimization problems and root finding for a vector field on a Riemannian manifold.

The convergence analyses of Riemannian symmetric rank-one trust region method and RBroyden family methods assume a smooth objective function. For partly smooth Lipschitz continuous objective functions, a variation of one of the RBroyden family methods, RBFGS, is shown to be work well empirically. In addition, the Riemannian gradient sampling method is shown to work well empirically for both a Lipschitz continuous and a non-Lipschitz continuous objective function associated with the important application nonlinear dimension reduction.

Efficient and effective implementations for a manifold in $\mathbb{R}^n$, a quotient manifold of total manifold in $\mathbb{R}^n$ and a product of manifolds, are presented. Results include efficient representations and operations of elements in a manifold, tangent vectors, linear operators, retractions and vector transports. Novel techniques for constructing and computing multiple kinds of vector transports are derived. In addition, the implementation details of all required objects for optimization on four manifolds, the Stiefel manifold, the sphere, the orthogonal group and the Grassmann manifold, are presented.

Basic numerical experiments for the Brockett cost function on the Stiefel manifold, the Rayleigh quotient on the Grassmann manifold and the minmax problem on the sphere (Lipschitz and non-Lipschitz forms), are used to illustrate the performance of the proposed methods and compare with existing optimization methods on manifolds. Three applications, Riemannian optimization for elastic shape analysis, a joint diagonalization problem for independent component analysis and a synchronization of rotation problem, that have smooth cost functions are used to show the advantages of the proposed methods. A secant-based nonlinear dimension reduction problem with a partly smooth function is used to show the advantages of the Riemannian gradient sampling method.

CHAPTER 1

INTRODUCTION

The dissertation investigates quasi-Newton optimization algorithms on a Riemannian manifold, optimization algorithms for partly smooth functions on a Riemannian manifold, and their analysis, implementation and evaluation. This is achieved by identifying key components of Riemannian optimization algorithms, analyzing the theoretical properties that influence the convergence of the associated algorithms, developing novel algorithms and implementations that are significantly more efficient than simple generalizations from $\mathbb{R}^n$ while achieving rigorously guaranteed convergence and applying them to a set of important problems.

The dissertation is organized as follows. In Chapter 1, an overview of the optimization problem on a Riemannian manifold is given followed by basic concepts of manifolds, a brief history of research on methods for optimization on manifolds and a summary of the basic principles on which the associated algorithms are built. The chapter ends with an overview of the research and the dissertation statement of the dissertation. In preparation for Chapters 3 and 4, Chapter 2 presents the fundamental concept, the secant condition, and methods of generalizing it to a Riemannian manifold. Chapter 3 presents the details of combining a trust region strategy with the Symmetric Rank-1 update algorithm and its theoretical analysis. In Chapters 4, 5 and 6, the Broyden family of algorithms combined with line search is defined and analyzed. The algorithm family and basic convergence analysis are given in Chapter 4; Necessary and sufficient conditions for superlinear convergence of quasi-Newton methods and inexact Newton methods, including Riemannian versions of Dennis Moré conditions, are proven in Chapter 5; and an analysis of the rate of convergence is presented in Chapter 6. Chapter 7 discusses a Riemannian gradient sampling algorithm used for the optimization of partly smooth functions. The relationship between methods for Euclidean constrained optimization and the methods derived in this dissertation for the optimization on a submanifold of the Euclidean space is discussed in Chapter 8. The crucial issue of efficient implementation of Riemannian optimization algorithms and their key primitives is discussed in Chapters 9 and 10. This includes a specific discussion of four manifolds: the Stiefel manifold, the sphere, the orthogonal group and the Grassmann manifold. Basic experimental comparisons of the performance of the algorithms proposed in this dissertation are presented in Chapter 11. Chapters 12 and 13 demonstrate the effectiveness the proposed algorithms for three applications with smooth cost functions: soft independent component analysis, synchronization of rotations and shape analysis using the elastic metric of Srivastava et al. [SKJJ11]. Chapter 14 considers an application with a smooth and a partly smooth cost function: secant-based nonlinear dimension reduction. Finally, conclusions are drawn and future work suggested in Chapter 15.

1.1 The Problem of Optimization on a Manifold

Optimization on Riemannian manifolds or Riemannian optimization considers finding an optimum of a real-valued function f defined on a Riemannian manifold, i.e.,

$$\min f(x), \text{ subject to } x \in \mathcal{M}, \tag{1.1.1}$$

where $\mathcal{M}$ is a Riemannian manifold. Roughly speaking, a manifold is a set endowed with coordinate patches that overlap smoothly.

One possible method to do optimization on manifold is to read the objective function f from coordinate patches. In this case, the problem becomes a classical optimization problem defined on an open subset of $\mathbb{R}^d$, where d is the dimension of the manifold. However, there are several reasons not to do this. First, the coordinate patches may not be available explicitly. Second, even if they exist, for each patch, we have to solve a constrained optimization for the subset of $\mathbb{R}^d$ which is not cheap. Third, when and how to change from one coordinate patch to another coordinate patch is problematic. What is more, some useful properties of the manifold may not be used by this idea.

If the manifold $\mathcal{M}$ is a subset of $\mathbb{R}^n$, then the problem (1.1.1) can be viewed as a Euclidean constrained optimization problem. The comparison of the Euclidean constrained optimization and Riemannian optimization with manifold in $\mathbb{R}^n$ is discussed in Chapter 8. The advantages of Riemannian optimization with manifold in $\mathbb{R}^n$ are:

1. All the iterates are on the manifold, i.e., they satisfy the constraints; this property allows us to stop the iteration early.

2. Optimization on manifold algorithms have the convergence properties of unconstrained optimization algorithms since these algorithms is to solve an unconstrained optimization over a constrained set.

3. There is no need to consider Lagrange multipliers or penalty functions.
4. Riemannian optimization is also a way of avoiding the Maratos effect.

If the objective function f has some continuous invariance properties, then optimization on manifold provides an approach to eliminate the invariance. There are several reasons to do this: efficiency; consistency; applicability of certain convergence results; avoidance of failure in certain algorithms, e.g., Newton's method, that do not behave satisfactorily in case of degeneracy.

The problem of minimizing a smooth objective function f on a Riemannian manifold has been a topic of much interest over the past few years due to several important applications. Recently considered applications include matrix completion problems [BA11, MMS11, Van12, DKM12], truss optimization [RW12], finite-element discretization of Cosserat rods [San10], matrix mean computation [BI13, ATV13], image and video-based recognition [TVSC11], electrostatics and electronic structure calculation [WY12], finance and chemistry [Bor12], multilinear algebra [SL10, IAVD11], low-rank learning [MMBS11, BA11], and blind source separation [KS12, SAGQ12]. Research efforts to develop and analyze optimization methods on manifolds can be traced back to the work of Luenberger [Lue72]. They include, among others, steepest-descent methods [Lue72], conjugate gradients [Smi94], Newton's method [Smi94, ADM02], and trust region methods [ABG07, BAG08]; see also [AMS08] for an overview.

1.2 Basic Principles

1.2.1 Unconstrained Optimization on a Constrained Space

To define a manifold, let us first define a chart and atlas. Consider a set $\mathcal{M}$. A one-to-one mapping ϕ from $\mathcal{U} \subset \mathcal{M}$ to an open subset of $\mathcal{R}^d$ is called a d-dimensional chart of the set $\mathcal{M}$. If a collection of charts $(\mathcal{U}_\alpha, \phi_\alpha)$ satisfies the following, then we call this collection an atlas:

1. $\bigcup_\alpha = \mathcal{M}$.
2. For any α, β with $\mathcal{U}_\alpha \bigcap \mathcal{U}_\beta \neq 0$, the sets $\phi_\alpha(\mathcal{U}_\alpha \bigcap \mathcal{U}_\beta)$ and $\phi_\beta(\mathcal{U}_\alpha \bigcap \mathcal{U}_\beta)$ are open sets in $\mathcal{R}^d$ and the change of coordinates
$$\phi_\beta \circ \phi_\alpha^{-1} : \mathbb{R}^d \to \mathbb{R}^d$$
is smooth. (We say that the elements of an atlas overlap smoothly.)

Two atlases $\mathcal{A}_1$ and $\mathcal{A}_2$ are equivalent if $\mathcal{A}_1 \cup \mathcal{A}_2$ is still an atlas. A (d-dimensional) manifold is a couple $(\mathcal{M}, \mathcal{A}^+)$, where $\mathcal{A}^+$ is a maximal atlas of $\mathcal{M}$ into $\mathbb{R}^d$, such that the topology induced by $\mathcal{A}^+$ is Hausdorff and second-countable. Hausdorff means every single point is a closed set and second-countable means there is a countable collection of open sets that generates all open sets by union. For example, a sphere is a manifold.

Optimization on manifolds can be thought of as unconstrained optimization on a constrained space. The ideas of algorithms for unconstrained optimization on an Euclidean space can be used for optimization on a manifold if many definitions are reconsidered. This reconsideration is crucial because the ideas are not extended simply from Euclidean space. For instance, addition and subtraction of two points on Euclidean space exist but do not exist for two points on manifold in general. In order to extend well-known optimization methods in Euclidean space to manifold, e.g., steepest descent, Newton method, trust regions and quasi-Newton, we must give specific generalizations of certain Euclidean definitions. We briefly discuss them in the following sections. The presentation follows [AMS08].

1.2.2 Tangent Space

In order to apply optimization algorithms based on line search, we must consider a direction on a manifold. Consider a smooth mapping, $\gamma : \mathbb{R} \to \mathcal{M}$, that satisfies $\gamma(0) = x$. Let us attempt to define the direction at x along γ. Similar to Euclidean space, the first idea that comes to mind is

$$\gamma'(0) = \lim_{h \to 0} \frac{\gamma(h) - \gamma(0)}{h}.$$

Unfortunately, the subtraction of two points, $\gamma(\tau + h), \gamma(\tau)$, may not be defined on a manifold. A solution to this problem is to consider a smooth function $f : \mathcal{M} \to \mathbb{R}$. We then have

$$(f \circ \gamma)'(0) = \lim_{h \to 0} \frac{f(\gamma(h)) - f(\gamma(0))}{h},$$

which is well-defined. We can define $\dot{\gamma}(0)$ as a mapping from $\mathcal{F}_x(\mathcal{M})$, the set of all smooth real-valued functions on a neighborhood of x

$$\dot{\gamma}(0)f = (f \circ \gamma)'(0),$$

This mapping is the direction at x along γ and it is also called a tangent vector to the curve γ at $t = 0$. The formal definition of tangent vectors follows.

Definition 1.2.1 (tangent vector)**.** *A tangent vector ξ_x to a manifold $\mathcal{M}$ at a point x is a mapping from $\mathcal{F}_x(\mathcal{M})$ to $\mathbb{R}$ such that there exists a curve γ on $\mathcal{M}$ with $\gamma(0) = x$, satisfying*

$$\xi_x f = \dot{\gamma}(0) f = \frac{df(\gamma(t))}{dt}|_{t=0}$$

for all $f \in \mathcal{F}_x(\mathcal{M})$. The curve γ is said to realize the tangent vector ξ_x. The point x is called the root of the tangent vector ξ_x.

The set of all tangent vectors at x is called the tangent space of x and the union of all tangent spaces is called the tangent bundle of the manifold, $\mathrm{T}\,\mathcal{M}$. This is a very important definition. It generalizes the idea of direction in a Euclidean space. Furthermore, the tangent space is a linear space, i.e., closed under linear combination, with the same dimension as the manifold, in which many basic operations are well-defined. So, instead of working on manifold, we work on a tangent space. However, eventually, we need to go back to the manifold and a operation called retraction is needed. This is discussed later.

1.2.3 Riemannian Metric

The tangent space at a point on the manifold provides us with a vector space of tangent vectors that give an idea of direction on the manifold. A Riemannian metric allows us to compute angle and length of directions (tangent vectors). A Riemannian metric g is defined on each tangent space of x as an inner product $g_x : \mathrm{T}_x\,\mathcal{M} \times \mathrm{T}_x\,\mathcal{M} \to \mathbb{R}$. We use the following to denote Riemannian metric

$$g_x(\eta, \xi) = \langle \eta, \xi \rangle_x,$$

where $\eta, \xi \in \mathrm{T}_x\,\mathcal{M}$ and the x is dropped when context permits. A notation, flat $\flat$, is also used in the later sections. $\xi^\flat$ denotes a function from $\mathrm{T}_x\,\mathcal{M}$ to $\mathbb{R}$, which is $\xi^\flat \eta = g(\xi, \eta)$ for all $\eta \in \mathrm{T}_x\,\mathcal{M}$. A Riemannian manifold is the combination $(\mathcal{M}, g)$.

We can get the length of a curve on Riemannian manifold by the norm induced by this inner product.

$$d(x, y) = \inf_{\gamma} \{ \int_0^1 \|\dot{\gamma}(t)\|_{g_{\gamma(t)}} dt \},$$

where γ is a curve on $\mathcal{M}$ with $\gamma(0) = x$ and $\gamma(1) = y$.

Once distance is defined, we can define the idea of a neighborhood of a point, which is denoted by $\mathcal{B}_\delta(x)$ and defined

$$\mathcal{B}_\delta(x) = \{y \in \mathcal{M} : d(x, y) < \delta\}.$$

This idea of neighborhoods is used to define local minimizers for a function defined on a manifold. Given a function $f : \mathcal{M} \to \mathbb{R}$, a point x^* is a strict local minimizer if there exists some $\delta > 0$ such that

$$f(x) < f(y) \text{ for all } y \in \mathcal{B}_\delta(x).$$

1.2.4 Affine Connections, Geodesics, Exponential Mapping and Parallel Translation

Let $\gamma(t)$ be a curve on a Riemannian manifold. $\dot{\gamma}(t)$ is defined to show the direction along the curve. Using the Riemannian metric, the length of $\dot{\gamma}(t)$ shows the speed of change on the curve. However, an analogy to a second derivative is required to define acceleration and, thereby, to generalize the Euclidean notion of a straight line between two points as being the one with zero acceleration. Likewise, the 'straight line' on a Riemannian manifold, called a geodesic, is a curve $\gamma(t)$ that has zero acceleration. To define acceleration, we need a differentiation operator applicable to tangent vectors in different tangent spaces since $\dot{\gamma}(t)$ is a vector field along the curve. On Riemannian manifolds, differential operators are called affine connections.

Let $\mathcal{X}(\mathcal{M})$ be the set of all smooth vector fields on $\mathcal{M}$. An affine connection ∇ is a mapping from $\mathcal{X}(\mathcal{M}) \times \mathcal{X}(\mathcal{M})$ to $\mathcal{X}(\mathcal{M})$. This is a differential operator and is required to satisfy the following properties. For $a, b \in \mathbb{R}$, $\eta, \xi, \zeta \in \mathcal{X}(\mathcal{M})$ and for any $x \in \mathcal{M}$, $f, g \in \mathcal{F}_x(\mathcal{M})$:

1. $\nabla_{f\eta+g\zeta}\xi = f\nabla_\eta\xi + g\nabla_\zeta\xi$: $\mathcal{F}(\mathcal{M})$-linearity in the first argument;

2. $\nabla_\eta(a\xi + b\zeta) = a\nabla_\eta\xi + b\nabla_\eta\zeta$: $\mathbb{R}$-linearity in the second argument; and

3. $\nabla_\eta(f\xi) = (\eta f)\xi + f\nabla_\eta\xi$: Product rule/Leibniz's law.

At a point x on $\mathcal{M}$, the connection maps tangent vectors $(\eta, \xi) \in \mathrm{T}_x\,\mathcal{M} \times \mathrm{T}_x\,\mathcal{M}$ to a tangent vector $\nabla_\eta\xi \in \mathrm{T}_x\,\mathcal{M}$. The result $\nabla_\eta\xi$ is a covariant derivative of ξ with respect to η. For a general manifold $\mathcal{M}$, there is an infinite number of affine connections. For a Riemannian manifold $(\mathcal{M}, g)$, one of the affine connections, called Riemannian connection or Levi-Civita connection, uniquely satisfies the following two additional conditions:

1. $(\nabla_\eta \xi - \nabla_\xi \eta) f = \eta(\xi f) - \xi(\eta f)$: symmetry; and
2. $\zeta \langle \eta, \xi \rangle = \langle \nabla_\zeta \eta, \xi \rangle + \langle \eta, \nabla_\zeta \xi \rangle$ (compatibility with the Riemannian metric).

The geodesic defined by an affine connection is a curve that satisfies

$$\nabla_{\dot{\gamma}(t)} \dot{\gamma}(t) := \frac{D^2}{dt^2} \gamma(t) := \frac{D}{dt} \dot{\gamma}(t) = 0.$$

A consequence of the compatibility with the Riemannian metric is that when the affine connection is the Riemannian connection, one of the geodesics between two points on the manifold (there may be many) is also a minimal length curve. This is consistent with the straight line in Euclidean space. In this dissertation, only the Riemannian connection is considered.

Given a point $x \in \mathcal{M}$ and a tangent vector $\eta \in \mathrm{T}_x \mathcal{M}$, there is a unique geodesic $\gamma(t; x, \eta)$ satisfying $\gamma(0) = x$ and $\dot{\gamma}(0) = \xi$. In addition, this geodesic satisfies the homogeneity property, $\gamma(t; x, a\eta) = \gamma(at; x, \eta)$. The mapping is called the exponential mapping at x and is denoted

$$\mathrm{Exp}_x : \mathrm{T}_x \mathcal{M} \to \mathcal{M} : \eta \to \mathrm{Exp}_x \eta = \gamma(1; x, \eta)$$

Exponential mapping provides a method to relate a tangent vector of x to an element in the neighborhood of x. When performing optimization algorithm, e.g. line-search-based or trust-region-based, exponential mapping allows us to move in the tangent space and then map the resulting tangent vector back to the manifold in a neighborhood of x.

Given that a series of tangent spaces, each defined by a point in the sequence produced by the optimization algorithm, are encountered while solving a problem, we may also need to compare or combine multiple direction vectors. As a result, they must be placed in a common frame of reference, i.e., they must be "transported" to a common tangent space. Since affine connection gives the idea of differentiation for tangent vectors in different tangent spaces, we can use it to define a vector transport: parallel translation.

A vector field ξ on a curve γ satisfying $\frac{D}{dt} \xi = \nabla_{\dot{\gamma}} \xi = 0$ is called parallel. Given $a \in \mathbb{R}$ in the domain of γ and $\xi_{\gamma(a)} \in \mathrm{T}_{\gamma(a)} \mathcal{M}$, there is a unique parallel vector field ξ on γ such that $\xi(a) = \xi_{\gamma(a)}$. The operator $P_\gamma^{b \leftarrow a}$ sending $\xi(a)$ to $\xi(b)$ is called parallel translation along γ. In other words, we have

$$\frac{D}{dt} (P_\gamma^{t \leftarrow a}(a)) = 0.$$

When ∇ is the Riemannian connection, the parallel translation is an isometry.

1.2.5 Gradient and Hessian

The gradient of a function shows the steepest ascent direction and is very useful for optimization in a Euclidean space. Since the gradient is a direction on a manifold, it should be a tangent vector. For a function f defined on a Riemannian manifold $(\mathcal{M}, g)$, the Riemannian gradient of f at x, $\operatorname{grad} f$, is the unique tangent vector such that

$$\langle \operatorname{grad} f(x), \eta \rangle_x = \mathrm{D}\, f(x)[\eta], \forall \eta \in \mathrm{T}_x\, \mathcal{M}.$$

This definition is consistent with the Euclidean gradient since for a function h defined on $\mathbb{R}^n$, the directional directive along v is

$$\lim_{\epsilon \to 0} \frac{h(x + \epsilon v) - h(x)}{\epsilon} = \operatorname{grad} h(x)^T v = \langle \operatorname{grad} h(x), v \rangle_2.$$

The Hessian is required in second-order optimization algorithms, such as Newton's Method. The Hessian of Euclidean function is the second derivative of the objective function. It contains the information of differentiating the gradient along some direction. For h defined on $\mathbb{R}^n$, the gradient is $\operatorname{grad} h(x) = \{\partial_i h(x)\}$ and the Hessian is $\operatorname{Hess} h(x) = \{\partial_{ij} h(x)\}$. Considering the derivative of $\operatorname{grad} h(x)$ along direction v gives

$$\lim_{\epsilon \to 0} \frac{\operatorname{grad} h(x + \epsilon v) - \operatorname{grad} h(x)}{\epsilon} = \operatorname{Hess} h(x) v.$$

This idea is used to define the Riemannian Hessian.

The Riemannian Hessian of f at x is the linear mapping from $\mathrm{T}_x\, \mathcal{M}$ to $\mathrm{T}_x\, \mathcal{M}$ defined by

$$\operatorname{Hess} f(x)[\eta] = \nabla_\eta \operatorname{grad} f(x),$$

for all $\eta \in \mathrm{T}_x\, \mathcal{M}$. From the symmetric property of the Riemannian connection, we know the Hessian is a self-adjoint(symmetric) operator with respect to Riemannian metric, i.e.

$$\langle \operatorname{Hess} f(x)[\eta], \xi \rangle_x = \langle \eta, \operatorname{Hess} f(x)[\xi] \rangle_x,$$

for all $\eta, \xi \in \mathrm{T}_x\, \mathcal{M}$.

1.2.6 Retraction and Vector Transport

In general, we work on the tangent space, either performing linear search or building a local model, to find a reasonable tangent vector to define the next iterate on the manifold. Retraction provides a method to map the tangent vector to the next iterate.

Definition 1.2.2 (retraction). *A retraction on a manifold $\mathcal{M}$ is a smooth mapping R from the tangent bundle $\mathrm{T}\,\mathcal{M}$ onto $\mathcal{M}$ with the following properties. Let R_x denote the restriction of R to $\mathrm{T}_x\,\mathcal{M}$.*

1. *$R_x(0_x) = x$, where 0_x denotes the zero element of $\mathrm{T}_x\,\mathcal{M}$.*
2. *With the canonical identification $\mathrm{T}_{0_x}\,\mathrm{T}_x\,\mathcal{M} \simeq \mathrm{T}_x\,\mathcal{M}$, R_x satisfies*
$$\mathrm{D}\,R_x(0_x) = \mathrm{id}_{\mathrm{T}_x\,\mathcal{M}},$$
where $\mathrm{id}_{\mathrm{T}_x\,\mathcal{M}}$ denotes the identity mapping on $\mathrm{T}_x\,\mathcal{M}$.

The exponential mapping is a special retraction. When we perform line search along a direction in the tangent space and use exponential mapping to map back to the manifold, we are actually performing a line search along the geodesic defined by the tangent vector. This was the basic idea used initially to define line search on a set of equality constraints in $\mathbb{R}^n$ by Luenberger [Lue72, Lue73]. Retraction provides a critical alternative to the exponential mapping which can often be too expensive to define an efficient Riemannian optimization method.

As we have seen, parallel translation provides an idea of moving tangent vectors between tangent spaces. However, since it is based on the idea of the exponential mapping it is also often too expensive to use in a practical method. Vector transport is an alternative built upon retraction.

Definition 1.2.3 (vector transport). *Vector transport on a manifold $\mathcal{M}$ is a smooth mapping*
$$T\mathcal{M} \oplus T\mathcal{M} \to T\mathcal{M} : (\eta_x, \xi_x) \mapsto \mathcal{T}_{\eta_x}(\xi_x) \in T\mathcal{M}$$
satisfying the following properties for all $x \in \mathcal{M}$.

- *(Associated retraction) There exists a retraction R, called the retraction associated with $\mathcal{T}$, such that the following diagram commutes*

$$\begin{tikzcd}
(\eta_x, \xi_x) \arrow[r, "\mathcal{T}"] \arrow[d] & \mathcal{T}_{\eta_x}(\xi_x) \arrow[d, "\pi"] \\
\eta_x \arrow[r, "R"'] & \pi\left(\mathcal{T}_{\eta_x}(\xi_x)\right)
\end{tikzcd}$$

where $\pi\left(\mathcal{T}_{\eta_x}(\xi_x)\right)$ denotes the foot of the tangent vector $\mathcal{T}_{\eta_x}(\xi_x)$.

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

Vector transport is called isometric if it also satisfies

$$g_{R(\eta_x)}(\mathcal{T}_{\eta_x}\xi_x, \mathcal{T}_{\eta_x}\zeta_x) = g_x(\xi_x, \zeta_x). \tag{1.2.1}$$

Vector transport by differentiated retraction is a vector transport given by

$$\mathcal{T}_{\eta_x}\xi_x = DR(\eta_x)[\xi_x],$$

where R is a retraction. We use $\mathcal{T}_S$ and $\mathcal{T}_R$ to denote an isometric vector transport and a differentiated retraction of R respectively. Vector transport is a very important for quasi-Newton algorithms which make use of the information from previous iterations and approximate the action of the Hessian. This information resides in different tangent spaces and the operator approximating the action of the Hessian must be moved through a series of tangent spaces. Without the use of vector transport the use of quasi-Newton algorithms on a range of manifolds would not be efficient enough for practical use. The theoretical and practical aspects of the design and implementation of retraction and vector transport is a key aspect of the research in this dissertation.

1.2.7 Coordinate Expressions

Coordinate expressions provide an approach to represent concepts on a manifold by concepts in a vector space. When we analyze a problem on a manifold in this dissertation, we use a "hat" to denote a coordinate expression.

Let $(\mathcal{U}, \varphi)$ be a chart of a manifold $\mathcal{M}$ and $x \in \mathcal{U}$. $\hat{x} \in \mathbf{R}^d$, the coordinates expression of x, is defined by $\hat{x} = \varphi(x)$. E_i, the i-th coordinate vector field of $(\mathcal{U}, \varphi)$, is defined by

$$(E_i f)(x) := \partial_i(f \circ \varphi^{-1})(\varphi(x)) = D(f \circ \varphi^{-1})(\varphi(x))[e_i].$$

These coordinate vector fields are smooth and every vector field ξ on $\mathcal{U}$ has a decomposition

$$\xi = \sum_i (\xi\varphi_i)E_i.$$

Therefore, $(E_i)_x, i = 1, \dots, d$ is a basis of $\mathrm{T}_x \mathcal{M}$ and the coordinate expression $\hat{\xi}_x$ of ξ_x with this basis is $(\xi_x \varphi_1, \dots, \xi_x \varphi_d)$. Since E_is are smooth vector fields on $\mathcal{U}$, one can always use QR decomposition to form a smooth orthonormal vector fields on $\mathcal{U}$, i.e., $(E_1, \dots, E_d) = (\tilde{E}_1, \dots, \tilde{E}_d)R$, where $\tilde{E}_1, \dots, \tilde{E}_d$ are orthonormal vector fields and $R : \mathcal{U} \to \mathbb{R}^{d \times d}$ is a smooth function and $R(x)$ is an upper triangle matrix for all $x \in \mathcal{U}$ [DE99]. Thus, the coordinate expression of ξ_x with the orthonormal basis can be obtained.

The coordinate expression of the metric at x is $(G_x)_{ij} = \langle E_i, E_j \rangle_x$ satisfying $g_x(\eta_x, \xi_x) = \hat{\eta}_x^T G_x \hat{\xi}_x$. Since a tangent space can be represented by $\mathbb{R}^d$, a linear operator $\mathcal{B}$ on a tangent space and a vector transport $\mathcal{T}$ admit matrix expressions $\hat{\mathcal{B}}$ and $\hat{\mathcal{T}}$ that are called coordinate expressions. Without loss of generality, one can always choose the orthonormal vector fields $\tilde{E}_1, \dots, \tilde{E}_d$. Therefore, the matrix expression of G_x is a identity and $\|\eta_x\| = \sqrt{g_x(\eta_x, \eta_x)} = \sqrt{\hat{\eta}_x^T \hat{\eta}_x} = \|\hat{\eta}_x\|_2$, where $\| \cdot \|_2$ denotes the Euclidean norm.

1.3 Historical Context

The concept of optimizing a real-valued function on manifold dates back to the work of Luenberger [Lue72, Lue73] in the early 1970s, if not earlier. Luenberger mentions the idea of performing line search along geodesics when geodesics are computationally feasible, which is definitely not always true. In general, computing the geodesics is rarely worth the effort. In most optimization on manifolds, an approximation of geodesic is enough to guarantee the desired convergence properties. What is more, many classical mathematical definitions in Riemannian geometry, such as geodesic, Levi-Civita connections, parallel vector transport, can be replaced by approximations.

Only recently, about 2002, researchers started to recognize the importance for a wide class of approximations of geodesics when optimizing on manifolds. Before then research was mostly theoretical: the central research question was to exploit differential-geometric objects in order to formulate optimization strategies on abstract nonlinear manifolds. The first research paper to focus on optimization on manifolds was Gabay [Gab82] on minimizing a differentiable function over a differential manifold in 1982 but it received only 8 citations before the year 2000. The area of optimization on manifolds started to gain wider popularity in the 1990s, notably with the seminal works of Helmke and Moore [HM94] and Edelman et al. [EAS98]. ISI records a total of 596 citations for [HM94], including 52 citations over the last two years.

Currently, most work is on making optimization on manifolds more practical and flexible and optimization on manifolds has become a very active area of research. The recent book [AMS08] provides an introduction to the area, with an emphasis on the necessary background in differential geometry instrumental to algorithmic development, and on guiding the reader through the concrete calculations that turn an abstract geometric algorithm into a numerical implementation. In 2008, the dissertation of C. Baker developed a complete theory for a Riemannian trust region Newton family of methods, implemented them in a numerical library, and analyzed their performance [Bak08]. In 2011, Qi gave an approach to generalize BFGS to a Riemannian manifold and developed the convergence analysis in her dissertation [Qi11]. Her convergence analysis restricts the approach of BFGS on Riemannian manifold to only work for exponential mapping and parallel vector transport. A recent paper by Ring and Wirth [RW12] provided another approach for BFGS on a Riemannian manifold. Instead of working on finite dimensional Riemannian manifold-s, their approach addresses infinite dimensional Riemannian manifolds. The convergence analysis is for both finite and infinite dimensional Riemannian manifolds with the latter depending on a specific assumption [RW12, Corollary 13]. They do not require exponential mapping and parallel vector transport. However, differentiated retraction is required which leads typically to excessive computational requirements.

1.4 Research Overview and Dissertation Statement

While the RTR-Newton-CG algorithm, analyzed and implemented in a reliable library by C. Baker [Bak08], has quadratic convergence property and has been investigated in practical situations by others, it requires the Hessian or its action which is not always easy to compute efficiently. Thus, an area of concentration in this dissertation is generalizing a family of quasi-Newton algorithm based on a Riemannian secant condition to remove the explicit need for the Hessian. This has been done in both the line search and trust region settings.

The work in this dissertation improves in several ways the earlier work on a Riemannian BFGS line search method (RBFGS) [Qi11] and that of Ring and Wirth [RW12]. Qi's RBFGS requires exponential mapping and parallel vector transport whose practical computation may not be possible. So even though they are available, they are often not good choice. Ring and Wirth's approach requires the differentiated retraction that also may suffer from complexity problems.

This dissertation proposes a systematic Riemannian generalization and analysis of the Euclidean optimization methods known as the restricted Broyden family method based on appropriately chosen retraction and vector transport. This resulting Riemannian family of line search methods subsumes the earlier RBFGS work. As part of analyzing the convergence rate of the restricted Riemannian Broyden family methods, the well-known sufficient and necessary conditions of superlinear convergence, Dennis Moré conditions, are generalized to the Riemannian setting for not only problems of optimization but also finding a zero of a vector field.

The idea of quasi-Newton approximation of the Hessian or its action can also be used effectively in a trust region setting. The restricted Broyden family in both Euclidean and Riemannian contexts preserve the positive definiteness of Hessian approximation and concentrate on approximating the action of the Hessian in a particular direction. As a result, they are most effective as the basis for a line search algorithm. The Euclidean Broyden family contains an interesting member outside the restricted set of algorithms – the Symmetric Rank-1 (SR1) method update, which does not preserve positive definiteness, and has lost favor as a line search algorithm. However, the method's tendency to produce indefinite Hessian approximations is based on the fact that it approximates the Hessian as an operator on multiple directions not just a particular line search direction. This turns out to be very useful in constrained optimization and for the definition of a local model required in a trust region setting.

Byrd et al. [BKS96] proved $n+1$-step superlinear convergence for a trust region method based on the Euclidean SR1 update. Furthermore, the performance of trust region with SR1 update is competitive with BFGS in their experiments. This is significant in that it removes the need for the Hessian yet retains the use of a trust region and its constrained optimization of a local model that often yields more robust performance than a line search method. The second part of the research in this dissertation proposes the generalization to Riemannian manifolds and the complete theoretical and empirical analysis of convergence of a trust region SR1 approach.

While the two activities above make a significant contribution to Riemannian optimization, the superlinear convergence analysis of such quasi-Newton and trust region algorithms depends on a smooth objective function. For a partly smooth objective function in a Euclidean setting, even though a complete convergence analysis is not yet available, Lewis and Overton [LO13] provided details experiments and showed that quasi-Newton algorithms work well and have observed linear convergence. Burke, Lewis and Overton [BLO05] gave a robust gradient sampling algorithm for partly smooth function proved its convergence, but not its rate. Since there is increasing interest in finding optima of a partly smooth function on Riemannian manifold, the third part of the dissertation generalizes gradient sampling to a Riemannian manifold and empirically analyze its convergence. This includes comparisons to applying one of our Riemannian quasi-Newton methods to the same problems.

The framework of retraction and vector transport is required in many implementation of Riemannian optimization algorithms, in particular Riemannian quasi-Newton methods and Riemannian gradient sampling method. The rigorous definitions of retraction and vector transport can be found in [AMS08] and the implementation for some specific manifolds is also discussed therein. However, there still is lack of discussions about implementation for Riemannian optimization, such as efficient and effective implementations of tangent vectors, metrics, linear operators of a tangent space, vector transports and so on. The fourth part of the dissertation gives a general discussion of implementation for some manifolds that can be represented by a vector in $\mathbb{R}^n$, i.e., manifolds in $\mathbb{R}^n$, quotient manifolds with total manifold in $\mathbb{R}^n$ and product of manifolds of the former two kinds of manifolds. Detailed discussions of the implementation are included on four important manifolds: the Stiefel manifold, the sphere, the orthogonal group and the Grassmann manifold.

Experiments and applications are presented in the last part of the dissertation. To better understand the proposed algorithms, we systematically compare the performance of the proposed algorithms for some classic and well-studied problems. In addition, four applications are also used to show the advantages of the proposed algorithms. Among them four have smooth enough cost functions: Riemannian optimization for elastic shape distance analysis, a joint diagonalization problem for independent component analysis, nonlinear dimension reduction problem, and a synchronization of rotations problem. and the efficiency and effectiveness of our Riemannian quasi-Newton methods are shown in the respective chapters. In secant-based nonlinear dimension reduction problem, two cost functions are proposed to solve a same problem – one is smooth and the other is partly smooth. The consequences of choosing either cost function is investigated empirically. For the smooth cost function, our Riemannian quasi-Newton methods are showed to outperform the proposed Riemannian algorithm in the original paper [BK05]. The Riemannian gradient sampling algorithm also shows encouraging performance for the partly smooth cost function.

CHAPTER 2

QUASI-NEWTON PREPARATION: SECANT CONDITIONS

2.1 Secant Condition on a Euclidean Space

Newton's method is locally quadratically convergent to any nondegenerate stationary points. In other words, if initial iterate x_0 is close enough to some stationary point x^*, then the method

$$x_{k+1} = x_k - (\operatorname{Hess} h(x))^{-1} \operatorname{grad} h(x) = x_k + d_k$$

produces a sequence such that

$$\lim_{k \to \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^2} < \infty.$$

For a minimization problem, the direction vector, d_k, is controlled to be a descent direction possibly with a scaling, i.e., $\alpha_k d_k$, to yield, in practice quadratic convergence to a local minimizer. These two basic ideas, descent and local superlinear convergence, can be used as the motivating properties to derive quasi-Newton methods.

Local superlinear convergence results from approximating the effect of Hessian in a particular direction. By Taylor's Theorem, we have

$$\operatorname{grad} h(x_{k+1}) = \operatorname{grad} h(x_k) + \operatorname{Hess} h(x_k)(x_{k+1} - x_k) + O(\|x_{k+1} - x_k\|^2).$$

Ignoring the high order term, gives

$$\operatorname{grad} h(x_{k+1}) - \operatorname{grad} h(x_k) \approx \operatorname{Hess} h(x_k)(x_{k+1} - x_k).$$

Using the equation as inspiration, we let the Hessian approximation B_{k+1} satisfy

$$\operatorname{grad} h(x_{k+1}) - \operatorname{grad} h(x_k) = B_{k+1}(x_{k+1} - x_k).$$

This equation is the Euclidean secant condition. Clearly, there are many B_{k+1} in the set of matrices, $\mathcal{S}$, that satisfy the secant condition and not all $B_{k+1} \in \mathcal{S}$ result in acceptable convergence. The secant condition only restricts the action of B_{k+1} in one direction and its action on the other directions is free. However, these other directions are present when determining the next direction vector $d_{k+1} = -B_{k+1}^{-1} \operatorname{grad} h(x_k)$ for a line search algorithm or when minimizing a local quadratic model defined by B_{k+1}. Therefore the rate of convergence is not guaranteed to be superlinear, in general, and the secant condition is not sufficient. Other conditions must be imposed and different conditions yield different quasi-Newton methods. In the following, we summarize the derivation of the well-known SR1 and the restricted Broyden family, see for example [NW06].

Let $y_k = \operatorname{grad} h(x_{k+1}) - \operatorname{grad} h(x_k)$ and $s_k = x_{k+1} - x_k$. The simplest symmetric condition yields the SR1 method (symmetric rank-1 update method). Given B_k, we update it to B_{k+1} by guaranteeing symmetry, using a rank-1 update, and satisfying the secant condition. These conditions define the unique update

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}.$$

This method is not easy to use in a line search idea because the direction $\eta_{k+1} = B_{k+1}^{-1} \operatorname{grad} h(x_{k+1})$ is not guaranteed to be a descent direction. It is more natural to combine SR1 with a trust region to guarantee descent.

Another well-known approach is to impose the following conditions

$$\begin{gathered}\min_B \|B - B_k\|_{W_B} \\ \text{s.t. } B = B^T, B s_k = y_k,\end{gathered}$$

where W_B is any matrix satisfying $W_B y_k = s_k$ and $\|A\|_{W_B} = \|W_B^{1/2} A W_B^{1/2}\|_F$. The next Hessian approximation, B_{k+1}, is the closest matrix to B_k that satisfies the secant condition. B_{k+1} is therefore obtained by making use of new information (secant condition) while preserving previous secant conditions as much as possible (minimum change to B_k). This idea leads to the Davidon-Fletcher-Powell (DFP) update

$$B_{k+1} = (I - \frac{y_k s_k^T}{y_k^T s_k}) B_k (I - \frac{s_k y_k^T}{y_k^T s_k}) + \frac{y_k y_k^T}{y_k^T s_k}.$$

It can be shown that if $s_k^T y_k > 0$ (Euclidean curvature condition) then the positive definiteness of B_k is preserved in B_{k+1} and $\eta_{k+1}^T \operatorname{grad} h(x_{k+1}) = - \operatorname{grad} h(x_{k+1}) B_{k+1} \operatorname{grad} h(x_{k+1}) < 0$, guaranteeing a descent direction.

Similarly, instead of preserving previous Hessian information as much as possible, we could preserve the inverse of Hessian approximation $H_k = B_k^{-1}$,

$$\begin{aligned} &\min_H \|H - H_k\|_{W_H} \\ \text{s.t. } &H = H^T, H y_k = s_k, \end{aligned}$$

where W_H is any matrix satisfying $W_H s_k = y_k$ and $\|A\|_{W_H} = \|W_H^{1/2} A W_H^{1/2}\|_F$. This leads to the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}.$$

As before if $s_k^T y_k > 0$ then B_{k+1} is positive definite and the direction $\eta_{k+1} = B_{k+1} \operatorname{grad} h(x_{k+1})$ is a descent direction. Since DFP and BFGS produce a sequence of positive definite Hessian approximations, they can be used in a line search algorithm, especially BFGS, since the action of the Hessian approximation and the Hessian are, ultimately, very close along the search direction.

The restricted Broyden family is defined by taking a convex combination of BFGS and DFP updates. It shares the property of preserving the positive definiteness of Hessian approximation. SR1 is in the Broyden family, i.e., a linear but not convex combination of BFGS and DFP updates. It does not, in general, preserve positive definiteness, but it provides better overall Hessian approximation than members of the restricted Broyden family [BKS96], [KBS93]. As a result it is more effective to combine SR1 with a trust region that makes use of all of the directional information of Hessian approximation. In particular, Byrd et al. proved $n + 1$-step superlinear convergence analysis of the resulting method [BKS96].

2.2 A Secant Condition on a Riemannian Manifold

Qi [Qi11] proposed a generalization of BFGS to RBFGS in her dissertation. She did not, however, consider its derivation from the point of view of rigorously defining a Riemannian form of the secant condition and then deducing the updates to the approximation of the Hessian on the tangent space of the current iterate. There are several possible generalizations of the Euclidean secant condition to a Riemannian manifold. In this section, we consider first the most natural generalization that easily lends itself to parallel translation and the exponential mapping and produces

Qi's RBFGS form of update. In Section 2.3, we consider other generalizations that address some of shortcomings of the natural approach.

Consider a function $f(x)$ defined on a manifold $\mathcal{M}$. Notice the Euclidean secant condition is from Taylor's Theorem for the gradient. Similarly, we have Taylor's Theorem for a vector field on a manifold rather than for $f(x)$ [AMS08, Lemma 7.4.7].

Theorem 2.2.1 (Taylor's Theorem)**.** *Let $x \in \mathcal{M}$, let $\mathcal{V}$ be a normal neighborhood of x, and let ζ be a C^1 tangent vector field on $\mathcal{M}$. Then, for all $y \in \mathcal{V}$,*

$$P_\gamma^{0\leftarrow 1}\zeta_y = \zeta_x + \nabla_\xi \zeta + \int_0^1 (P_\gamma^{0\leftarrow \tau} \nabla \gamma'(\tau)\zeta - \nabla_\xi \zeta) d\tau,$$

where γ is the unique minimizing geodesic satisfying $\gamma(0) = x$ and $\gamma(1) = y$, and $\xi = \operatorname{Exp}_x^{-1} y = \gamma'(0)$.

Applying Taylor's Theorem, we have

$$\begin{aligned} P_{\gamma_k}^{0\leftarrow 1} \operatorname{grad} f(x_{k+1}) &= \operatorname{grad} f(x_k) + \nabla_\xi \operatorname{grad} f(x_k) \\ &\quad + \int_0^1 (P_{\gamma_k}^{0\leftarrow \tau} \nabla \gamma_k'(\tau) \operatorname{grad} f(x_k) - \nabla_\xi \operatorname{grad} f(x_k)) d\tau, \end{aligned}$$

where γ_k is the unique minimizing geodesic satisfying $\gamma_k(0) = x_k$ and $\gamma_k(1) = x_{k+1}$, and $\xi = \operatorname{Exp}_{x_k}^{-1} x_{k+1} = \gamma_k'(0)$. Then by ignoring the integral remainder term, we have

$$P_{\gamma_k}^{0\leftarrow 1} \operatorname{grad} f(x_{k+1}) - \operatorname{grad} f(x_k) \approx \nabla_\xi \operatorname{grad} f(x_k) = \operatorname{Hess} f(x_k) \operatorname{Exp}_{x_k}^{-1} x_{k+1}.$$

This is very similar to the Euclidean secant condition. However, the above is defined on $\mathrm{T}_{x_k} \mathcal{M}$, the desired Hessian approximation $\mathcal{B}_{k+1}$ must be an operator on $\mathrm{T}_{x_{k+1}} \mathcal{M}$. Applying parallel translation, yields a Riemannian secant condition

$$\operatorname{grad} f(x_{k+1}) - P_{\gamma_k}^{1\leftarrow 0} \operatorname{grad} f(x_k) = \mathcal{B}_{k+1}(P_{\gamma_k}^{1\leftarrow 0} \operatorname{Exp}_{x_k}^{-1} x_{k+1}). \tag{2.2.1}$$

In the following, y_k denotes $\operatorname{grad} f(x_{k+1}) - P_{\gamma_k}^{1\leftarrow 0} \operatorname{grad} f(x_k)$ and s_k denotes $P_{\gamma_k}^{1\leftarrow 0} \operatorname{Exp}_{x_k} x_{k+1}$. Using this form of a Riemannian secant condition, we can generalize the SR1 update to a Riemannian manifold. Instead of symmetry, we require $\mathcal{B}_{k+1}$ to be self-adjoint with respect to the Riemannian metric. We have

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

where $\tilde{\mathcal{B}}_k = P_{\gamma_k}^{1\leftarrow 0}\mathcal{B}_k P_{\gamma_k}^{0\leftarrow 1}$.

Similarly, the conditions that result in DFP and BFGS generalize to a Riemannian manifold as

$$\begin{aligned}\text{DFP: } & \min_{\mathcal{B}} \|\mathcal{B} - \tilde{\mathcal{B}}_k\|_{W_{\mathcal{B}}} \\ & \text{s.t. } \mathcal{B} = \mathcal{B}^*, \mathcal{B}s_k = y_k. \\ \text{BFGS: } & \min_{\mathcal{H}} \|\mathcal{H} - \tilde{\mathcal{H}}_k\|_{W_{\mathcal{H}}} \\ & \text{s.t. } \mathcal{H} = \mathcal{H}^*, \mathcal{H}y_k = s_k.\end{aligned}$$

where $\mathcal{A}^*$ denotes the adjoint operator of $\mathcal{A}$ and $\|\mathcal{A}\|_W = \|\hat{W}^{1/2}G^{1/2}\hat{A}G^{-1/2}W^{1/2}\|_F$, G is the matrix expression of the metric and hat denotes matrix expression for the operators A and W. We then have the DFP and BFGS update for manifold

$$\begin{aligned}\text{DFP: } \mathcal{B}_{k+1} &= (\mathrm{id} - \frac{y_k s_k^\flat}{y_k^\flat s_k})\tilde{\mathcal{B}}_k(\mathrm{id} - \frac{s_k y_k^\flat}{y_k^\flat s_k}) + \frac{y_k y_k^\flat}{y_k^\flat s_k} \\ \text{BFGS: } \mathcal{B}_{k+1} &= \tilde{\mathcal{B}}_k - \frac{\tilde{\mathcal{B}}_k s_k (\tilde{\mathcal{B}}_k s_k)^\flat}{s_k^\flat \tilde{\mathcal{B}}_k s_k} + \frac{y_k y_k^\flat}{y_k^\flat s_k}.\end{aligned}$$

As in the Euclidean case, we define the Broyden family on a Riemannian manifold by taking a combination of the Riemannian DFP and Riemannian BFGS operators defined by a $\phi_k \in \mathbb{R}$. The restricted Broyden family is defined by taking a convex combination with $0 \le \phi_k \le 1$. This can be expressed equivalently as the combination of DFP and BFGS updates below

$$\mathcal{B}_{k+1} = \tilde{\mathcal{B}}_k - \frac{\tilde{\mathcal{B}}_k s_k (\tilde{\mathcal{B}}_k^* s_k)^\flat}{(\tilde{\mathcal{B}}_k^* s_k)^\flat s_k} + \frac{y_k y_k^\flat}{y_k^\flat s_k} + \phi_k g(s_k, \tilde{\mathcal{B}}_k s_k) v_k v_k^\flat,$$

where

$$v_k = \frac{y_k}{g(y_k, s_k)} - \frac{\tilde{\mathcal{B}}_k s_k}{g(s_k, \tilde{\mathcal{B}}_k s_k)}.$$

The Riemannian SR1 derived above is a member of the Broyden family but is not the restricted Broyden family.

All of the quasi-Newton update formulas above are from the Riemannian secant condition (2.2.1). The condition (2.2.1) explicitly uses the exponential mapping and parallel translation. This is not required. Alternate forms of the secant condition that subsume (2.2.1) can be derived by using retraction and vector transport. This is discussed in the next section.

2.3 Retraction, Vector Transport and a Secant Condition

Retraction and vector transport are critical to the success of Riemannian optimization algorithms such as Riemannian quasi-Newton methods. Retraction is used to get the next iterate and vector transport is use to compare tangent vectors in different tangent space and to transport operators on one tangent space to another tangent space, e.g. $\tilde{\mathcal{B}}_k = P_{\gamma_k}^{1\leftarrow 0}\mathcal{B}_k P_{\gamma_k}^{0\leftarrow 1}$. Secant condition (2.2.1), due to its origins in Taylor's Theorem, is required to use the exponential mapping and parallel translation. We now consider approximations of the exponential mapping and parallel translation that induce a secant condition while providing sufficiently fast convergence in the resulting algorithms.

For our purposes in optimization, the vector transport used in the secant conditions is required to be an isometry. Experiments by Qi [Qi11] indicate that non-isometric vector transport can be used but it is not provably convergent in general, at least thus far.

The Hessian is always a self-adjoint operator. In theory, as we show in Chapter 5, the Hessian approximations are not required to be self-adjoint for superlinear convergence of a Riemannian optimization algorithm. For quasi-Newton methods, however, and in particular the updates discussed above, the self-adjoint requirement is imposed and it makes the portions of the analysis contained in this work tractable. Therefore, we assume $\tilde{\mathcal{B}}_k$ is self-adjoint and require $\mathcal{B}_{k+1}$ to be as well. From the formula of transporting an operator

$$\tilde{\mathcal{B}}_k = \mathcal{T}_{\eta_k} \circ \mathcal{B}_k \circ \mathcal{T}_{\eta_k}^{-1},$$

we know an isometric vector transport $\mathcal{T}$ guarantees that $\tilde{\mathcal{B}}_k$ is self-adjoint if $\mathcal{B}_k$ is self-adjoint.

Let $\mathcal{T}_S$ denote the isometric vector transport. The SR1 update is given by

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

where $\tilde{\mathcal{B}}_k = \mathcal{T}_{S_\eta}\mathcal{B}_k\mathcal{T}_{S_\eta}^{-1}$.

The first version of the BFGS method on a Riemannian manifold was given by Qi [Qi11] and, as noted above, is based on the exponential mapping and parallel translation (an isometry) as is Qi's convergence analysis.

Ring and Wirth used retraction and vector transport to define their Riemannian BFGS [RW12]. Although not explicitly derived in this manner, their update follows from an alternative secant condition. For a finite dimensional Riemannian manifold, instead of applying the Riemannian Taylor's Theorem, they considered the composition of function and retraction $f \circ R_x(\eta_x)$. Since the tangent space is a vector space, they apply the Euclidean Taylor's Theorem to the composition on the tangent space and derive a secant condition. In their infinite dimensional Riemannian manifold work, they use a secant condition of the same form but do not explicitly relate it to a generalization of Taylor's Theorem. The drawback of this approach is that two vector transports are required: one isometry and one derived from a differentiated retraction. As a result, the approach is quite limited in choice of vector transport and can produce very costly basic operations. A finite dimensional form of the secant condition of Ring and Wirth is

$$(\operatorname{grad} f(x_{k+1})^\flat \mathcal{T}_{R_{\xi_k}} - \operatorname{grad} f(x_k)^\flat)\mathcal{T}_{S_{\xi_k}}^{-1} = (\mathcal{B}_{k+1}\mathcal{T}_{S_{\xi_k}}\xi_k)^\flat.$$

This condition and the algorithm based upon it are compared to ours in later sections.

In this dissertation, we develop an idea that depends as little as possible on the information in differentiated retraction. An isometric vector transport is required but interestingly is not sufficient to preserve the positive definiteness of the Hessian approximations. For the Euclidean restricted Broyden family, the positive definite property is guaranteed by the curvature condition $s_k^T y_k > 0$. The condition $s_k^T y_k > 0$ is, in turn, guaranteed by the second Wolfe condition that is typically imposed when choosing the step size in a Euclidean line search. However, the most natural way of generalizing the second Wolfe condition to a Riemannian manifold does not guarantee $g(s_k, y_k) > 0$ which is also necessary and sufficient for the positive definite Hessian approximation for Riemannian manifolds. This is discussed in detail in Chapter 4. We overcome this difficulty by imposing a novel condition called the 'locking condition'

$$\mathcal{T}_{S_\xi}\xi = \beta\mathcal{T}_{R_\xi}\xi, \quad \beta = \frac{\|\xi\|}{\|\mathcal{T}_{R_\xi}\xi\|},$$

where $\mathcal{T}_R$ is the associated differentiated retraction. In other words, we need only impose a relationship between the selected retraction and vector transport, and the associated transport defined by differentiation in a single direction. This facilitates the derivation of a potentially efficient algorithm and its rigorous convergence analysis.

CHAPTER 3

A RIEMANNIAN TRUST REGION WITH SYMMETRIC RANK-ONE UPDATE METHOD

3.1 Introduction

The trust region method is a well-known technique in optimization [CGT00] and it was extended to Riemannian manifolds in [ABG07] (or see [AMS08, Ch. 7]), and found applications, e.g., in [JBAS10, VV10, IAVD11, MMBS11, BA11]. Trust region methods construct a quadratic model m_k of the objective function f around the current iterate x_k and produce a candidate new iterate by (approximately) minimizing the model m_k within a region where it is "trusted". Depending on the discrepancy between f and m_k at the candidate new iterate, the size of the trust region is updated and the candidate new iterate is accepted or rejected.

For lack of efficient techniques to produce a second-order term in m_k that is inexact but nevertheless guarantees superlinear convergence, the Riemannian trust region (RTR) framework loses some of its appeal when the exact second-order term—the Hessian of f—is not available. This is in contrast with the Euclidean case, where several strategies exist to build an inexact second-order term that preserves superlinear convergence of the trust region method. Among these strategies, the symmetric rank-one (SR1) update is favored in view of its simplicity and because it preserves symmetry without unnecessarily enforcing positive definiteness; see, e.g., [NW06, §6.2] for a more detailed discussion. The $n + 1$ step q-superlinear rate of convergence of the SR1 trust region method was shown by Byrd et al. [BKS96] using a sophisticated analysis that builds on the results in [CGT91, KBS93].

In Chapter 3, motivated by the situation described above, we introduce a generalization of the classical (i.e., Euclidean) SR1 trust region method to the Riemannian setting (1.1.1). Besides making use of basic Riemannian geometric concepts (tangent space, Riemannian metric, gradient), the new method, called RTR-SR1, relies on the notions of retraction and vector transport introduced in [ADM02, AMS08]. A detailed global and local convergence analysis is given. A limited-memory version of RTR-SR1, referred to as LRTR-SR1, is also introduced. Numerical experiments show that the RTR-SR1 method displays the expected convergence properties. When the Hessian of f is not available, RTR-SR1 thus offers an attractive way of tackling (1.1.1) by a trust region approach. Moreover, even when the Hessian of f *is* available, making use of it can be expensive computationally, and the numerical experiments show that ignoring the Hessian information and resorting instead to the RTR-SR1 approach can be beneficial.

Another contribution of Chapter 3 with respect to [BKS96] is an extension of the analysis to allow for inexact solutions of the trust region subproblem—compare (3.3.3) with [BKS96, (2.4)]. This extension makes it possible to resort to inner iterations such as the Steihaug–Toint truncated CG method (see [AMS08, §7.3.2] for its Riemannian extension) while staying within the assumptions of the convergence analysis.

Chapter 3 is organized as follows. The RTR-SR1 method is stated and discussed in Section 3.2. The convergence analysis is carried out in Section 3.3. The limited-memory version is introduced in Section 3.4. Experiments illustrating the performance of RTR-SR1 are presented for several application problems in the associated chapters.

3.2 The Riemannian SR1 Trust Region Method

The proposed Riemannian SR1 trust region (RTR-SR1) method is described in Algorithm 1. Algorithm 1 can be viewed as a Riemannian version of the classical (Euclidean) SR1 trust region method (see, e.g., [NW06, Algorithm 6.2]). It can also be viewed as an SR1 version of the Riemannian trust region framework [AMS08, algorithm. 10 p. 142]. Therefore, several pieces of information given in [AMS08, Ch. 7] remain relevant for Algorithm 1.

Within the Riemannian trust region framework, the characterizing aspect of Algorithm 1 lies in the update mechanism for the Hessian approximation $\mathcal{B}_k$. The proposed update mechanism, based on formula (3.2.2) and on Step 6 of Algorithm 1, is a rather straightforward Riemannian generalization of the classical SR1 update

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}.$$

Significantly less straightforward is the Riemannian generalization of the superlinear convergence result in Section 3.3.4. (Observe that the local convergence result [AMS08, Theorem 7.4.11] does not apply here because the Hessian approximation condition [AMS08, (7.36)] is not guaranteed to hold.)

Algorithm 1 Riemannian trust region with symmetric rank-one update (RTR-SR1)

Input: Riemannian manifold $\mathcal{M}$ with Riemannian metric g; retraction R; isometric vector transport $\mathcal{T}_S$; differentiable real-valued objective function f on $\mathcal{M}$; initial iterate $x_0 \in \mathcal{M}$; initial Hessian approximation $\mathcal{B}_0$, symmetric with respect to g.

```
1:  Choose $\Delta_0 > 0$, $\nu \in (0,1)$, $c \in (0, 0.1)$, $\tau_1 \in (0,1)$ and $\tau_2 > 1$; Set $k \leftarrow 0$;
2:  Obtain $s_k \in \mathrm{T}_{x_k} \mathcal{M}$ by (approximately) solving
        $s_k = \arg \min_{s \in \mathrm{T}_{x_k} \mathcal{M}} m_k(s) = \arg \min_{s \in \mathrm{T}_{x_k} \mathcal{M}} f(x_k) + g(\operatorname{grad} f(x_k), s) + \frac{1}{2} g(s, \mathcal{B}_k s), \text{s.t. } \|s\| \leq \Delta_k; \quad (3.2.1)$
3:  Set $\rho_k \leftarrow \frac{f(x_k) - f(R_{x_k}(s_k))}{m_k(0) - m_k(s_k)}$;
4:  Let $y_k = \mathcal{T}_{S_{s_k}}^{-1} \operatorname{grad} f(R_{x_k}(s_k)) - \operatorname{grad} f(x_k)$; If $|g(s_k, y_k - \mathcal{B}_k s_k)| < \nu \|s_k\| \|y_k - \mathcal{B}_k s_k\|$, then
    $\tilde{\mathcal{B}}_{k+1} = \mathcal{B}_k$, otherwise define the linear operator $\tilde{\mathcal{B}}_{k+1} : \mathrm{T}_{x_k} \mathcal{M} \to \mathrm{T}_{x_k} \mathcal{M}$ by
        $\tilde{\mathcal{B}}_{k+1} = \mathcal{B}_k + \frac{(y_k - \mathcal{B}_k s_k)(y_k - \mathcal{B}_k s_k)^\flat}{g(s_k, y_k - \mathcal{B}_k s_k)}, \quad \text{(SR1)}; \quad (3.2.2)$
5:  if $\rho_k > c$ then
6:      $x_{k+1} \leftarrow R_{x_k}(s_k)$; $\mathcal{B}_{k+1} \leftarrow \mathcal{T}_{S_{s_k}} \circ \tilde{\mathcal{B}}_{k+1} \circ \mathcal{T}_{S_{s_k}}^{-1}$;
7:  else
8:      $x_{k+1} \leftarrow x_k$; $\mathcal{B}_{k+1} \leftarrow \tilde{\mathcal{B}}_{k+1}$;
9:  end if
10: if $\rho_k > \frac{3}{4}$ then
11:     if $\|s_k\| \geq 0.8 \Delta_k$ then
12:         $\Delta_{k+1} \leftarrow \tau_2 \Delta_k$;
13:     else
14:         $\Delta_{k+1} \leftarrow \Delta_k$;
15:     end if
16: else if $\rho_k < 0.1$ then
17:     $\Delta_{k+1} \leftarrow \tau_1 \Delta_k$;
18: else
19:     $\Delta_{k+1} \leftarrow \Delta_k$;
20: end if
21: $k \leftarrow k + 1$, goto 2 until convergence.
```

The Riemannian SR1 update uses tangent vectors at the current iterate to produce a new Hessian approximation at the next iterate, hence the need to perform a vector transport (see Step 6) from the current iterate to the next.

The symmetry requirement on $\mathcal{B}_0$ with respect to the Riemannian metric g means that

$$g(\mathcal{B}_0\xi_{x_0}, \eta_{x_0}) = g(\xi_{x_0}, \mathcal{B}_0\eta_{x_0})$$

for all $\xi_{x_0}, \eta_{x_0} \in \mathrm{T}_{x_0}\mathcal{M}$. It is readily seen from (3.2.2) and Step 6 of Algorithm 1 that $\mathcal{B}_k$ is symmetric for all k. Note however that $\mathcal{B}_k$ is, in general, not positive definite.

A possible stopping criterion for Algorithm 1 is $\|\operatorname{grad} f(x_k)\| < \epsilon$ for some specified $\epsilon > 0$.

In the spirit of [RW12, Remark 4], we point out that it is possible to formulate the SR1 update (3.2.2) in the new tangent space $T_{x_{k+1}}\mathcal{M}$; in the present case of SR1, the algorithm remains equivalent since the vector transport is isometric.

Otherwise, Algorithm 1 does not call for comments other than those made in [AMS08, Ch. 7]. In particular, we point out that the meaning of "approximately" in Step 2 of Algorithm 1 depends on the desired convergence results. It is shown in the convergence analysis (Section 3.3) that enforcing the Cauchy decrease (3.3.2) is enough to ensure global convergence to stationary points, but another condition such as (3.3.3) is needed to guarantee superlinear convergence. The truncated CG method, discussed in [AMS08, §7.3.2] in the Riemannian context, is an inner iteration for Step 2 that returns an s_k satisfying conditions (3.3.2) and (3.3.3).

3.3 Convergence Analysis of RTR-SR1

3.3.1 Notation and Standing Assumptions

Throughout the convergence analysis, unless otherwise specified, we let $\{x_k\}$, $\{\mathcal{B}_k\}$, $\{\tilde{\mathcal{B}}_k\}$, $\{s_k\}$, $\{y_k\}$, and $\{\Delta_k\}$ be infinite sequences generated by Algorithm 1, and we make use of the notation introduced in that algorithm. We let Ω denote the sublevel set of x_0, i.e.,

$$\Omega = \{x \in \mathcal{M} : f(x) \leq f(x_0)\}.$$

The global and local convergence analyses each make standing assumptions at the beginning of their respective sections. The numbered assumptions introduced below are not standing assumptions and will be invoked specifically whenever needed.

3.3.2 Global Convergence Analysis

In some results, we will assume for the retraction R that there exists $\mu > 0$ and $\delta_\mu > 0$ such that

$$\|\xi\| \geq \mu \operatorname{dist}(x, R_x(\xi)) \quad \text{for all } x \in \Omega, \text{ for all } \xi \in \mathrm{T}_x \mathcal{M}, \|\xi\| \leq \delta_\mu. \tag{3.3.1}$$

This corresponds to [AMS08, (7.25)] restricted to the sublevel set Ω. Such a condition is instrumental in the global convergence analysis of Riemannian trust region schemes. Note that, in view of [RW12, Lemma 6], condition (3.3.1) can be shown to hold globally under the condition that R has equicontinuous derivatives.

The next assumption corresponds to [BKS96, (A3)].

Assumption 3.3.1. *The sequence of linear operators $\{\mathcal{B}_k\}$ is bounded by a constant M such that $\|\mathcal{B}_k\| \leq M$ for all k.*

We will often require that the trust region subproblem (3.2.1) is solved accurately enough that, for some positive constants σ_1 and σ_2,

$$m_k(0) - m_k(s_k) \geq \sigma_1 \| \operatorname{grad} f(x_k)\| \min\{\Delta_k, \sigma_2 \frac{\| \operatorname{grad} f(x_k)\|}{\|\mathcal{B}_k\|}\}, \tag{3.3.2}$$

and that

$$\mathcal{B}_k s_k = -\operatorname{grad} f(x_k) + \delta_k \text{ with } \|\delta_k\| \leq \| \operatorname{grad} f(x_k)\|^{1+\theta}, \quad \text{whenever } \|s_k\| \leq 0.8\Delta_k, \tag{3.3.3}$$

where $\theta > 0$ is a constant. These conditions are generalizations of [BKS96, (2.3–4)]. Observe that, even if we restrict to the Euclidean case, condition (3.3.3) remains weaker than condition [BKS96, (2.4)]. The purpose of introducing δ_k in (3.3.3) is to encompass stopping criteria such as [AMS08, (7.10)] that do not require the computation of an exact solution of the trust region subproblem. We point out in particular that (3.3.2) and (3.3.3) hold if the approximate solution of the trust region subproblem (3.2.1) is obtained from the truncated CG method, described in [AMS08, §7.3.2] in the Riemannian context.

We can now state and prove the main global convergence results. Point (iii) generalizes [BKS96, Theorem 2.1] while points (i) and (ii) are based on [AMS08, §7.4.1].

Theorem 3.3.1 (convergence)**.** *(i) If $f \in C^2$ is bounded below on the sublevel set Ω, Assumption 3.3.1 holds, condition* (3.3.2) *holds, and* (3.3.1) *is satisfied then* $\lim_{k\to\infty} \operatorname{grad} f(x_k) = 0$. *(ii)*

If $f \in C^2$, $\mathcal{M}$ is compact, Assumption 3.3.1 holds, and (3.3.2) *holds then* $\lim_{k\to\infty} \operatorname{grad} f(x_k) = 0$, *$\{x_k\}$ has at least one limit point, and every limit point of $\{x_k\}$ is a stationary point of f. (iii) If $f \in C^2$, the sublevel set Ω is compact, f has a unique stationary point x^* in Ω, Assumption 3.3.1 holds, condition* (3.3.2) *holds, and* (3.3.1) *is satisfied then $\{x_k\}$ converges to x^*.*

Proof. (i) Observe that the proof of [AMS08, Theorem 7.4.4] still holds when condition [AMS08, (7.25)] is weakened to its restriction (3.3.1) to Ω. Indeed, since the trust region method is a descent iteration, it follows that all iterates are in Ω. The assumptions thus allow us to conclude, by [AMS08, Theorem 7.4.4], that $\lim_{k\to\infty} \operatorname{grad} f(x_k) = 0$. (ii) It follows from [AMS08, Proposition 7.4.5] and [AMS08, Corollary 7.4.6] that all the assumptions of [AMS08, Theorem 7.4.4] hold. Hence $\lim_{k\to\infty} \operatorname{grad} f(x_k) = 0$, and every limit point is thus a stationary point of f. Since $\mathcal{M}$ is compact, $\{x_k\}$ is guaranteed to have at least one limit point. (iii) Again by [AMS08, Theorem 7.4.4], we get that $\lim_{k\to\infty} \operatorname{grad} f(x_k) = 0$. Since $\{x_k\}$ belongs to the compact set Ω and cannot have limit points other than x^*, it follows that $\{x_k\}$ converges to x^*. □

3.3.3 More Notation and Standing Assumptions

For the purpose of conducting a local convergence analysis, we now assume that $\{x_k\}$ converges to a point x^*. Moreover, we assume throughout that $f \in C^2$.

We let $\mathcal{U}_{\text{trn}}$ be a *totally retractive neighborhood* of x^*, a concept inspired from the notion of totally normal neighborhood (see [dC92, §3.3]). By this, we mean that there is $\delta_{\text{trn}} > 0$ such that, for each $y \in \mathcal{U}_{\text{trn}}$, we have that $R_y(\mathbb{B}(0_y, \delta_{\text{trn}})) \supseteq \mathcal{U}_{\text{trn}}$ and $R_y(\cdot)$ is a diffeomorphism on $\mathbb{B}(0_y, \delta_{\text{trn}})$, where $\mathbb{B}(0_y, \delta_{\text{trn}})$ denotes the ball of radius δ_{trn} in $\mathrm{T}_y\,\mathcal{M}$ centered at the origin 0_y. The existence of a totally retractive neighborhood can be shown along the lines of [dC92, Theorem 3.3.7]. We assume without loss of generality that $\{x_k\} \subset \mathcal{U}_{\text{trn}}$. Whenever we consider an inverse retraction $R_x^{-1}y$, we implicitly assume that $x, y \in \mathcal{U}_{\text{trn}}$.

3.3.4 Local Convergence Analysis

The purpose of this section is to obtain a superlinear convergence result for Algorithm 1, stated in Theorem 3.3.2. The analysis can be viewed as a Riemannian generalization of the local analysis in [BKS96, §2]. As we proceed, we will point out the main hurdles that had to be overcome in the generalization. The analysis makes use of several preparation lemmas, independent of Algorithm 1, that are of potential interest in the broader context of Riemannian optimization. These preparation lemmas become trivial or well known in the Euclidean context.

The next assumption corresponds to a part of [BKS96, (A1)].

Assumption 3.3.2. *The point x^* is a nondegenerate local minimizer of f. In other words,* $\operatorname{grad} f(x^*) = 0$ *and* $\operatorname{Hess} f(x^*)$ *is positive definite.*

The next assumption generalizes the assumption, contained in [BKS96, (A1)], that the Hessian of f is Lipschitz continuous near x^*. (Recall that $\mathcal{T}_S$ is the vector transport invoked in Algorithm 1.) Note that the assumption holds if $f \in C^3$; see Lemma 3.3.4.

Assumption 3.3.3. *There exists a constant c_0 such that for all $x, y \in \mathcal{U}_{\text{trn}}$,*

$$\| \operatorname{Hess} f(y) - \mathcal{T}_{S_\eta} \operatorname{Hess} f(x) \mathcal{T}_{S_\eta}^{-1} \| \leq c_0 \operatorname{dist}(x, y),$$

where $\eta = R_x^{-1} y$.

The next assumption is introduced to handle the Riemannian case; in the classical Euclidean setting, Assumption 3.3.4 follows from Assumption 3.3.3. Assumption 3.3.4 is mild since it holds if $f \in C^3$, as shown in Lemma 3.3.4.

Assumption 3.3.4. *There exists a constant c_0 such that for all $x, y \in \mathcal{U}_{\text{trn}}$, all $\xi_x \in \mathrm{T}_x \mathcal{M}$ with $R_x(\xi_x) \in \mathcal{U}_{\text{trn}}$, and all $\xi_y \in \mathrm{T}_y \mathcal{M}$ with $R_y(\xi_y) \in \mathcal{U}_{\text{trn}}$, it holds that*

$$\| \operatorname{Hess} \hat{f}_y(\xi_y) - \mathcal{T}_{S_\eta} \operatorname{Hess} \hat{f}_x(\xi_x) \mathcal{T}_{S_\eta}^{-1} \| \leq c_0 (\|\xi_y\| + \|\xi_x\| + \|\eta\|),$$

where $\eta = R_x^{-1}(y)$, $\hat{f}_x = f \circ R_x$, and $\hat{f}_y = f \circ R_y$.

The next assumption corresponds to [BKS96, (A2)]. It implies that no updates of $\mathcal{B}_k$ are skipped. In the Euclidean case, Khalfan et al. [KBS93] show that this is usually the case in practice.

Assumption 3.3.5.

$$|g(s_k, y_k - \mathcal{B}_k s_k)| \geq \nu \|s_k\| \|y_k - \mathcal{B}_k s_k\|$$

The next assumption is introduced to handle the Riemannian case. It states that the iterates eventually *continuously stay* in the totally retractive neighborhood $\mathcal{U}_{\text{trn}}$ (the terminology is borrowed from [ATV13, Definition 2.8]). The assumption is needed, in particular, for Lemma 3.3.5.

Assumption 3.3.6. *There exists N such that, for all $k \geq N$ and all $t \in [0, 1]$, it holds that $R_{x_k}(ts_k) \in \mathcal{U}_{\mathrm{trn}}$.*

The next lemma is proved in [GQA12, Lemma 14.1].

Lemma 3.3.1. *Let $\mathcal{M}$ be a Riemannian manifold, let $\mathcal{U}$ be a compact coordinate neighborhood in $\mathcal{M}$, and let the hat denote coordinate expressions. Then there are $c_2 > c_1 > 0$ such that, for all $x, y \in \mathcal{U}$, we have*

$$c_1 \|\hat{x} - \hat{y}\|_2 \leq \operatorname{dist}(x, y) \leq c_2 \|\hat{x} - \hat{y}\|_2.$$

Lemma 3.3.2. *Let $\mathcal{M}$ be a Riemannian manifold endowed with a retraction R and let $\bar{x} \in \mathcal{M}$. Then there exist $a_0 > 0$, $a_1 > 0$, and $\delta_{a_0,a_1} > 0$ such that for all x in a sufficiently small neighborhood of $\bar{x}$ and all $\xi, \eta \in \mathrm{T}_x \mathcal{M}$ with $\|\xi\| \leq \delta_{a_0,a_1}$ and $\|\eta\| \leq \delta_{a_0,a_1}$, it holds that*

$$a_0 \|\xi - \eta\| \leq \operatorname{dist}(R_x(\eta), R_x(\xi)) \leq a_1 \|\xi - \eta\|.$$

Proof. Since R is smooth, we can choose a neighborhood small enough such that R satisfies the condition of [RW12, Lemma 6], and the result follows from that lemma. □

The following lemma follows from Lemma 3.3.2 by taking $\eta = 0$. We state it separately for convenience as we will frequently invoke it in the analysis.

Lemma 3.3.3. *Let $\mathcal{M}$ be a Riemannian manifold endowed with retraction R and let $\bar{x} \in \mathcal{M}$. Then there exist $a_0 > 0$, $a_1 > 0$, and $\delta_{a_0,a_1} > 0$ such that for all x in a sufficiently small neighborhood of $\bar{x}$ and all $\xi \in \mathrm{T}_x \mathcal{M}$ with $\|\xi\| \leq \delta_{a_0,a_1}$, it holds that*

$$a_0 \|\xi\| \leq \operatorname{dist}(x, R_x(\xi)) \leq a_1 \|\xi\|.$$

Lemma 3.3.4. *If $f \in C^3$, then Assumptions 3.3.3 and 3.3.4 hold.*

Proof. First, we prove that Assumption 3.3.3 holds. Define a function $h : \mathcal{M} \times \mathcal{M} \times \mathrm{T}\mathcal{M} \to \mathrm{T}\mathcal{M}, (x, y, \xi_y) \to \mathcal{T}_{S_\eta} \operatorname{Hess} f(x) \mathcal{T}_{S_\eta}^{-1} \xi_y$, where $\eta = R_x^{-1}(y)$. Since $f \in C^3$, we know that $h(x, y, \xi_y)$ is C^1. Therefore, there exists b_0 such that for all $x, y \in \mathcal{U}_{\mathrm{trn}}$, $\xi_y \in \mathrm{T}_y \mathcal{M}$, $\|\xi_y\| = 1$,

$$\begin{aligned}
\|h(y, y, \xi_y) - h(x, y, \xi_y)\| &\leq b_0 \operatorname{dist}(\{y, y, \xi_y\}, \{x, y, \xi_y\}) \\
&\leq b_1 \|\{\hat{y}, \hat{y}, \hat{\xi}_y\} - \{\hat{x}, \hat{y}, \hat{\xi}_y\}\|_2 \text{ (by Lemma 3.3.1)} \\
&= b_1 \|\hat{y} - \hat{x}\|_2 \\
&\leq b_2 \operatorname{dist}(y, x), \text{ (by Lemma 3.3.1)}
\end{aligned}$$

where b_0, b_1 and b_2 are some constants. So we have

$$\begin{aligned} b_2 \operatorname{dist}(y,x) &\geq \|h(y,y,\xi_y) - h(x,y,\xi_y)\| \\ &= \|(\operatorname{Hess} f(y) - \mathcal{T}_{S_\eta} \operatorname{Hess} f(x) \mathcal{T}_{S_\eta}^{-1})[\xi_y]\| \end{aligned}$$

Choose ξ_y, $\|\xi_y\| = 1$ such that

$$\|(\operatorname{Hess} f(y) - \mathcal{T}_{S_\eta} \operatorname{Hess} f(x) \mathcal{T}_{S_\eta}^{-1})[\xi_y]\| = \|(\operatorname{Hess} f(y) - \mathcal{T}_{S_\eta} \operatorname{Hess} f(x) \mathcal{T}_{S_\eta}^{-1})\|.$$

We obtain

$$\|(\operatorname{Hess} f(y) - \mathcal{T}_{S_\eta} \operatorname{Hess} f(x) \mathcal{T}_{S_\eta}^{-1})\| \leq b_2 \operatorname{dist}(y,x).$$

To prove Assumption 3.3.4, we redefine h as $h(y,x,\xi_x) = \mathcal{T}_{S_\eta} \operatorname{Hess} \hat{f}_x(\xi_x) \mathcal{T}_{S_\eta}^{-1}$. Based on the description of coordinate expressions in Section 1.2.7, we use orthonormal vector fields to obtain the coordinate expression of h, denoted by $\hat{h}$. Therefore, the manifold norm and the Euclidean norm of coordinate expressions are the same and we have

$$\|\operatorname{Hess} \hat{f}_y(\xi_y) - \mathcal{T}_{S_\eta} \operatorname{Hess} \hat{f}_x(\xi_x) \mathcal{T}_{S_\eta}^{-1}\| = \|\operatorname{Hess} \hat{f}_y(\hat{\xi}_y) - \hat{\mathcal{T}}_{S_\eta} \operatorname{Hess} \hat{f}_x(\hat{\xi}_x) \hat{\mathcal{T}}_{S_\eta}^{-1}\|_2. \tag{3.3.4}$$

Since $f \in C^3$, we know that $\hat{h}$ is also in C^1. Hence there exists a constant b_3 such that

$$\|\hat{h}(\hat{y},\hat{y},\hat{\xi}_y) - \hat{h}(\hat{y},\hat{x},\hat{\xi}_x)\|_2 \leq b_3 \|\{\hat{y},\hat{y},\hat{\xi}_y\} - \{\hat{y},\hat{x},\hat{\xi}_x\}\|_2.$$

Therefore

$$\begin{aligned} \|\operatorname{Hess} \hat{f}_y(\hat{\xi}_y) - \hat{\mathcal{T}}_{S_\eta} \operatorname{Hess} \hat{f}_x(\hat{\xi}_x) \hat{\mathcal{T}}_{S_\eta}^{-1}\|_2 &= \|\hat{h}(\hat{y},\hat{y},\hat{\xi}_y) - \hat{h}(\hat{y},\hat{x},\hat{\xi}_x)\|_2 \\ &\leq b_3 \|\{\hat{y},\hat{y},\hat{\xi}_y\} - \{\hat{y},\hat{x},\hat{\xi}_x\}\|_2 \\ &\leq b_4(\|\hat{y} - \hat{x}\|_2 + \|\hat{\xi}_y\|_2 + \|\hat{\xi}_x\|_2) \\ &\leq b_5(\operatorname{dist}(x,y) + \|\hat{\xi}_y\|_2 + \|\hat{\xi}_x\|_2) \text{ (by Lemma 3.3.1)} \\ &\leq b_6(\|\eta\| + \|\xi_y\| + \|\xi_x\|) \text{ (by Lemma 3.3.3)} \end{aligned}$$

This and (3.3.4) gives us Assumption 3.3.4. □

The next lemma generalizes [BKS96, Lemma 2.2]. The key difference with the Euclidean case is the following: in the Euclidean case, when s_k is accepted, we simply have $\|s_k\| = \|x_{k+1} - x_k\|$, while in the Riemannian generalization, we invoke Assumption 3.3.6 and Lemma 3.3.3 to deduce that

$\|s_k\| \leq \frac{1}{a_0}\,\mathrm{dist}(x_{k+1}, x_k)$. Note that Assumption 3.3.6 cannot be removed. To see this, consider for example the unit sphere with the exponential mapping, where we can have $x_k = x_{k+1}$ with $\|s_k\| = 2\pi$.

Lemma 3.3.5. *Suppose Assumption 3.3.6 holds. Then either*

$$\Delta_k \to 0 \tag{3.3.5}$$

or there exist $K > 0$ and $\Delta > 0$ such that for all $k > K$

$$\Delta_k = \Delta. \tag{3.3.6}$$

In either case $s_k \to 0$.

Proof. Let $\Delta = \liminf \Delta_k$ and suppose first that $\Delta > 0$. From line 11 of Algorithm 1, if Δ_k is increased, then $\|s_k\| \geq 0.8\Delta_k$ and $x_{k+1} = R_{x_k} s_k$, which implies by Lemma 3.3.3 and Assumption 3.3.6 that $\mathrm{dist}(x_k, x_{k+1}) \geq a_0 0.8\Delta_k$. The latter inequality cannot hold for infinitely many values of k since $x_k \to x^*$ and $\liminf \Delta_k > 0$. Hence, there exists $K \geq 0$ such that Δ_k is not increased for any $k \geq K$. Since $\Delta > 0$, this implies that $\Delta_k \geq \Delta$ for all $k \geq K$. In view of the trust region update mechanism in Algorithm 1 and since $\Delta = \liminf \Delta_k$, we also know that, for some $K_1 > K$, $\Delta_{K_1} < \frac{1}{\tau_1}\Delta$. If the trust region radius were to be decreased we would have $\Delta_{K_1+1} < \Delta$, which we have ruled out. Since neither increase nor decrease can occur, we must have $\Delta_k = \Delta$ for all $k \geq K_1$.

Suppose now that $\Delta = 0$. Since $x_k \to x^*$, for every $\epsilon > 0$ there exists $K_\epsilon \geq 0$ such that $\mathrm{dist}(x_{k+1}, x_k) < \epsilon$ for all $k \geq K_\epsilon$. Since $\liminf \Delta_k = 0$, there exists $j \geq K_\epsilon$ such that $\Delta_j < \epsilon$. But since Δ_k is increased only if $\Delta_k \leq \frac{1}{0.8}\|s_k\| \leq \frac{1}{0.8a_0}\,\mathrm{dist}(x_{k+1}, x_k) < \frac{\epsilon}{0.8a_0}$, and the increase factor is τ_2, we have that $\Delta_k < \frac{\tau_2\epsilon}{0.8a_0}$ for all $k \geq j$. Therefore (3.3.5) follows.

To show that $\|s_k\| \to 0$, note that if (3.3.5) is true, then clearly $\|s_k\| \to 0$. If (3.3.6) is true, then for all $k > K$, the step s_k is accepted and $\|s_k\| \leq \frac{1}{a_0}\,\mathrm{dist}(x_{k+1}, x_k)$ (by Lemma 3.3.3), hence $\|s_k\| \to 0$ since $\{x_k\}$ converges. □

Lemma 3.3.6. *Let $\mathcal{M}$ be a Riemannian manifold endowed with two vector transports $\mathcal{T}_1$ and $\mathcal{T}_2$, and let $\bar{x} \in \mathcal{M}$. Then there exist a constant a_4 and a neighborhood $\mathcal{U}$ of $\bar{x}$ such that for all $x, y \in \mathcal{U}$ and all $\xi \in \mathrm{T}_y\,\mathcal{M}$,*

$$\|\mathcal{T}_{1_\eta}^{-1}\xi - \mathcal{T}_{2_\eta}^{-1}\xi\| \leq a_4\|\xi\|\|\eta\|,$$

where $\eta = R_x^{-1}y$.

Proof. Let $T_1(\hat{x},\hat{\eta})$ and $T_2(\hat{x},\hat{\eta})$ denote the coordinate expression of $\mathcal{T}_{1_\eta}^{-1}$ and $\mathcal{T}_{2_\eta}^{-1}$, respectively. We have

$$\begin{aligned}\|\mathcal{T}_{1_\eta}^{-1}\xi - \mathcal{T}_{2_\eta}^{-1}\xi\| &\leq b_0\|(T_1(\hat{x},\hat{\eta}) - T_2(\hat{x},\hat{\eta}))\hat{\xi}\|_2 \\ &\leq b_0\|\hat{\xi}\|_2\|T_1(\hat{x},\hat{\eta}) - T_2(\hat{x},\hat{\eta})\|_2 \\ &\leq b_1\|\hat{\xi}\|_2\|\hat{\eta}\|_2 \text{ (since } T_1(\hat{x},0) = T_2(\hat{x},0) \text{ and both } T_1 \text{ and } T_2 \text{ are smooth)} \\ &\leq b_2\|\xi\|\|\eta\|\end{aligned}$$

for some constants b_0, b_1, and b_2. □

The next lemma is proved in [GQA12, Lemma 14.5].

Lemma 3.3.7. *Let F be a C^1 vector field on a Riemannian manifold $\mathcal{M}$ and let $\bar{x} \in \mathcal{M}$ be a nondegenerate zero of F. Then there exist a neighborhood $\mathcal{U}$ of $\bar{x}$ and $a_5, a_6 > 0$ such that for all $x \in \mathcal{U}$,*

$$a_5 \operatorname{dist}(x,\bar{x}) \leq \|F(x)\| \leq a_6 \operatorname{dist}(x,\bar{x}).$$

In the Euclidean case, the next lemma holds with $\tilde{a}_7 = 0$ and reduces to the Fundamental Theorem of Calculus.

Lemma 3.3.8. *Let F be a C^1 vector field on a Riemannian manifold $\mathcal{M}$, let R be a retraction on $\mathcal{M}$, and let $\bar{x} \in \mathcal{M}$. Then there exist a neighborhood $\mathcal{U}$ of $\bar{x}$ and a constant $\tilde{a}_7$ such that for all $x, y \in \mathcal{U}$,*

$$\|P_\gamma^{0\leftarrow 1}F(y) - F(x) - (\int_0^1 P_\gamma^{0\leftarrow t}\mathbb{D}F(\gamma(t))P_\gamma^{t\leftarrow 0}dt)\eta\| \leq \tilde{a}_7\|\eta\|^2,$$

where $\eta = R_x^{-1}(y)$ and P_γ is the parallel translation along the curve γ given by $\gamma(t) = R_x(t\eta)$.

Proof. Define $G : [0,1] \to \mathrm{T}_x\mathcal{M} : t \mapsto G(t) = P_\gamma^{0\leftarrow t}F(\gamma(t))$. Observe that $G(0) = F(x)$ and $G(1) = P_\gamma^{0\leftarrow 1}F(y)$. We have

$$\begin{aligned}G'(t) &= \frac{d}{d\epsilon}G(t+\epsilon)|_{\epsilon=0} \\ &= P_\gamma^{0\leftarrow t}\frac{d}{d\epsilon}P_\gamma^{t\leftarrow t+\epsilon}F(\gamma(t+\epsilon))|_{\epsilon=0} \\ &= P_\gamma^{0\leftarrow t}\mathbb{D}F(\gamma(t))[\frac{d}{d\epsilon}\gamma(t+\epsilon)]|_{\epsilon=0} \\ &= P_\gamma^{0\leftarrow t}\mathbb{D}F(\gamma(t))[\mathcal{T}_{R_{t\eta}}\eta],\end{aligned}$$

where we have used an expression of the covariant derivative $\mathbb{D}$ in terms of the parallel translation P (see, e.g., [Cha06, theorem I.2.1]), and where $\mathcal{T}_{R_{t\eta}}\eta = \frac{d}{dt}(R(t\eta))$. Since $G(1) - G(0) = \int_0^1 G'(t)dt$, we obtain

$$\begin{aligned}
&\|P_\gamma^{0\leftarrow 1} F(y) - F(x) - \int_0^1 P_\gamma^{0\leftarrow t} \mathbb{D}F(\gamma(t)) P_\gamma^{t\leftarrow 0} \eta dt\| \\
&= \| \int_0^1 P_\gamma^{0\leftarrow t} \mathbb{D}F(\gamma(t))(\mathcal{T}_{R_{t\eta}}\eta - P_\gamma^{t\leftarrow 0}\eta) dt\| \\
&\leq \int_0^1 \|P_\gamma^{0\leftarrow t} \mathbb{D}F(\gamma(t)) P_\gamma^{t\leftarrow 0}\| \|(P_\gamma^{0\leftarrow t} \mathcal{T}_{R_{t\eta}}\eta - \eta)\| dt \\
&\leq \int_0^1 \|P_\gamma^{0\leftarrow t} \mathbb{D}F(\gamma(t)) P_\gamma^{t\leftarrow 0}\| \|(P_\gamma^{0\leftarrow t} \mathcal{T}_{R_{t\eta}}\eta - \mathcal{T}_{R_{t\eta}}^{-1}\mathcal{T}_{R_{t\eta}}\eta)\| dt \\
&\leq b_0 \|\eta\|^2 \text{ (by Lemma 3.3.6)}
\end{aligned}$$

where b_0 is some constant. □

Lemma 3.3.9. *Suppose Assumptions 3.3.2 and 3.3.3 hold. Then there exist a neighborhood $\mathcal{U}$ and a constant a_7 such that for all x_1, $\tilde{x}_1$, x_2, and $\tilde{x}_2 \in \mathcal{U}$, we have*

$$|g(\mathcal{T}_{S_\zeta}\xi_1, y_2) - g(\mathcal{T}_{S_\zeta}y_1, \xi_2)| \leq a_7 \max\{\mathrm{dist}(x_1, x^*), \mathrm{dist}(x_2, x^*), \mathrm{dist}(\tilde{x}_1, x^*), \mathrm{dist}(\tilde{x}_2, x^*)\} \|\xi_1\| \|\xi_2\|,$$

where $\zeta = R_{x_1}^{-1}(x_2)$, $\xi_1 = R_{x_1}^{-1}(\tilde{x}_1)$, $\xi_2 = R_{x_2}^{-1}(\tilde{x}_2)$, $y_1 = \mathcal{T}_{S_{\xi_1}}^{-1} \operatorname{grad} f(\tilde{x}_1) - \operatorname{grad} f(x_1)$, and $y_2 = \mathcal{T}_{S_{\xi_2}}^{-1} \operatorname{grad} f(\tilde{x}_2) - \operatorname{grad} f(x_2)$.

Proof. Define $\bar{y}_1 = P_{\gamma_1}^{0\leftarrow 1} \operatorname{grad} f(\tilde{x}_1) - \operatorname{grad} f(x_1)$ and $\bar{y}_2 = P_{\gamma_2}^{0\leftarrow 1} \operatorname{grad} f(\tilde{x}_2) - \operatorname{grad} f(x_2)$, where P is the parallel transport, $\gamma_1(t) = R_{x_1}(t\xi_1)$, and $\gamma_2(t) = R_{x_2}(t\xi_2)$. From Lemma 3.3.8, we have

$$\|\bar{y}_1 - \bar{H}_1(x_1, \tilde{x}_1)\xi_1\| \leq b_0 \|\xi_1\|^2 \quad \text{and} \quad \|\bar{y}_2 - \bar{H}_2(x_2, \tilde{x}_2)\xi_2\| \leq b_0 \|\xi_2\|^2, \tag{3.3.7}$$

where $\bar{H}_1(x_1, \tilde{x}_1) = \int_0^1 P_{\gamma_1}^{0\leftarrow t} \operatorname{Hess} f(\gamma_1(t)) P_{\gamma_1}^{t\leftarrow 0} dt$, $\bar{H}_2(x_2, \tilde{x}_2) = \int_0^1 P_{\gamma_2}^{0\leftarrow t} \operatorname{Hess} f(\gamma_2(t)) P_{\gamma_2}^{t\leftarrow 0} dt$, and

b_0 is a constant. It follows that

$$\begin{aligned}
&|g(\mathcal{T}_{S_\zeta}\xi_1, y_2) - g(\mathcal{T}_{S_\zeta} y_1, \xi_2)| \\
\leq\ &|g(\mathcal{T}_{S_\zeta}\xi_1, \bar{y}_2) - g(\mathcal{T}_{S_\zeta}\bar{y}_1, \xi_2)| + |g(\mathcal{T}_{S_\zeta}\xi_1, y_2 - \bar{y}_2) - g(\mathcal{T}_{S_\zeta}(y_1 - \bar{y}_1), \xi_2)| \\
\leq\ &|g(\mathcal{T}_{S_\zeta}\xi_1, \bar{H}_2(x_2, \tilde{x}_2)\xi_2) - g(\mathcal{T}_{S_\zeta}\bar{H}_1(x_1, \tilde{x}_1)\xi_1, \xi_2)| + b_1(\|\xi_1\| + \|\xi_2\|)\|\xi_1\|\|\xi_2\| \text{ (by (3.3.7))} \\
&+ |g(\mathcal{T}_{S_\zeta}\xi_1, \mathcal{T}_{S_{\zeta_2}}^{-1} \operatorname{grad} f(\tilde{x}_2) - P_{\gamma_2}^{0\leftarrow 1} \operatorname{grad} f(\tilde{x}_2))| \\
&+ |g(\mathcal{T}_{S_\zeta}(\mathcal{T}_{S_{\zeta_1}}^{-1} \operatorname{grad} f(\tilde{x}_1) - P_{\gamma_1}^{0\leftarrow 1} \operatorname{grad} f(\tilde{x}_1)), \xi_2)| \\
\leq\ &|g(\mathcal{T}_{S_\zeta}\xi_1, \bar{H}_2(x_2, \tilde{x}_2)\xi_2) - g(\mathcal{T}_{S_\zeta}\bar{H}_1(x_1, \tilde{x}_1)\xi_1, \xi_2)| + b_1(\|\xi_1\| + \|\xi_2\|)\|\xi_1\|\|\xi_2\| \\
&+ b_2\|\xi_1\|\|\xi_2\|\|\operatorname{grad} f(\tilde{x}_2)\| + b_3\|\xi_1\|\|\xi_2\|\|\operatorname{grad} f(\tilde{x}_1)\|, \text{ (by Lemma 3.3.6)} \qquad (3.3.8)
\end{aligned}$$

where b_1, b_2 and b_3 are positive constants. Since average Hessian is self-adjoint, we have

$$|g(\mathcal{T}_{S_\zeta}\xi_1, \bar{H}_2(x_2, \tilde{x}_2)\xi_2) - g(\mathcal{T}_{S_\zeta}\bar{H}_1(x_1, \tilde{x}_1)\xi_1, \xi_2)| = |g(\bar{H}_2(x_2, \tilde{x}_2)\mathcal{T}_{S_\zeta}\xi_1, \xi_2) - g(\mathcal{T}_{S_\zeta}\bar{H}_1(x_1, \tilde{x}_1)\xi_1, \xi_2)| \qquad (3.3.9)$$

By Lemma 3.3.3 and 3.3.7, we have

$$\begin{aligned}
&b_2\|\xi_1\|\|\xi_2\|\|\operatorname{grad} f(\tilde{x}_2)\| + b_3\|\xi_1\|\|\xi_2\|\|\operatorname{grad} f(\tilde{x}_1)\| \\
\leq\ &b_4\|\xi_1\|\|\xi_2\|(\operatorname{dist}(x_1, \tilde{x}_1) + \operatorname{dist}(x_2, \tilde{x}_2) + \operatorname{dist}(\tilde{x}_2, x^*) + \operatorname{dist}(\tilde{x}_1, x^*)) \qquad (3.3.10)
\end{aligned}$$

where b_4 is a positive constant. Applying (3.3.9) and (3.3.10) to (3.3.8) and using the triangle inequality of distance, we have

$$\begin{aligned}
&|g(\mathcal{T}_{S_\zeta}\xi_1, y_2) - g(\mathcal{T}_{S_\zeta} y_1, \xi_2)| \\
\leq\ &b_5\|\xi_1\|\|\xi_2\| \max\{\operatorname{dist}(x_1, x^*), \operatorname{dist}(x_2, x^*), \operatorname{dist}(\tilde{x}_1, x^*), \operatorname{dist}(\tilde{x}_2, x^*)\} \text{ (by)} \\
&+ |g(\bar{H}_2(x_2, \tilde{x}_2)\mathcal{T}_{S_\zeta}\xi_1, \xi_2) - g(\mathcal{T}_{S_\zeta}\bar{H}_1(x_1, \tilde{x}_1)\xi_1, \xi_2)| \qquad (3.3.11)
\end{aligned}$$

where b_5 is a positive constant. Using coordinate expressions, $T(\hat{x}_1, \hat{x}_2)$ to denote $\mathcal{T}_\zeta$ and G_{x_2} to denote the matrix expression of the Riemannian metric at x_2, we have

$$\begin{aligned}
&|g(\bar{H}_2(x_2, \tilde{x}_2)\mathcal{T}_{S_\zeta}\xi_1, \xi_2) - g(\mathcal{T}_{S_\zeta}\bar{H}_1(x_1, \tilde{x}_1)\xi_1, \xi_2)| \\
=\ &|\hat{\xi}_1^T T(\hat{x}_1, \hat{x}_2)^T \hat{\bar{H}}_2(\hat{x}_2, \hat{\tilde{x}}_2)^T G_{x_2}\hat{\xi}_2 - \hat{\xi}_1^T \hat{\bar{H}}_1(\hat{x}_1, \hat{\tilde{x}}_1)^T T(\hat{x}_1, \hat{x}_2)^T G_{x_2}\hat{\xi}_2| \\
\leq\ &\|\hat{\xi}_1\|_2 \|T(\hat{x}_1, \hat{x}_2)^T \hat{\bar{H}}_2(\hat{x}_2, \hat{\tilde{x}}_2)^T - \hat{\bar{H}}_1(\hat{x}_1, \hat{\tilde{x}}_1)^T T(\hat{x}_1, \hat{x}_2)^T\|_2 \|G_{x_2}\|_2 \|\hat{\xi}_2\|_2. \qquad (3.3.12)
\end{aligned}$$

Define a function

$$J(\hat{x}_1, \hat{\hat{x}}_1, \hat{x}_2, \hat{\hat{x}}_2) = T(\hat{x}_1, \hat{x}_2)^T \hat{\hat{H}}_2(\hat{x}_2, \hat{\hat{x}}_2)^T - \hat{\hat{H}}_1(\hat{x}_1, \hat{\hat{x}}_1)^T T(\hat{x}_1, \hat{x}_2)^T.$$

We can see that when $(\hat{x}_1^T, \hat{\hat{x}}_1^T) = (\hat{x}_2^T, \hat{\hat{x}}_2^T)$, $J = 0$. Since, in view of Assumption 3.3.3, J is Lipschitz continuous, it follows that (3.3.12) becomes

$$\begin{aligned} |g(\bar{H}_2 \mathcal{T}_{S_\zeta} \xi_1, \xi_2) - g(\mathcal{T}_{S_\zeta} \bar{H}_1 \xi_1, \xi_2)| &\leq b_6 \|(\hat{x}_1^T, \hat{\hat{x}}_1^T) - (\hat{x}_2^T, \hat{\hat{x}}_2^T)\|_2 \|\hat{\xi}_1\|_2 \|\hat{\xi}_2\|_2 \\ &\leq b_7 \|\xi_1\| \|\xi_2\| \max\{\mathrm{dist}(x_1, x_2), \mathrm{dist}(\tilde{x}_1, \tilde{x}_2)\}, \end{aligned}$$

where b_6, b_7 are some constants. Combining this equation with (3.3.11), we obtain

$$|g(\mathcal{T}_{S_\zeta} \xi_1, y_2) - g(\mathcal{T}_{S_\zeta} y_1, \xi_2)| \leq b_8 \|\xi_1\| \|\xi_2\| \max\{\mathrm{dist}(x_1, x^*), \mathrm{dist}(x_2, x^*), \mathrm{dist}(\tilde{x}_1, x^*), \mathrm{dist}(\tilde{x}_2, x^*)\},$$

where b_8 is a constant. □

Lemma 3.3.10. *Let $\mathcal{M}$ be a Riemannian manifold endowed with a vector transport $\mathcal{T}$ with associated retraction R, and let $\bar{x} \in \mathcal{M}$. Then there is a neighborhood $\mathcal{U}$ of $\bar{x}$ and a_8 such that for all $x, y \in \mathcal{U}$,*

$$\begin{aligned} \| \mathrm{id} - \mathcal{T}_\xi^{-1} \mathcal{T}_\eta^{-1} \mathcal{T}_\zeta \| &\leq a_8 \max(\mathrm{dist}(x, \bar{x}), \mathrm{dist}(y, \bar{x})), \\ \| \mathrm{id} - \mathcal{T}_\zeta^{-1} \mathcal{T}_\eta \mathcal{T}_\xi \| &\leq a_8 \max(\mathrm{dist}(x, \bar{x}), \mathrm{dist}(y, \bar{x})), \end{aligned}$$

where $\xi = R_{\bar{x}}^{-1} x$, $\eta = R_x^{-1} y$, $\zeta = R_{\bar{x}}^{-1} y$, and $\| \cdot \|$ is an induced norm.

Proof. We choose coordinate expression such that the matrix expression of the Riemannian metric at $\bar{x}$ is the identity. Let $L(x, y)$ denote $\mathcal{T}_{R_x^{-1} y}$. We have

$$\| \mathrm{id} - \mathcal{T}_\xi^{-1} \mathcal{T}_\eta^{-1} \mathcal{T}_\zeta \| = \| I - L(\bar{x}, x)^{-1} L(x, y)^{-1} L(\bar{x}, y) \|.$$

Define a function $J(\bar{x}, \xi, \zeta) = I - L(\bar{x}, R_{\bar{x}}(\xi))^{-1} L(R_{\bar{x}}(\xi), R_{\bar{x}}(\zeta))^{-1} L(\bar{x}, R_{\bar{x}}(\zeta))$. Notice that J is a smooth function and $J(\bar{x}, 0_{\bar{x}}, 0_{\bar{x}}) = 0$. So

$$\begin{aligned} \|J(\bar{x}, \xi, \zeta)\| &= \|J(\bar{x}, \xi, \zeta) - J(\bar{x}, 0_{\bar{x}}, 0_{\bar{x}})\| \\ &= \|\hat{J}(\hat{\bar{x}}, \hat{\xi}, \hat{\zeta}) - \hat{J}(\hat{\bar{x}}, \hat{0}_{\bar{x}}, \hat{0}_{\bar{x}})\|_2 \\ &\leq b_0(\|\hat{\xi}\|_2 + \|\hat{\zeta}\|_2) \text{ (smoothness of } J) \\ &\leq b_1(\mathrm{dist}(x, \bar{x}) + \mathrm{dist}(y, \bar{x})) \text{ (by Lemma 3.3.3)} \\ &\leq b_2 \max(\mathrm{dist}(x, \bar{x}), \mathrm{dist}(y, \bar{x})), \end{aligned}$$

where b_0, b_1 and b_2 are some constants. So

$$\|\operatorname{id} - \mathcal{T}_\xi^{-1}\mathcal{T}_\eta^{-1}\mathcal{T}_\zeta\| \leq b_2 \max(\operatorname{dist}(x, \bar{x}), \operatorname{dist}(y, \bar{x})).$$

This concludes the first part of the proof. The second part of the result follows from a similar argument. □

The next lemma generalizes [CGT91, Lemma 1]. It is instrumental in the proof of Lemma 3.3.13 below. In the Euclidean setting, it is possible to give an expression for a_9 and a_{10} in terms of c of Assumption 3.3.3 and ν of Assumption 3.3.5. In the Riemannian setting, we could not obtain such an expression, in part because the constant b_2 that appears in the proof below is no longer zero. However, the existence of a_9 and a_{10} can still be shown, under the assumption that $\{x_k\}$ converges to x^*, and this is all we need in order to carry on with Lemma 3.3.13.

Lemma 3.3.11. *Suppose Assumptions 3.3.1, 3.3.2, 3.3.3, and 3.3.5 hold. Then*

$$y_j - \tilde{\mathcal{B}}_{j+1}s_j = 0 \tag{3.3.13}$$

for all j. Moreover, there exist constants a_9 and a_{10} such that

$$\|y_j - (\mathcal{B}_i)_j s_j\| \leq a_9 a_{10}^{i-j-2}\epsilon_{i,j}\|s_j\| \tag{3.3.14}$$

for all j, $i \geq j+1$, where $\epsilon_{i,j} = \max_{j \leq k \leq i} \operatorname{dist}(x_k, x^*)$ and

$$(\mathcal{B}_i)_j = \mathcal{T}_{S_{\zeta_{j,i}}}^{-1}\mathcal{B}_i\mathcal{T}_{S_{\zeta_{j,i}}}$$

with $\zeta_{j,i} = R_{x_j}^{-1}(x_i)$.

Proof. From (3.2.2), we have

$$\tilde{\mathcal{B}}_{j+1}s_j = (\mathcal{B}_j + \frac{(y_j - \mathcal{B}_j s_j)(y_j - \mathcal{B}_j s_j)^\flat}{g(s_j, y_j - \mathcal{B}_j s_j)})s_j = y_j.$$

This yields (3.3.13), as well as (3.3.14) with $i = j+1$. The proof of (3.3.14) for $i > j+1$ is by induction. We choose $k \geq j+1$ and assume that (3.3.14) holds for all $i = j+1, \ldots, k$. Let

$r_k = y_k - \mathcal{B}_k s_k$. We have

$$\begin{aligned}
&|g(r_k, \mathcal{T}_{S_{\zeta_{j,k}}} s_j)| = |g(y_k - \mathcal{B}_k s_k, \mathcal{T}_{S_{\zeta_{j,k}}} s_j)| \\
\leq\ & |g(y_k, \mathcal{T}_{S_{\zeta_{j,k}}} s_j) - g(s_k, \mathcal{T}_{S_{\zeta_{j,k}}} y_j)| + |g(s_k, \mathcal{T}_{S_{\zeta_{j,k}}} (y_j - (\mathcal{B}_k)_j s_j))| \\
& + |g(s_k, \mathcal{T}_{S_{\zeta_{j,k}}} ((\mathcal{B}_k)_j s_j)) - g(\mathcal{B}_k s_k, \mathcal{T}_{S_{\zeta_{j,k}}} s_j)| \\
\leq\ & |g(y_k, \mathcal{T}_{S_{\zeta_{j,k}}} s_j) - g(s_k, \mathcal{T}_{S_{\zeta_{j,k}}} y_j)| + \|\mathcal{T}_{S_{\zeta_{j,k}}} (y_j - (\mathcal{B}_k)_j s_j)\| \|s_k\| \\
& + |g(s_k, \mathcal{B}_k \mathcal{T}_{S_{\zeta_{j,k}}} s_j) - g(\mathcal{B}_k s_k, \mathcal{T}_{S_{\zeta_{j,k}}} s_j)| \\
\leq\ & |g(y_k, \mathcal{T}_{S_{\zeta_{j,k}}} s_j) - g(s_k, \mathcal{T}_{S_{\zeta_{j,k}}} y_j)| + b_0 a_9 a_{10}^{k-j-2} \epsilon_{k,j} \|s_j\| \|s_k\| \\
& (\mathcal{B}_k \text{ self-adjoint and induction assumption}) \\
\leq\ & b_0 a_9 a_{10}^{k-j-2} \epsilon_{k,j} \|s_j\| \|s_k\| + b_1 \epsilon_{k+1,j} \|s_k\| \|s_j\|, \text{ (by Lemma 3.3.9)}
\end{aligned}$$

where b_0 and b_1 are some constants. It follows that

$$\begin{aligned}
& \|y_j - (\mathcal{B}_{k+1})_j s_j\| \\
=\ & \|y_j - \mathcal{T}^{-1}_{S_{\zeta_{j,k+1}}} \mathcal{B}_{k+1} \mathcal{T}_{S_{\zeta_{j,k+1}}} s_j\| \\
=\ & \|y_j - \mathcal{T}^{-1}_{S_{\zeta_{j,k+1}}} \mathcal{T}_{S_{s_k}} \tilde{\mathcal{B}}_{k+1} \mathcal{T}^{-1}_{S_{s_k}} \mathcal{T}_{S_{\zeta_{j,k+1}}} s_j\| \\
\leq\ & \|y_j - \mathcal{T}^{-1}_{S_{\zeta_{j,k}}} \tilde{\mathcal{B}}_{k+1} \mathcal{T}_{S_{\zeta_{j,k}}} s_j\| + \|\mathcal{T}^{-1}_{S_{\zeta_{j,k}}} \tilde{\mathcal{B}}_{k+1} \mathcal{T}_{S_{\zeta_{j,k}}} s_j - \mathcal{T}^{-1}_{S_{\zeta_{j,k+1}}} \mathcal{T}_{S_{s_k}} \tilde{\mathcal{B}}_{k+1} \mathcal{T}^{-1}_{S_{s_k}} \mathcal{T}_{S_{\zeta_{j,k+1}}} s_j\| \\
\leq\ & \|y_j - ((\mathcal{B}_k)_j + \mathcal{T}^{-1}_{S_{\zeta_{j,k}}} \frac{(r_k)(r_k)^\flat}{g(s_k, r_k)} \mathcal{T}_{S_{\zeta_{j,k}}}) s_j\| + b_2 \epsilon_{k+1,j} \|s_j\| \\
& \text{(by Lemma 3.3.10, Assumption 3.3.1, and (3.2.2))} \\
\leq\ & \|y_j - (\mathcal{B}_k)_j s_j\| + b_3 \frac{|g(r_k, \mathcal{T}_{S_{\zeta_{j,k}}} s_j)|}{\|s_k\|} + b_2 \epsilon_{k+1,j} \|s_j\| \text{ (by Assumption 3.3.5)} \\
\leq\ & a_9 a_{10}^{k-j-2} \epsilon_{k,j} \|s_j\| + b_3 b_0 a_9 a_{10}^{k-j-2} \epsilon_{k,j} \|s_j\| + b_3 b_1 \epsilon_{k,j} \|s_j\| + b_2 \epsilon_{k+1,j} \|s_j\| \\
\leq\ & (a_9 a_{10}^{k-j-2} + b_3 b_0 a_9 a_{10}^{k-j-2} + b_3 b_1 + b_2) \epsilon_{k+1,j} \|s_j\|, \text{ (note that } \epsilon_{k,j} \leq \epsilon_{k+1,j})
\end{aligned}$$

where b_2, b_3 are some constant. Because b_0, b_1, b_2 and b_3 are independent of a_9 and a_{10}, we can choose a_9 and a_{10} large enough such that

$$(a_9 a_{10}^{k-j-2} + b_3 b_0 a_9 a_{10}^{k-j-2} + b_3 b_1 + b_2) \leq a_9 a_{10}^{k+1-j-2}.$$

for all j, $k \geq j+1$. Take for example, $a_9 > 1$ and $a_{10} > 1 + b_3 b_0 + b_3 b_1 + b_2$. Therefore

$$\|y_j - (\mathcal{B}_{k+1})_j s_j\| \leq a_9 a_{10}^{k+1-j-2} \epsilon_{k+1,j} \|s_j\|.$$

This concludes the argument by induction. □

Lemma 3.3.12. *If Assumption 3.3.3 holds then there exist a neighborhood $\mathcal{U}$ of x^* and a constant a_{11} such that for all $x_1, x_2 \in \mathcal{U}$, it holds that*

$$\|y - \mathcal{T}_{S_{\zeta_1}} \operatorname{Hess} f(x^*) \mathcal{T}_{S_{\zeta_1}}^{-1} s\| \leq a_{11} \|s\| \max\{\operatorname{dist}(x_1, x^*), \operatorname{dist}(x_2, x^*)\},$$

where $\zeta_1 = R_{x^*}^{-1}(x_1)$, $s = R_{x_1}^{-1}(x_2)$, $y = \mathcal{T}_{S_s}^{-1} \operatorname{grad} f(x_2) - \operatorname{grad} f(x_1)$.

Proof. Define $\bar{y} = P_\gamma^{0 \leftarrow 1} \operatorname{grad} f(x_2) - \operatorname{grad} f(x_1)$, where P is the parallel transport along the curve γ defined by $\gamma(t) = R_{x_1}(ts)$. From Lemma 3.3.8, we have

$$\|\bar{y} - \bar{H} s\| \leq b_0 \|s\|^2, \tag{3.3.15}$$

where $\bar{H} = \int_0^1 P_\gamma^{0 \leftarrow t} \operatorname{Hess} f(\gamma(t)) P_\gamma^{t \leftarrow 0} dt$ and b_0 is a constant. We then have

$$\begin{aligned}
&\|y - \mathcal{T}_{S_{\zeta_1}} \operatorname{Hess} f(x^*) \mathcal{T}_{S_{\zeta_1}}^{-1} s\| \\
\leq\ & \|y - \bar{y}\| + \|\bar{y} - \bar{H} s\| + \|\bar{H} s - \mathcal{T}_{S_{\zeta_1}} \operatorname{Hess} f(x^*) \mathcal{T}_{S_{\zeta_1}}^{-1} s\| \\
=\ & \|\mathcal{T}_{S_\zeta}^{-1} \operatorname{grad} f(x_2) - P_\gamma^{0 \leftarrow 1} \operatorname{grad} f(x_2)\| + b_0 \|s\|^2 + \|\bar{H} - \mathcal{T}_{S_{\zeta_1}} \operatorname{Hess} f(x^*) \mathcal{T}_{S_{\zeta_1}}^{-1}\| \|s\| \\
\leq\ & b_1 \|s\| \max\{\operatorname{dist}(x_1, x^*), \operatorname{dist}(x_2, x^*)\} + b_0 \|s\|^2 \text{ (by Lemma 3.3.6)} \\
& + (\| \int_0^1 P_\gamma^{0 \leftarrow t} \operatorname{Hess} f(\gamma(t)) P_\gamma^{t \leftarrow 0} dt - \operatorname{Hess} f(x_1)\| \\
& + \| \operatorname{Hess} f(x_1) - \mathcal{T}_{S_{\zeta_1}} \operatorname{Hess} f(x^*) \mathcal{T}_{S_{\zeta_1}}^{-1}\|) \|s\| \\
\leq\ & b_2 \|s\| \max\{\operatorname{dist}(x_1, x^*), \operatorname{dist}(x_2, x^*)\}, \text{ (by Assumption 3.3.3)}
\end{aligned}$$

where b_1 and b_2 are some constants. □

With these technical lemmas in place, we now start the Riemannian generalization of the sequence of lemmas in [BKS96] that leads to the main result [BKS96, Theorem 2.7], generalized here as Theorem 3.3.2. For an easier comparison with [BKS96], in the rest of the convergence analysis, we let n (instead of d) denote the dimension of the manifold $\mathcal{M}$.

The next lemma generalizes [BKS96, Lemma 2.3], itself a slight variation of [KBS93, Lemma 3.2]. The proof of [BKS96, Lemma 2.3] involves considering the span of a few s_j's. In the Riemannian setting, a difficulty arises from the fact that the s_j's are not in the same tangent space. We overcome this difficulty by transporting the s_j's to $\mathrm{T}_{x^*} \mathcal{M}$.

Lemma 3.3.13. *Let s_k be such that $R_{x_k}(s_k) \to x^*$. If Assumptions 3.3.1, 3.3.2, 3.3.3, and 3.3.5 hold then there exists $K \geq 0$ such that for any set of $n+1$ steps $S = \{s_{k_j} : K \leq k_1 < \ldots < k_{n+1}\}$, there exists an index k_m with $m \in \{2, 3, \ldots, n+1\}$ such that*

$$\frac{\|(\mathcal{B}_{k_m} - H_{k_m})s_{k_m}\|}{\|s_{k_m}\|} < (a_{12}a_{10}^{k_{n+1}-k_1-2} + \bar{a}_{12})\epsilon_S^{\frac{1}{n}},$$

where $\epsilon_S = \max_{1\leq j\leq n+1}\{\mathrm{dist}(x_{k_j}, x^*), \mathrm{dist}(R_{x_{k_j}}(s_{k_j}), x^*)\}$, $H_{k_m} = \mathcal{T}_{S_{\zeta_{k_m}}} \operatorname{Hess} f(x^*) \mathcal{T}_{S_{\zeta_{k_m}}}^{-1}$, $\zeta_{k_m} = R_{x^*}^{-1} x_{k_m}$, $a_{12}, \bar{a}_{12}$ are some constants, and n is the dimension of the manifold.

Proof. Given S, for $j = 1, 2, \ldots, n+1$, define

$$S_j = \left[\frac{\bar{s}_{k_1}}{\|\bar{s}_{k_1}\|}, \frac{\bar{s}_{k_2}}{\|\bar{s}_{k_2}\|}, \ldots, \frac{\bar{s}_{k_j}}{\|\bar{s}_{k_j}\|}\right],$$

where $\bar{s}_{k_i} = \mathcal{T}_{S_{\zeta_{k_i}}}^{-1} s_{k_i}, i = 1, 2, \ldots, j$. The proof is organized as follows. We will first obtain in (3.3.24) that there exists $m \in [2, n+1]$ and $u \in \mathbb{R}^{m-1}$, $w \in \mathrm{T}_{x^*}\mathcal{M}$ such that $\bar{s}_{k_m}/\|\bar{s}_{k_m}\| = S_{m-1}u - w$, S_{m-1} has full column rank and is well conditioned, and $\|w\|$ is small. We will also obtain in (3.3.26) that $(\mathcal{T}_{S_{\zeta_{k_m}}}^{-1} \mathcal{B}_{k_m} \mathcal{T}_{S_{\zeta_{k_m}}} - \operatorname{Hess} f(x^*))S_{m-1}$ is small due to the Hessian approximating properties of the SR1 update given in Lemma 3.3.12 above. The conclusion follows from these two results.

Let G_* denote the matrix expression of inner product of $\mathrm{T}_{x^*}\mathcal{M}$ and $\hat{S}_j$ denote the coordinate expression of S_j, for $j \in \{1, \ldots, n\}$. Let κ_j be the smallest singular value of $G_*^{1/2}\hat{S}_j$ and define $\kappa_{n+1} = 0$. We have

$$1 = \kappa_1 \geq \kappa_2 \ldots \geq \kappa_{n+1} = 0.$$

Let m be the smallest integer for which

$$\frac{\kappa_m}{\kappa_{m-1}} < \epsilon_S^{\frac{1}{n}}. \tag{3.3.16}$$

Since $m \leq n+1$ and $\kappa_1 = 1$, we have

$$\kappa_{m-1} = \kappa_1(\frac{\kappa_2}{\kappa_1}) \ldots (\frac{\kappa_{m-1}}{\kappa_{m-2}}) > \epsilon_S^{(m-2)/n} > \epsilon_S^{(n-1)/n}. \tag{3.3.17}$$

Since $x_k \to x^*$ and $R_{x_k}(s_k) \to x^*$, we can assume that $\epsilon_S \in (0, (\frac{1}{4})^n)$ for all k. Now, we choose $z \in \mathbb{R}^m$ such that

$$\|G_*^{1/2}\hat{S}_m z\|_2 = \kappa_m \|z\|_2 \tag{3.3.18}$$

and

$$z = \begin{pmatrix} u \\ -1 \end{pmatrix},$$

where $u \in \mathbb{R}^{m-1}$. (The last component of z is nonzero due to that m is the smallest such that (3.3.16) is true.) Let $w = S_m z$ and its coordinate expression $\hat{w} = \hat{S}_m z$. From the definition of $G_*^{1/2}\hat{S}_m$ and z, we have

$$G_*^{1/2}\hat{S}_{m-1}u - G_*^{1/2}\hat{w} = \frac{G_*^{1/2}\hat{\bar{s}}_{k_m}}{\|G_*^{1/2}\hat{\bar{s}}_{k_m}\|_2}, \tag{3.3.19}$$

where $\hat{\bar{s}}_{k_m}$ is the coordinate expression of $\bar{s}_{k_m}$. Since κ_{m-1} is the smallest singular value of $G_*^{1/2}\hat{S}_{m-1}$, we have that

$$\|u\|_2 \leq \frac{1}{\kappa_{m-1}}\|G_*^{1/2}\hat{S}_{m-1}u\|_2 = \frac{1}{\kappa_{m-1}}\|G_*^{1/2}\hat{w} + \frac{G_*^{1/2}\hat{\bar{s}}_{k_m}}{\|G_*^{1/2}\hat{\bar{s}}_{k_m}\|_2}\|_2 \leq \frac{\|G_*^{1/2}\hat{w}\|_2 + 1}{\kappa_{m-1}} = \frac{\|w\| + 1}{\kappa_{m-1}} \tag{3.3.20}$$

$$< \frac{\|G_*^{1/2}\hat{w}\|_2 + 1}{\epsilon_S^{(n-1)/n}} = \frac{\|w\| + 1}{\epsilon_S^{(n-1)/n}}. \text{ (by (3.3.17))} \tag{3.3.21}$$

Using (3.3.18) and (3.3.20), we have that

$$\begin{aligned}\|w\|^2 &= \|G_*^{1/2}\hat{w}\|_2^2 = \|G_*^{1/2}\hat{S}_m z\|_2^2 = \kappa_m^2\|z\|_2^2 = \kappa_m^2(1 + \|u\|_2^2) \\ &\leq \kappa_m^2 + (\frac{\kappa_m}{\kappa_{m-1}})^2(\|G_*^{1/2}\hat{w}\|_2 + 1)^2 = \kappa_m^2 + (\frac{\kappa_m}{\kappa_{m-1}})^2(\|w\| + 1)^2.\end{aligned}$$

Therefore, since (3.3.16) implies that $\kappa_m < \epsilon_S^{1/n}$, using (3.3.16),

$$\|w\|^2 < \epsilon_S^{2/n} + \epsilon_S^{2/n}(\|w\| + 1)^2 < 4\epsilon_S^{2/n}(\|w\| + 1)^2. \tag{3.3.22}$$

This implies

$$\|w\|(1 - 2\epsilon_S^{1/n}) < 2\epsilon_S^{1/n},$$

and hence $\|w\| < 1$, since $\epsilon_S < (\frac{1}{4})^n$. Therefore, (3.3.21) and (3.3.22) imply that

$$\|u\|_2 < \frac{2}{\epsilon_S^{(n-1)/n}} \tag{3.3.23}$$

$$\|w\| < 4\epsilon_S^{1/n}.. \tag{3.3.24}$$

Equation (3.3.24) is the announced result that w is small. The bound (3.3.23) will also be invoked below.

Now we show that $\|(\mathcal{T}_{S_{\zeta_{k_j}}}^{-1}\mathcal{B}_{k_j}\mathcal{T}_{S_{\zeta_{k_j}}} - \operatorname{Hess} f(x^*))S_{j-1}\|$ is small for all $j \in [2, n+1]$ (and thus in particular for $j = m$). By Lemma 3.3.11, we have

$$\|y_i - (\mathcal{B}_{k_j})_i s_i\| \leq a_9 a_{10}^{k_j - i - 2}\epsilon_{k_j,i}\|s_i\| \leq a_9 a_{10}^{k_{n+1} - k_1 - 2}\epsilon_S\|s_i\|, \tag{3.3.25}$$

for all $i \in \{k_1, k_2, \ldots, k_{j-1}\}$. Therefore,

$$\begin{aligned}
&\|(\mathcal{T}_{S_{\zeta_{k_j}}}^{-1}\mathcal{B}_{k_j}\mathcal{T}_{S_{\zeta_{k_j}}} - \operatorname{Hess} f(x^*))\frac{\bar{s}_i}{\|\bar{s}_i\|}\| \\
&\leq \|\frac{\mathcal{T}_{S_{\zeta_i}}^{-1} y_i - \mathcal{T}_{S_{\zeta_{k_j}}}^{-1}\mathcal{B}_{k_j}\mathcal{T}_{S_{\zeta_{k_j}}}\bar{s}_i}{\|\bar{s}_i\|}\| + \|\frac{\mathcal{T}_{S_{\zeta_i}}^{-1} y_i - \operatorname{Hess} f(x^*)\bar{s}_i}{\|\bar{s}_i\|}\| \\
&\leq \|\frac{\mathcal{T}_{S_{\zeta_i}}^{-1} y_i - \mathcal{T}_{S_{\zeta_{k_j}}}^{-1}\mathcal{B}_{k_j}\mathcal{T}_{S_{\zeta_{k_j}}}\bar{s}_i}{\|\bar{s}_i\|}\| + b_1\epsilon_S \text{ (by Lemma 3.3.12)} \\
&= \|\frac{\mathcal{T}_{S_{\zeta_i}}^{-1}(y_i - \mathcal{T}_{S_{\zeta_i}}\mathcal{T}_{S_{\zeta_{k_j}}}^{-1}\mathcal{B}_{k_j}\mathcal{T}_{S_{\zeta_{k_j}}}\mathcal{T}_{S_{\zeta_i}}^{-1} s_i)}{\|\bar{s}_i\|}\| + b_1\epsilon_S \\
&\leq b_2\frac{\|(y_i - (\mathcal{B}_{k_j})_i s_i)\|}{\|s_i\|} + b_3\epsilon_S \text{ (by Lemma 3.3.10 and Assumption 3.3.1)} \\
&\leq (b_4 a_{10}^{k_{n+1}-k_1-2} + b_3)\epsilon_S \text{ (by (3.3.25))}
\end{aligned}$$

where b_2, b_3 and b_4 are some constants. Therefore, we have that for any $j \in [2, n+1]$,

$$\|(\mathcal{T}_{S_{\zeta_{k_j}}}^{-1}\mathcal{B}_{k_j}\mathcal{T}_{S_{\zeta_{k_j}}} - \operatorname{Hess} f(x^*))S_{j-1}\|_{g,2} \leq b_5\epsilon_S, \tag{3.3.26}$$

where $b_5 = \sqrt{n}(b_4 a_{10}^{k_{n+1}-k_1-2} + b_3)$ and $\|\cdot\|_{g,2}$ is the norm induced by the Riemannian metric g and the Euclidean norm, i.e., $\|A\|_{g,2} = \sup \|Av\|_g / \|v\|_2$.

We can now conclude the proof as follows. Using (3.3.19) and (3.3.26) with $j = m$, (3.3.23) and (3.3.24), we have

$$\begin{aligned}
&\frac{\|(\mathcal{T}_{S_{\zeta_{k_m}}}^{-1}\mathcal{B}_{k_m}\mathcal{T}_{S_{\zeta_{k_m}}} - \operatorname{Hess} f(x^*))\bar{s}_m\|}{\|\bar{s}_m\|} \\
&= \|(\mathcal{T}_{S_{\zeta_{k_m}}}^{-1}\mathcal{B}_{k_m}\mathcal{T}_{S_{\zeta_{k_m}}} - \operatorname{Hess} f(x^*))(S_{m-1}u - w)\| \\
&\leq \|(\mathcal{T}_{S_{\zeta_{k_m}}}^{-1}\mathcal{B}_{k_m}\mathcal{T}_{S_{\zeta_{k_m}}} - \operatorname{Hess} f(x^*))S_{m-1}\|_{g,e}\|u\|_2 + \|(\mathcal{T}_{S_{\zeta_{k_m}}}^{-1}\mathcal{B}_{k_m}\mathcal{T}_{S_{\zeta_{k_m}}} - \operatorname{Hess} f(x^*))\|\|w\| \\
&\leq b_5\epsilon_S\frac{2}{\epsilon_S^{(n-1)/n}} + (M + \operatorname{Hess} f(x^*))4\epsilon_S^{1/n} \text{ (by Assumption 3.3.1)} \\
&\leq (2b_5 + b_6)\epsilon_S^{1/n}
\end{aligned}$$

where b_6 is some constant. Finally,

$$\begin{aligned}
\frac{\|(\mathcal{B}_{k_m} - H_{k_m})s_{k_m}\|}{\|s_{k_m}\|} &= \frac{\|(\mathcal{B}_{k_m} - \mathcal{T}_{S_{\zeta_{k_m}}}\operatorname{Hess} f(x^*)\mathcal{T}_{S_{\zeta_{k_m}}}^{-1})s_{k_m}\|}{\|s_{k_m}\|} \\
&= \frac{\|(\mathcal{T}_{S_{\zeta_{k_m}}}^{-1}\mathcal{B}_{k_m}\mathcal{T}_{S_{\zeta_{k_m}}} - \operatorname{Hess} f(x^*))\bar{s}_{k_m}\|}{\|\bar{s}_{k_m}\|} \\
&\leq (2b_5 + b_6)\epsilon_S^{1/n}
\end{aligned}$$

completes the proof. □

The next lemma generalizes [BKS96, Lemma 2.4]. Its proof is a translation of the proof of [BKS96, Lemma 2.4], where we invoke two manifold-specific results: the equality of $\operatorname{Hess} f(x^*)$ and $\operatorname{Hess}(f \circ R_{x^*})(0_{x^*})$ (which holds in view of [AMS08, Proposition 5.5.6] since x^* is a critical point of f), and the bound in Lemma 3.3.3 on the retraction R.

Lemma 3.3.14. *Suppose that Assumptions 3.3.1, 3.3.2, 3.3.3, 3.3.4, 3.3.5 and 3.3.6 hold and the trust region subproblem* (3.2.1) *is solved accurately enough for (3.3.2) to hold. Then there exists N such that for any set of $p > n$ consecutive steps $s_{k+1}, s_{k+1}, \ldots, s_{k+p}$ with $k \geq N$, there exists a set, $\mathcal{G}_k$, of at least $p - n$ indices contained in the set $\{i : k+1 \leq i \leq k+p\}$ such that for all $j \in \mathcal{G}_k$,*

$$\frac{\|(\mathcal{B}_j - H_j)s_j\|}{\|s_j\|} < a_{13}\epsilon_k^{\frac{1}{n}},$$

where $a_{13} = a_{12}a_{10}^{p-2} + \bar{a}_{12}$, $H_j = \mathcal{T}_{S_{\zeta_j}} \operatorname{Hess} f(x^*)\mathcal{T}_{S_{\zeta_j}}^{-1}$, $\zeta_j = R_{x^*}^{-1}x_j$, and

$$\epsilon_k = \max_{k+1 \leq j \leq k+p}\{\operatorname{dist}(x_j, x^*), \operatorname{dist}(R_{x_j}(s_j), x^*)\}.$$

Furthermore, for k sufficiently large, if $j \in \mathcal{G}_k$, then

$$\|s_j\| < a_{14}\operatorname{dist}(x_j, x^*), \tag{3.3.27}$$

where a_{14} is a constant, and

$$\rho_j \geq 0.75. \tag{3.3.28}$$

Proof. By Lemma 3.3.5, $s_k \to 0$. Therefore, by Lemma 3.3.13, applied to the set

$$\{s_k, s_{k+1}, \ldots, s_{k+p}\}, \tag{3.3.29}$$

there exists N such that for any $k \geq N$ there exists an index l_1, with $k+1 \leq l_1 \leq k+p$ satisfying

$$\frac{\|(\mathcal{B}_{l_1} - H_{l_1})s_{l_1}\|}{\|s_{l_1}\|} < a_{13}\epsilon_k^{\frac{1}{n}},$$

where $a_{13} = a_{12}a_{10}^{p-2} + \bar{a}_{12}$. Now we can apply Lemma 3.3.13 to the set $\{s_k, s_{k+1}, \ldots, s_{k+p}\} - s_{l_1}$ to get l_2. Repeating this $p - n$ times, we get a set of $p - n$ indices $\mathcal{G}_k = \{l_1, l_2, \ldots, l_{p-n}\}$ such that if $j \in \mathcal{G}_k$, then

$$\frac{\|(\mathcal{B}_j - H_j)s_j\|}{\|s_j\|} < a_{13}\epsilon_k^{\frac{1}{n}}. \tag{3.3.30}$$

We show (3.3.27) next. Consider $j \in \mathcal{G}_k$. By (3.3.30), we have

$$g(s_j, (H_j - \mathcal{B}_j)s_j) \leq \|s_j\| \|(H_j - \mathcal{B}_j)s_j\| \leq a_{13}\epsilon_k^{\frac{1}{n}} \|s_j\|^2.$$

Therefore,

$$\begin{aligned} g(s_j, \mathcal{B}_j s_j) &\geq g(s_j, H_j s_j) - a_{13}\epsilon_k^{\frac{1}{n}} \|s_j\|^2 \\ &> b_0 \|s_j\|^2, \text{ (choosing } k \text{ large enough)} \end{aligned}$$

where b_0 is a constant and we have

$$\begin{aligned} 0 \leq m_j(0) - m_j(s_j) &= -g(\operatorname{grad} f(x_j), s_j) - \frac{1}{2} g(s_j, \mathcal{B}_j s_j) \\ &\leq \| \operatorname{grad} f(x_j)\| \|s_j\| - \frac{1}{2} b_0 \|s_j\|^2 \\ &\leq b_1 \operatorname{dist}(x_j, x^*) \|s_j\| - \frac{1}{2} b_0 \|s_j\|^2, \text{ (by Lemma 3.3.7)} \end{aligned}$$

where b_1 is some constant. This yields (3.3.27).

Finally, we show (3.3.28). Let $j \in \mathcal{G}_k$ and define $\hat{f}_x(\eta) = f(R_x(\eta))$. It follows that

$$\begin{aligned} &|f(x_j) - f(R_{x_j}(s_j)) - (m_j(0) - m_j(s_j))| \\ = &|f(x_j) - f(R_{x_j}(s_j)) + g(\operatorname{grad} f(x_j), s_j) + \frac{1}{2} g(s_j, \mathcal{B}_j s_j)| \\ = &|\hat{f}_{x_j}(0_{x_j}) - \hat{f}_{x_j}(s_j) + g(\operatorname{grad} f(x_j), s_j) + \frac{1}{2} g(s_j, \mathcal{B}_j s_j)| \\ = &|\frac{1}{2} g(s_j, \mathcal{B}_j s_j) - \int_0^1 g(\operatorname{Hess} \hat{f}_{x_j}(\tau s_j)[s_j], s_j)(1-\tau) d\tau| \text{ (by Taylor's theorem)} \\ \leq &|\frac{1}{2} g(s_j, \mathcal{B}_j s_j) - \frac{1}{2} g(s_j, H_j s_j)| + |\frac{1}{2} g(s_j, H_j s_j) - \int_0^1 g(\operatorname{Hess} \hat{f}_{x_j}(\tau s_j)[s_j], s_j)(1-\tau) d\tau| \\ = &|\frac{1}{2} g(s_j, (\mathcal{B}_j - H_j) s_j)| \\ &+ |\int_0^1 (g(s_j, \mathcal{T}_{S_{\zeta_j}} \operatorname{Hess} f(x^*) \mathcal{T}_{S_{\zeta_j}}^{-1} s_j) - g(\operatorname{Hess} \hat{f}_{x_j}(\tau s_j)[s_j], s_j))(1-\tau) d\tau| \\ \leq &\|s_j\|^2 \int_0^1 \|(\mathcal{T}_{S_{\zeta_j}} \operatorname{Hess} \hat{f}_{x^*}(0_{x^*}) \mathcal{T}_{S_{\zeta_j}}^{-1} - \operatorname{Hess} \hat{f}_{x_j}(\tau s_j))\| (1-\tau) d\tau \text{ (by [AMS08, proposition 5.5.6])} \\ &+ \frac{1}{2} \|s_j\| \|(\mathcal{B}_j - H_j) s_j\| \\ \leq &b_3 \|s_j\|^2 (\operatorname{dist}(x_j, x^*) + \|s_j\|) + b_2 \|s_j\|^2 \epsilon_k^{\frac{1}{n}} \text{ (by (3.3.30), Lemma 3.3.3 and Assumption 3.3.4)} \\ \leq &b_4 \|s_j\|^2 \epsilon_k^{\frac{1}{n}}, \text{ (by (3.3.27) and } \operatorname{dist}(x_j, x^*) \text{ is smaller than } \epsilon_k^{\frac{1}{n}} \text{ eventually)} \end{aligned}$$

where b_2, b_3 and b_4 are some constants. In view of (3.3.27) and Lemma 3.3.7, we have

$$\|s_j\| < b_5 \| \operatorname{grad} f(x_j)\|,$$

where b_5 is some constant. Combining with $\|s_j\| \leq \Delta_j$, we obtain

$$\|s_j\|^2 \leq b_5 \| \operatorname{grad} f(x_j)\| \min\{\Delta_j, b_5 \| \operatorname{grad} f(x_j)\|\}.$$

Noticing (3.3.2), we have

$$|f(x_j) - f(R_{x_j}(s_j)) - (m_j(0) - m_j(s_j))| \leq b_6 \epsilon_k^{\frac{1}{n}} (m_j(0) - m_j(s_j)),$$

where b_6 is a constant. This implies (3.3.28). □

The next result generalizes [BKS96, Lemma 2.5] in two ways: the Euclidean setting is extended to the Riemannian setting, and inexact solves are allowed by the presence of δ_k. The main hurdle that we had to overcome in the Riemannian generalization is that the equality $\operatorname{dist}(x_k + s_k, x^*) = \|s_k - \xi_k\|$ does not necessarily hold. As we will see, Lemma 3.3.2 comes to our rescue.

Lemma 3.3.15. *Suppose Assumptions 3.3.2 and 3.3.3 hold. If the quantities*

$$e_k = \operatorname{dist}(x_k, x^*) \text{ and } \frac{\|(\mathcal{B}_k - H_k)s_k\|}{\|s_k\|}$$

are sufficiently small and if $\mathcal{B}_k s_k = -\operatorname{grad} f(x_k) + \delta_k$ *with* $\|\delta_k\| \leq \| \operatorname{grad} f(x_k)\|^{1+\theta}$*, then*

$$\operatorname{dist}(R_{x_k}(s_k), x^*) \leq a_{15} \frac{\|(\mathcal{B}_k - H_k)s_k\|}{\|s_k\|} e_k + a_{16} e_k^{1+\min\{\theta,1\}}, \tag{3.3.31}$$

$$h(R_{x_k}(s_k)) \leq a_{17} \frac{\|(\mathcal{B}_k - H_k)s_k\|}{\|s_k\|} h(x_k) + a_{18} h^{1+\min\{\theta,1\}}(x_k), \tag{3.3.32}$$

and

$$a_{19} h(x_k) \leq e_k \leq a_{20} h(x_k) \tag{3.3.33}$$

where a_{15}, a_{16}, a_{17} *and* a_{18} *are some constants and* $h(x) = (f(x) - f(x^*))^{\frac{1}{2}}$.

Proof. By definition of s_k, we have

$$s_k = H_k^{-1}[(H_k - \mathcal{B}_k)s_k - \operatorname{grad} f(x_k) + \delta_k]. \tag{3.3.34}$$

Define $\xi_k = R_{x_k}^{-1} x^*$. Therefore, letting γ be the curve defined by $\gamma(t) = R_{x_k}(t\xi_k)$, we have

$$
\begin{aligned}
&\|s_k - \xi_k\| \\
&= \|H_k^{-1}[(H_k - \mathcal{B}_k)s_k - \operatorname{grad} f(x_k) + \delta_k - H_k\xi_k]\| \\
&\leq b_0(\|(H_k - \mathcal{B}_k)s_k\| + \|P_\gamma^{0\leftarrow 1} \operatorname{grad} f(x^*) - \operatorname{grad} f(x_k) - (\int_0^1 P_\gamma^{0\leftarrow t} \operatorname{Hess} f(\gamma(t)) P_\gamma^{t\leftarrow 0} dt)\xi_k\| \\
&+ \|(\int_0^1 P_\gamma^{0\leftarrow t} \operatorname{Hess} f(\gamma(t)) P_\gamma^{t\leftarrow 0} dt)\xi_k - \operatorname{Hess} f(x_k)\xi_k\| + \|\operatorname{Hess} f(x_k)\xi_k - H_k\xi_k\| + \|\delta_k\|) \\
&\leq b_0(\|(H_k - \mathcal{B}_k)s_k\| + b_1\|\xi_k\|^{1+\min\{\theta,1\}} \text{ (by Lemmas 3.3.7 and 3.3.8)} \\
&+ \|(\int_0^1 P_\gamma^{0\leftarrow t} \operatorname{Hess} f(\gamma(t)) P_\gamma^{t\leftarrow 0} dt)\xi_k - \operatorname{Hess} f(x_k)\xi_k\| + \|\operatorname{Hess} f(x_k) - H_k\|\|\xi_k\|) \\
&\leq b_0\|(H_k - \mathcal{B}_k)s_k\| + b_0 b_1\|\xi_k\|^{1+\min\{\theta,1\}} + b_0 b_3\|\xi_k\|^2 \text{ (by Assumption 3.3.3)} \\
&\leq b_0\|(H_k - \mathcal{B}_k)s_k\| + b_4\|\xi_k\|^{1+\min\{\theta,1\}} \qquad (3.3.35)
\end{aligned}
$$

where b_1, b_2, b_3 and b_4 are some constants. From Lemma 3.3.2, we have

$$\operatorname{dist}(R_{x_k}(s_k), x^*) = \operatorname{dist}(R_{x_k}(s_k), R_{x_k}(\xi_k)) \leq b_5\|s_k - \xi_k\|, \qquad (3.3.36)$$

where b_5 is a constant. Combining (3.3.35) and (3.3.36) and using Lemma 3.3.3, we obtain

$$\operatorname{dist}(R_{x_k}(s_k), x^*) \leq b_0 b_5\|(H_k - \mathcal{B}_k)s_k\| + \bar{b}_4 b_5 e_k^{1+\min\{\theta,1\}}. \qquad (3.3.37)$$

From (3.3.34), for k large enough such that $\|H_k^{-1}\|\|(H_k - \mathcal{B}_k)s_k\| \leq \frac{1}{2}\|s_k\|$, we have

$$\|s_k\| \leq \frac{1}{2}\|s_k\| + \|H_k^{-1}\|(\|\operatorname{grad} f(x_k)\| + \|\operatorname{grad} f(x_k)\|^{1+\theta}).$$

Using Lemma 3.3.7, this yields

$$\|s_k\| \leq b_6 \operatorname{dist}(x_k, x^*),$$

where b_6 is a constant. Using the latter in (3.3.37) yields

$$\operatorname{dist}(R_{x_k}(s_k), x^*) \leq b_0 b_5 b_6 \frac{\|(H_k - \mathcal{B}_k)s_k\|}{\|s_k\|} \operatorname{dist}(x_k, x^*) + \bar{b}_4 b_5 e_k^{1+\min\{\theta,1\}},$$

which shows (3.3.31).

We next show (3.3.33). Define $\hat{f}_x(\eta) = f(R_x(\eta))$ and let $\zeta_k = R_{x^*}^{-1} x_k$. We have, for some $t \in (0,1)$,

$$
\begin{aligned}
\hat{f}_{x^*}(\zeta_k) - \hat{f}_{x^*}(0_{x^*}) &= g(\operatorname{grad} f(x^*), \zeta_k) + g(\operatorname{Hess} \hat{f}_{x^*}(t\zeta_k)[\zeta_k], \zeta_k) \\
&= g(\operatorname{Hess} \hat{f}_{x^*}(t\zeta_k)[\zeta_k], \zeta_k),
\end{aligned}
$$

where we have used (Euclidean) Taylor's theorem to get the first equality and the fact that x^* is a critical point of f (Assumption 3.3.2) for the second one. Therefore, since $\operatorname{Hess} \hat{f}_{x^*} = \operatorname{Hess} f(x^*)$ is positive definite (in view of [AMS08, Proposition 5.5.6] and Assumption 3.3.2), there exist b_7 and b_8 such that

$$b_7(\hat{f}_{x^*}(\zeta_k) - \hat{f}_{x^*}(0_{x^*})) \leq \|\zeta_k\|^2 \leq b_8(\hat{f}_{x^*}(\zeta_k) - \hat{f}_{x^*}(0_{x^*}))$$

Then, using Lemma 3.3.3, we obtain that there exist b_9 and b_{10} such that

$$b_9(f(x_k) - f(x^*)) \leq \operatorname{dist}(x_k, x^*)^2 \leq b_{10}(f(x_k) - f(x^*)).$$

In other words,

$$b_9 h^2(x_k) \leq e_k^2 \leq b_{10} h^2(x_k),$$

and we have shown (3.3.33). Combining it with (3.3.31), we get (3.3.32). □

With Lemmas 3.3.14 and 3.3.15 in place, the rest of the local convergence analysis is essentially a translation of the analysis in [BKS96]. The next lemma generalizes [BKS96, Lemma 2.6].

Lemma 3.3.16. *If Assumptions 3.3.1, 3.3.2, 3.3.3, 3.3.4, 3.3.5, and 3.3.6 hold and the subproblem is solved accurately enough for* (3.3.2) *and* (3.3.3) *to hold then,*

$$\lim_{k\to\infty} \frac{h_k}{\Delta_k} = 0,$$

where $h_k = h(x_k)$.

Proof. Let p be the smallest integer greater than $2n + n(-\ln \tau_1 / \ln \tau_2)$, where τ_1 and τ_2 are defined in Algorithm 1. Then

$$\tau_1^n \tau_2^{p-2n} \geq 1. \tag{3.3.38}$$

Applying Lemma 3.3.14 with this value of p, there exists N such that if $k \geq N$, then there exists a set of at least $p - n$ indices, $\mathcal{G}_k \subset \{j : k+1 \leq j \leq k+p\}$, such that if $j \in \mathcal{G}_k$, then

$$\frac{\|(\mathcal{B}_j - H_j)s_j\|}{\|s_j\|} < c\epsilon_k^{\frac{1}{n}} \tag{3.3.39}$$

$$\rho_j \geq 0.75.$$

We now show that for such steps,

$$\frac{h_{j+1}}{\Delta_{j+1}} \leq \frac{1}{\tau_2}\frac{h_j}{\Delta_j}. \tag{3.3.40}$$

If $\|s_j\| \geq 0.8\Delta_j$, then since from Step 12 of Algorithm 1, $\Delta_{j+1} = \tau_2 \Delta_j$ and since $\{h_i\}$ is decreasing, (3.3.40) follows. If on the other hand $\|s_j\| < 0.8\Delta_j$, then from Step 14 of Algorithm 1, we have that $\Delta_{j+1} = \Delta_j$. Also since the trust region is inactive, by condition (3.3.3), we have that $\mathcal{B}_j s_j = -\operatorname{grad} f(x_j) + \delta_k$, $\|\delta_k\| \leq \|\operatorname{grad} f(x_j)\|^{1+\theta}$. Therefore, in view of (3.3.32) in Lemma 3.3.15 and of (3.3.39), if N is large enough, we have that

$$h_{j+1} \leq \frac{1}{\tau_2} h_j.$$

This implies that (3.3.40) is true for all $j \in \mathcal{G}_j$, where $k \geq N$.

In addition, note that for any j, $h_{j+1} \leq h_j$ and $\Delta_{j+1} \geq \tau_1 \Delta_j$ and so

$$\frac{h_{j+1}}{\Delta_{j+1}} \leq \frac{1}{\tau_1} \frac{h_j}{\Delta_j}. \tag{3.3.41}$$

Since (3.3.40) is true for $p - n$ values of $j \in \mathcal{G}_k$ and (3.3.41) holds for all j, we have that for all $k \geq N$,

$$\frac{h_{k+p}}{\Delta_{k+p}} \leq (\frac{1}{\tau_1})^n (\frac{1}{\tau_2})^{p-n} \frac{h_k}{\Delta_k} \leq (\frac{1}{\tau_2})^n \frac{h_k}{\Delta_k},$$

where the second inequality follows from (3.3.38). Therefore, starting at $k = N$, it follows that

$$\frac{h_{N+lp}}{\Delta_{N+lp}} \to 0$$

as $l \to \infty$. Using (3.3.41) again, we complete the proof. □

The next result generalizes [BKS96, Theorem 2.7].

Theorem 3.3.2. *If Assumptions 3.3.1, 3.3.2, 3.3.3, 3.3.4, 3.3.5, and 3.3.6 hold and the subproblem is solved accurately enough for* (3.3.2) *and* (3.3.3) *to hold then, the sequence* $\{x_k\}$ *generated by Algorithm 1 is* $n+1$*-step q-superlinear (where* n *denotes the dimension of* $\mathcal{M}$*); i.e.,*

$$\frac{\operatorname{dist}(x_{k+n+1}, x^*)}{\operatorname{dist}(x_k, x^*)} \to 0.$$

Proof. By Lemma 3.3.14, there exists N such that if $k \geq N$, then the set of steps $\{s_{k+1}, \dots, s_{k+n+1}\}$ contains at least one step s_{k+j}, $1 \leq j \leq n+1$, for which

$$\frac{\|(\mathcal{B}_j - H_j)s_j\|}{\|s_j\|} < a_{13} \epsilon_k^{\frac{1}{n}}.$$

By (3.3.27) in Lemma 3.3.14 and (3.3.33) in Lemma 3.3.15 (when checking the assumptions, recall the standing assumption made in Section 3.3.3 that $e_k := \mathrm{dist}(x_k, x^*) \to 0$), there exists a constant b_0 such that

$$\|s_{k+j}\| < b_0 h_{k+j}.$$

Therefore, by Lemma 3.3.16, if N is large enough and $k \geq N$, then $\|s_{k+j}\| < 0.8\Delta_{k+j}$. By (3.3.3), this implies $\mathcal{B}_{k+j} s_{k+j} = -\operatorname{grad} f(x_{k+j}) + \delta_{k+j}$, with $\|\delta_{k+j}\| \leq \|\operatorname{grad} f(x_{k+j})\|^{1+\theta}$. Thus by inequality (3.3.32) of Lemma 3.3.15, if N is large enough and $k \geq N$, then

$$h_{k+j+1} = h(R_{x_{k+j}}(s_{k+j})) \leq (a_{17} a_{13} \epsilon_k^{\frac{1}{n}} + a_{18} h_{k+j}^{\min\{\theta,1\}}) h_{k+j}.$$

The first equality holds because (3.3.28) implies that the step is accepted. Since the sequence $\{h_i\}$ is decreasing, this implies that

$$h_{k+n+1} \leq (a_{17} a_{13} \epsilon_k^{\frac{1}{n}} + a_{18} h_{k+j}^{\min\{\theta,1\}}) h_k$$

By (3.3.33),

$$\begin{aligned} e_{k+n+1} &\leq a_{20} h_{k+n+1} \\ &\leq a_{20}(a_{17} a_{13} \epsilon_k^{\frac{1}{n}} + a_{18} h_{k+j}^{\min\{\theta,1\}}) h_k \\ &\leq a_{20}(a_{17} a_{13} \epsilon_k^{\frac{1}{n}} + a_{18} (\frac{e_k}{a_{19}})^{\min\{\theta,1\}}) \frac{e_k}{a_{19}}. \end{aligned}$$

This implies $n+1$-step q-superlinear convergence. □

It is also possible to extend to the Riemannian setting the result [BKS96, Theorem 2.8] that the percentage of $\mathcal{B}_k$ being positive semidefinite approaches 1 provided that $\mathcal{B}_k$ is positive semidefinite whenever $\|s_k\| \leq 0.8\Delta_k$. In the proof of [BKS96, Theorem 2.8], replace Lemma 2.6 by Lemma 3.3.16, Lemma 2.4 by Lemma 3.3.14, (2.14) by (3.3.27), and (2.9) by (3.3.33).

3.4 Limited Memory Version of RTR-SR1

In RTR-SR1 (Algorithm 1), storing $\mathcal{B}_{k+1} = \mathcal{T}_{\eta_k} \circ \tilde{\mathcal{B}}_{k+1} \circ \mathcal{T}_{\eta_k}^{-1}$ in matrix form may be inefficient for two reasons. The first reason, which is also present in the Euclidean case, is that $\tilde{\mathcal{B}}_{k+1} = \mathcal{B}_k + \frac{(y_k - \mathcal{B}_k s_k)(y_k - \mathcal{B}_k s_k)^\flat}{g(s_k, y_k - \mathcal{B}_k s_k)}$ is a rank-one modification of $\mathcal{B}_k$. The second reason, specific to the Riemannian setting, is that when $\mathcal{M}$ is a low-codimension submanifold of a Euclidean space $\mathcal{E}$, it may be beneficial to express $\mathcal{T}_{\eta_k}$ as the restriction to $\mathrm{T}_{x_k}\mathcal{M}$ of a low-rank modification of the identity, e.g., (9.2.17) and (9.2.20). Instead of storing full dense matrices, it may then be beneficial to store a few vectors that implicitly represent them. This is the purpose of the limited memory version of RTR-SR1 presented in this section.

The proposed limited memory RTR-SR1, called LRTR-SR1, is described in Algorithm 2. It relies on a Riemannian generalization of the compact representation of the classical (Euclidean) SR1 matrices presented in [BNS94, §5]. We set $\mathcal{B}_0 = \mathrm{id}$. At step $k > 0$, we first choose a basic Hessian approximation $\mathcal{B}_0^k$, which in the Riemannian setting becomes a linear transformation of $\mathrm{T}_{x_k}\mathcal{M}$. We advocate the choice

$$\mathcal{B}_0^k = \gamma_k \,\mathrm{id},$$

where

$$\gamma_k = \frac{g(y_{k-1}, y_{k-1})}{g(s_{k-1}, y_{k-1})},$$

which generalizes a choice usually made in the Euclidean case [NW06, (7.20)]. As in the Euclidean case, we let $S_{k,m}$ and $Y_{k,m}$ contain the (at most) m most recent corrections, which in the Riemannian setting must be transported to $\mathrm{T}_{x_k}\mathcal{M}$, yielding $S_{k,m} = \{s_{k-\ell}^{(k)}, s_{k-\ell+1}^{(k)}, \ldots, s_{k-1}^{(k)}\}$ and $Y_{k,m} = \{y_{k-\ell}^{(k)}, y_{k-\ell+1}^{(k)}, \ldots, y_{k-1}^{(k)}\}$, where $\ell = \min\{m, k\}$ and where $s^{(k)}$ denotes s transported to $\mathrm{T}_{x_k}\mathcal{M}$. We then have the following Riemannian generalization of the limited-memory update based on [BNS94, (5.2)]:

$$\mathcal{B}_k = \mathcal{B}_0^k + (Y_{k,m} - \mathcal{B}_0^k S_{k,m})(P_{k,m} - S_{k,m}^{\flat}\mathcal{B}_0^k S_{k,m})^{-1}(Y_{k,m} - \mathcal{B}_0^k S_{k,m})^{\flat}, \quad k > 0,$$

where $P_{k,m} = D_{k,m} + L_{k,m} + L_{k,m}^T$, $D_{k,m} = \mathrm{diag}\{g(s_{k-\ell}, y_{k-\ell}), g(x_{k-\ell+1}, y_{k-\ell+1}), \ldots, g(s_{k-1}, y_{k-1})\}$, and

$$(L_{k,m})_{i,j} = \begin{cases} g(s_{k-\ell+i-1}, y_{k-\ell+j-1}), & \text{if } i > j; \\ 0, & \text{otherwise.} \end{cases}$$

Moreover, letting $Q_{k,m}$ denote the matrix $S_{k,m}^{\flat} S_{k,m}$, we obtain

$$\mathcal{B}_k = \gamma_k \,\mathrm{id} + (Y_{k,m} - \gamma_k S_{k,m})(P_{k,m} - \gamma_k Q_{k,m})^{-1}(Y_{k,m} - \gamma_k S_{k,m})^{\flat}, \quad k > 0. \tag{3.4.1}$$

For all $\eta \in \mathrm{T}_{x_k}\mathcal{M}$, $\mathcal{B}_k\eta$ can thus be obtained from (3.4.1) using $Y_{k,m}$, $S_{k,m}$, $P_{k,m}$ and $Q_{k,m}$. This is how $\mathcal{B}_k$ is defined in Algorithm 2, except that the technicality that the $\mathcal{B}$ update may be skipped is also taken into account therein.

Algorithm 2 Limited-memory RTR-SR1 (LRTR-SR1)

```
Input: Riemannian manifold $\mathcal{M}$ with Riemannian metric $g$; retraction $R$; isometric vector transports $\mathcal{T}_S$; smooth function $f$ on $\mathcal{M}$; initial iterate $x_0 \in \mathcal{M}$;
1: Choose an integer $m > 0$ and real numbers $\Delta_0 > 0$, $\nu \in (0,1)$, $c \in (0, 0.1)$, $\tau_1 \in (0,1)$ and $\tau_2 > 1$; Set $k \leftarrow 0$, $\ell \leftarrow 0$, $\gamma_0 \leftarrow 1$;
2: Obtain $s_k \in \mathrm{T}_{x_k} \mathcal{M}$ by (approximately) solving
       $s_k = \min_{s \in \mathrm{T}_{x_k} \mathcal{M}} m_k(s) = \min_{s \in \mathrm{T}_{x_k} \mathcal{M}} f(x_k) + g(\mathrm{grad} f(x_k), s) + \frac{1}{2} g(s, \mathcal{B}_k s), \text{s.t. } \|s\| \leq \Delta_k,$
   where $\mathcal{B}_k$ is defined in accordance with (3.4.1);
3: Set $\rho_k \leftarrow \frac{f(x_k) - f(R_{x_k}(s_k))}{m_k(0) - m_k(s_k)}$;
4: Set $y_k \leftarrow \mathcal{T}_{S_{\eta_k}}^{-1} \mathrm{grad} f(R_{x_k}(s_k)) - \mathrm{grad} f(x_k)$;
5: if $|g(s_k, y_k - \mathcal{B}_k s_k)| \geq \nu \|s_k\| \|y_k - \mathcal{B}_k s_k\|$ then
6:     $\gamma_{k+1} \leftarrow \frac{g(y_k, y_k)}{g(s_k, y_k)}$; Add $s_k^{(k)}$ and $y_k^{(k)}$ into storage; If $\ell \geq m$, then discard vector pair $\{s_{k-\ell}^{(k)}, y_{k-\ell}^{(k)}\}$ from storage, else $\ell \leftarrow \ell + 1$; Compute matrices $P_{k,m}$ and $Q_{k,m}$ by updating $P_{k-1,m}$ and $Q_{k-1,m}$ if available;
7: else
8:     Set $\gamma_{k+1} \leftarrow \gamma_k$, $P_{k+1,m} \leftarrow P_{k,m}$, $Q_{k+1,m} \leftarrow Q_{k,m}$ and $\{s_k^{(k)}, y_k^{(k)}\} \leftarrow \{s_{k-1}^{(k)}, y_{k-1}^{(k)}\}, \ldots, \{s_{k-\ell+1}^{(k)}, y_{k-\ell+1}^{(k)}\} \leftarrow \{s_{k-\ell}^{(k)}, y_{k-\ell}^{(k)}\}$.
9: end if
10: if $\rho_k > c$ then
11:     $x_{k+1} \leftarrow R_{x_k}(s_k)$; Transport $s_{k-\ell+1}^{(k)}, s_{k-\ell+2}^{(k)}, \ldots, s_k^{(k)}$ and $y_{k-\ell+1}^{(k)}, y_{k-\ell+2}^{(k)}, \ldots, y_k^{(k)}$ from $\mathrm{T}_{x_k} \mathcal{M}$ to $\mathrm{T}_{x_{k+1}} \mathcal{M}$ by $\mathcal{T}_S$;
12: else
13:     $x_{k+1} \leftarrow x_k$;
14: end if
15: if $\rho_k > \frac{3}{4}$ then
16:     if $\|\eta_k\| \geq 0.8 \Delta_k$ then
17:         $\Delta_{k+1} \leftarrow \tau_2 \Delta_k$;
18:     else
19:         $\Delta_{k+1} \leftarrow \Delta_k$;
20:     end if
21: else if $\rho_k < 0.1$ then
22:     $\Delta_{k+1} \leftarrow \tau_1 \Delta_k$;
23: else
24:     $\Delta_{k+1} \leftarrow \Delta_k$;
25: end if
26: $k \leftarrow k + 1$, goto 2 until convergence.
```

CHAPTER 4

A BROYDEN FAMILY OF QUASI-NEWTON METHOD

4.1 Introduction

In the classical Euclidean setting, the Broyden class (see, e.g., [NW06, §6.3]) is a family of quasi-Newton methods that depend on a real parameter, ϕ. Its Hessian approximation update formula is $B_{k+1} = (1 - \phi_k)B_{k+1}^{\mathrm{BFGS}} + \phi_k B_{k+1}^{\mathrm{DFP}}$, where B_{k+1}^{BFGS} stands for the update obtained by the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method and B_{k+1}^{DFP} for the update of the Davidon–Fletcher–Powell (DFP) method. Therefore, all members of the Broyden class satisfy the well-known *secant equation*, central to many quasi-Newton methods. For many years, BFGS, $\phi = 0$, was the preferred member of the family, as it tends to perform better in numerical experiments. Analyzing the entire Broyden class was nevertheless a topic of interest for the insight given into the properties of quasi-Newton methods; see [BNY87] and the many references therein. Subsequently, it was found that negative values of ϕ are desirable [ZT88, BLN92] and recent results reported in [LV07b] indicate that a significant improvement can be obtained by exploiting the freedom offered by ϕ.

The idea of quasi-Newton methods on manifolds is not new, however, the literature of which we are aware restricts consideration to the BFGS quasi-Newton method. Gabay [Gab82] discussed a version using parallel translation on submanifolds of $\mathbb{R}^n$. Brace and Manton [BM06] applied a version on the Grassmann manifold to the problem of weighted low-rank approximations. Qi [QGA10] compared the performance of different vector transports for a version of BFGS on Riemannian manifolds. Savas and Lim proposed a BFGS and limited memory BFGS methods for problems with cost functions defined on a Grassmann manifold and applied the methods to the best multilinear rank approximation problem. Ring and Wirth [RW12] systematically analyzed a version of BFGS on Riemannian manifolds which requires differentiated retraction. Seibert et al. [SKH13] discussed the freedom available when generalizing BFGS to Riemannian manifolds and analyzed one generalization of BFGS method on Riemannian manifolds that are isometric to $\mathbb{R}^n$.

In view of the above considerations, generalizing the Broyden family to manifolds is an appealing endeavor, which we undertake in this chapter. For $\phi = 0$ (BFGS) the proposed algorithm is quite similar to the BFGS method of Ring and Wirth [RW12]. Notably, both methods rely on the framework of retraction and vector transport developed in [ADM02, AMS08]. The BFGS method of [RW12] is more general in the sense that it also considers infinite-dimensional manifolds. On the other hand, a characteristic of our work is that we strive to resort as little as possible to the derivative of the retraction. Specifically, the definition of y_k (which corresponds to the usual difference of gradients) in [RW12] involves $\mathrm{D} f_{R_{x_k}}(s_k)$, whose Riesz representation is $(\mathrm{D} R_{x_k}(s_k))^* \nabla f|_{R_{x_k}(s_k)}$, where R is the retraction and ∇ is the Levi-Civita affine connection. In contrast, our definition of y_k relies on the same isometric vector transport as the one that appears in the Hessian approximation update formula. This can be advantageous in situations where R is defined by means of a constraint restoration procedure that does not admit a closed-form expression. It may also be the case that the chosen R admits a closed-form expression but that its derivative is unknown to the user. The price to pay for using the isometric vector transport in y_k is satisfying a novel "locking condition". Fortunately, we show simple procedures that, given either an isometric vector transport or a retraction, can produce a retraction or an isometric vector transport such that the pair satisfies the locking condition. As a result, efficient and convergent algorithms can be developed. Another contribution with respect to [RW12] is that we propose a limited-memory version of the quasi-Newton algorithm for large-scale problems.

Chapter 4 is organized as follows. The RBroyden family of algorithms and the "locking condition" are defined in Section 4.2. A convergence analysis is presented in Section 4.3. Two methods of constructing an isometric vector transport and a method of constructing a retraction related to the "locking condition" are derived in Section 4.4. The limited-memory RBFGS is described in Section 4.5. Experiments illustrating the performance of the methods for several applications are presented in later appropriate chapters.

4.2 RBroyden Family of Methods

The proposed RBroyden family algorithm is described in Algorithm 3 where the isometric vector transport $\mathcal{T}_S$ is not necessarily smooth. Instead, the required properties are $\mathcal{T}_S \in C^0$ and for any

Algorithm 3 RBroyden family method

```
Input: Riemannian manifold $\mathcal{M}$ with Riemannian metric $g$; a retraction $R$; isometric vector transport $\mathcal{T}_S$, with $R$ as associated retraction, that satisfies (4.2.6); continuously differentiable real-valued function $f$ on $\mathcal{M}$, bounded below; initial iterate $x_0 \in \mathcal{M}$; initial Hessian approximation $\mathcal{B}_0$ which is a linear transformation of the tangent space $\mathrm{T}_{x_0}\mathcal{M}$ that is symmetric positive definite with respect to the metric $g$; convergence tolerance $\varepsilon > 0$; Wolfe condition constants $0 < c_1 < \frac{1}{2} < c_2 < 1$;
$k \leftarrow 0$;
while $\|\operatorname{grad} f(x_k)\| > \varepsilon$ do
    Obtain $\eta_k \in \mathrm{T}_{x_k}\mathcal{M}$ by solving $\mathcal{B}_k\eta_k = -\operatorname{grad} f(x_k)$;
    Set $x_{k+1} = R_{x_k}(\alpha_k\eta_k)$ where $\alpha_k > 0$ is computed from a line search procedure to satisfy the Wolfe conditions
        $$f(x_{k+1}) \leq f(x_k) + c_1\alpha_k g(\operatorname{grad} f(x_k), \eta_k), \tag{4.2.1}$$
        $$\frac{d}{dt} f(R(t\eta_k))|_{t=\alpha_k} \geq c_2 \frac{d}{dt} f(R(t\eta_k)|_{t=0}; \tag{4.2.2}$$
    Set $x_{k+1} = R_{x_k}(\alpha_k\eta_k)$;
    Define $s_k = \mathcal{T}_{S_{\alpha_k\eta_k}}\alpha_k\eta_k$ and $y_k = \beta_k^{-1}\operatorname{grad} f(x_{k+1}) - \mathcal{T}_{S_{\alpha_k\eta_k}}\operatorname{grad} f(x_k)$, where $\beta_k = \frac{\|\alpha_k\eta_k\|}{\|\mathcal{T}_{R_{\alpha_k\eta_k}}\alpha_k\eta_k\|}$ and $\mathcal{T}_R$ is the differentiated retraction of $R$.
    Define the linear operator $\mathcal{B}_{k+1} : \mathrm{T}_{x_{k+1}}\mathcal{M} \to \mathrm{T}_{x_{k+1}}\mathcal{M}$ by
        $$\mathcal{B}_{k+1}p = \tilde{\mathcal{B}}_k p - \frac{g(s_k, \tilde{\mathcal{B}}_k p)}{g(s_k, \tilde{\mathcal{B}}_k s_k)}\tilde{\mathcal{B}}_k s_k + \frac{g(y_k, p)}{g(y_k, s_k)} y_k + \phi_k g(s_k, \tilde{\mathcal{B}}_k s_k) g(v_k, p) v_k,$$
    for all $p \in \mathrm{T}_{x_{k+1}}\mathcal{M}$ or equivalently
        $$\mathcal{B}_{k+1} = \tilde{\mathcal{B}}_k - \frac{\tilde{\mathcal{B}}_k s_k (\tilde{\mathcal{B}}_k^* s_k)^\flat}{(\tilde{\mathcal{B}}_k^* s_k)^\flat s_k} + \frac{y_k y_k^\flat}{y_k^\flat s_k} + \phi_k g(s_k, \tilde{\mathcal{B}}_k s_k) v_k v_k^\flat, \tag{4.2.3}$$
    where
        $$v_k = \frac{y_k}{g(y_k, s_k)} - \frac{\tilde{\mathcal{B}}_k s_k}{g(s_k, \tilde{\mathcal{B}}_k s_k)},$$
    $\phi_k$ is any number in the open interval $(\phi_k^c, \infty)$, $\phi_k^c = 1/(1 - \mu_k)$, $\mu_k = (g(y_k, \tilde{\mathcal{B}}_k^{-1} y_k) g(s_k, \tilde{\mathcal{B}}_k s_k))/g(y_k, s_k)^2$, $\tilde{\mathcal{B}}_k = \mathcal{T}_{S_{\alpha_k\eta_k}} \circ \mathcal{B}_k \circ \mathcal{T}_{S_{\alpha_k\eta_k}}^{-1}$, $*$ denotes the adjoint with respect to $g$ [1].
    $k \leftarrow k + 1$;
end while
```

$\bar{x} \in \mathcal{M}$, there exists a neighborhood $\mathcal{U}$ of $\bar{x}$ and a constant c_0 such that for all $x, y \in \mathcal{U}$

$$\|\mathcal{T}_{S_\eta} - \mathcal{T}_{R_\eta}\| \leq c_0 \|\eta\| \tag{4.2.4}$$

$$\|\mathcal{T}_{S_\eta}^{-1} - \mathcal{T}_{R_\eta}^{-1}\| \leq c_0 \|\eta\| \tag{4.2.5}$$

where $\eta = R_x^{-1}(y)$. In the following analysis, we use only these two properties of isometric vector transport.

In Algorithm 3, we require the isometric vector transport $\mathcal{T}_S$ to satisfy the *locking condition*

$$\mathcal{T}_{S_\xi} \xi = \beta \mathcal{T}_{R_\xi} \xi, \quad \beta = \frac{\|\xi\|}{\|\mathcal{T}_{R_\xi} \xi\|}, \tag{4.2.6}$$

for all $\xi \in \mathrm{T}_x \mathcal{M}$ and all $x \in \mathcal{M}$. Practical ways of building such a $\mathcal{T}_S$ given a retraction and vice versa are discussed in Section 4.4. Observe that, throughout Algorithm 3, the differentiated retraction $\mathcal{T}_R$ only appears in the form $\mathcal{T}_{R_\xi} \xi$, which is equal to $\frac{d}{dt} R(t\xi)|_{t=1}$. Hence $\mathcal{T}_{R_{\alpha_k \eta_k}} \alpha_k \eta_k$ is just the velocity vector of the line search curve $\alpha \mapsto R(\alpha \eta_k)$ at time α_k and we are only required to be able to evaluate the differentiated retraction in the direction transported. The computational efficiency that results is also discussed in Section 4.4.

The isometry condition (1.2.1) and the locking condition (4.2.6) are imposed on $\mathcal{T}_S$ notably because, as shown in Lemma 4.2.1, they ensure that the second Wolfe condition (4.2.2) implies $g(s_k, y_k) > 0$. Much as in the Euclidean case, it is essential that $g(s_k, y_k) > 0$, otherwise the secant condition $\mathcal{B}_{k+1} s_k = y_k$ cannot hold with $\mathcal{B}_{k+1}$ positive definite, whereas positive definiteness of the $\mathcal{B}_k$'s is key to guarantee that the search directions η_k are descent directions. It is possible to state Algorithm 3 without imposing the isometry and locking conditions, but then it becomes an open question whether the main convergence results would still hold. Clearly, some intermediate results would fail to hold and, assuming that the main results still hold, a completely different approach would probably be required to prove them.

When $\phi = 0$, the updating formula (4.2.3) reduces to the Riemannian BFGS formula of [QGA10]. However, a crucial difference between Algorithm 3 and the Riemannian BFGS of [QGA10] lies in the definition of y_k. Its definition in [QGA10] corresponds to setting β_k to 1 instead of $\frac{\|\alpha_k \eta_k\|}{\|\mathcal{T}_{R_{\alpha_k \eta_k}} \alpha_k \eta_k\|}$. Our choice of β_k allows for a convergence analysis under more general assumptions than those of the convergence analysis of Qi [Qi11]. Indeed, the convergence analysis of the Riemannian BFGS of [QGA10], found in [Qi11], assumes that retraction R is set to the exponential mapping and that vector transport $\mathcal{T}_S$ is set to the parallel translation. These specific choices remain legitimate in Algorithm 3, hence the convergence analysis given here subsumes the one in [Qi11]; however, several other choices become possible, as discussed in more detail in Section 4.4.

Lemma 4.2.1 proves that Algorithm 3 is well-defined for $\phi_k \in (\phi_k^c, \infty)$.

Lemma 4.2.1. *Algorithm 3 constructs infinite sequences* $\{x_k\}$, $\{\mathcal{B}_k\}$, $\{\tilde{\mathcal{B}}_k\}$, $\{\alpha_k\}$, $\{s_k\}$, *and* $\{y_k\}$, *unless the stopping criterion in Step 2 is satisfied at some iteration. For all* k, *the Hessian approximation* $\mathcal{B}_k$ *is symmetric positive definite with respect to metric* g, $\eta_k \neq 0$, *and*

$$g(s_k, y_k) \geq (c_2 - 1)\alpha_k g(\operatorname{grad} f(x_k), \eta_k). \tag{4.2.7}$$

Proof. We first show that (4.2.7) holds when all the involved quantities exist and $\eta_k \neq 0$. Define $\tilde{m}_k(t) = f(R_{x_k}(t\eta_k/\|\eta_k\|))$. We have

$$\begin{aligned}
g(s_k, y_k) &= g(\mathcal{T}_{S_{\alpha_k\eta_k}}\alpha_k\eta_k, \beta_k^{-1}\operatorname{grad} f(x_{k+1}) - \mathcal{T}_{S_{\alpha_k\eta_k}}\operatorname{grad} f(x_k))\\
&= g(\mathcal{T}_{S_{\alpha_k\eta_k}}\alpha_k\eta_k, \beta_k^{-1}\operatorname{grad} f(x_{k+1})) - g(\mathcal{T}_{S_{\alpha_k\eta_k}}\alpha_k\eta_k, \mathcal{T}_{S_{\alpha_k\eta_k}}\operatorname{grad} f(x_k))\\
&= g(\beta_k^{-1}\mathcal{T}_{S_{\alpha_k\eta_k}}\alpha_k\eta_k, \operatorname{grad} f(x_{k+1})) - g(\alpha_k\eta_k, \operatorname{grad} f(x_k)) \text{ (by isometry)}\\
&= g(\mathcal{T}_{R_{\alpha_k\eta_k}}\alpha_k\eta_k, \operatorname{grad} f(x_{k+1})) - g(\alpha_k\eta_k, \operatorname{grad} f(x_k)) \text{ (by (4.2.6))}\\
&= \alpha_k\|\eta_k\|\left(\frac{d\tilde{m}_k(\alpha_k\|\eta_k\|)}{dt} - \frac{d\tilde{m}_k(0)}{dt}\right).
\end{aligned} \tag{4.2.8}$$

Note that guaranteeing (4.2.8), which will be used frequently, is the key reason for imposing the locking condition (4.2.6). From the second Wolfe condition (4.2.2), we have

$$\frac{d\tilde{m}_k(\alpha_k\|\eta_k\|)}{dt} \geq c_2\frac{d\tilde{m}_k(0)}{dt}. \tag{4.2.9}$$

Therefore,

$$\frac{d\tilde{m}_k(\alpha_k\|\eta_k\|)}{dt} - \frac{d\tilde{m}_k(0)}{dt} \geq (c_2 - 1)\frac{d\tilde{m}_k(0)}{dt} = (c_2 - 1)\frac{1}{\|\eta_k\|}g(\operatorname{grad} f(x_k), \eta_k). \tag{4.2.10}$$

The claim (4.2.7) follows from (4.2.8) and (4.2.10).

When $\mathcal{B}_k$ is symmetric positive definite, η_k is a descent direction. Observe that the function $\alpha \mapsto f(R(\alpha\eta_k))$ remains a continuously differentiable function from $\mathbb{R}$ to $\mathbb{R}$ which is bounded below. Therefore, the classical result in [NW06, Lemma 3.1] guarantees the existence of a step size,α_k, that satisfies the Wolfe conditions.

The claims are proved by induction. They hold for $k = 0$ in view of the assumptions on $\mathcal{B}_0$ of Step 3 and of the results above. Assume now that the claims hold for some k. From (4.2.7), we have

$$g(s_k, y_k) \geq (1 - c_2)\alpha_k g(\operatorname{grad} f(x_k), -\eta_k) = (1 - c_2)\alpha_k g(\operatorname{grad} f(x_k), \mathcal{B}_k^{-1} \operatorname{grad} f(x_k)) > 0.$$

Recall that, in the Euclidean case, $s_k^T y_k > 0$ is a necessary and sufficient condition for the existence of a positive-definite secant update (see [DS83, Lemma 9.2.1]), and that BFGS is such an update [DS83, (9.2.10)]. From the generalization of these results to the Riemannian case (see [Qi11, Lemmas 2.4.1 and 2.4.2]), it follows that $\mathcal{B}_{k+1}$ is symmetric positive definite when $\phi_k \equiv 0$.

Consider the function $h(\phi_k) : \mathbb{R} \to \mathbb{R}^d$ which gives the eigenvalues of $\mathcal{B}_{k+1}$. Since $\mathcal{B}_{k+1}$ is symmetric positive definite when $\phi_k \equiv 0$, we know all entries of $h(0)$ are greater than 0. By calculations similar to those for the Euclidean case ([BLN92]), we have

$$\det(\mathcal{B}_{k+1}) = \det(\mathcal{B}_k)\frac{g(y_k, s_k)}{g(s_k, \mathcal{B}_k s_k)}(1 + \phi_k(u_k - 1)).$$

So $\det(\mathcal{B}_{k+1}) = 0$ if and only if $\phi_k = \phi_k^c < 0$. In other words, $h(\phi_k)$ has one or more 0 entries if and only if $\phi_k = \phi_k^c$. In addition, since all entries of $h(0)$ are greater than 0 and $h(\phi_k)$ is a continuous function, we have that all entries of $h(\phi_k)$ are greater than 0 if and only if $\phi_k > \phi_k^c$. Therefore, the operator $\mathcal{B}_{k+1}$ is positive definite when $\phi_k > \phi_k^c$. The symmetry of $\mathcal{B}_{k+1}$ is easily verified. □

4.3 Global Convergence Analysis

In this section, global convergence is proven under a generalized convexity assumption and for $\phi_k \in [0, 1 - \delta]$, where δ is any number in $(0, 1]$. The behavior of the Riemannian Broyden methods with ϕ_k not necessarily in this interval is explored in the experiments. Note the result derived in this section also guarantees local convergence to an isolated local minimizer.

4.3.1 Basic Assumptions and Definitions

Throughout the convergence analysis, $\{x_k\}$, $\{\mathcal{B}_k\}$, $\{\tilde{\mathcal{B}}_k\}$, $\{\alpha_k\}$, $\{s_k\}$, $\{y_k\}$, and $\{\eta_k\}$, are infinite sequences generated by Algorithm 3, Ω denotes the sublevel set $\{x : f(x) \leq f(x_0)\}$, and x^* is a local minimizer of f in the level set Ω. The existence of such an x^* is guaranteed if Ω is compact, which happens, in particular, whenever the manifold $\mathcal{M}$ is compact.

The convergence analysis depends on the property of (strong) retraction-convexity formalized in Definition 4.3.1 and the following two additional assumptions.

Definition 4.3.1. *For a function $f : \mathcal{M} \to \mathbb{R} : x \mapsto f(x)$ on a Riemannian manifold $\mathcal{M}$ with retraction R define $\tilde{m}_{x,\eta}(t) = f(R_x(t\eta))$ for $x \in \mathcal{M}$ and $\eta \in \mathrm{T}_x \mathcal{M}$. The function f is retraction-convex with respect to the retraction R in a set $\mathcal{S}$ if for all $x \in \mathcal{S}$, all $\eta \in \mathrm{T}_x \mathcal{M}$ and $\|\eta\| = 1$, $\tilde{m}_{x,\eta}(t)$ is convex for all t which satisfies $R_x(t\eta) \in \mathcal{S}$. Moreover, f is strongly retraction-convex in $\mathcal{S}$ if $\tilde{m}_{x,\eta}(t)$ is strongly convex for all $x \in \mathcal{S}$ and all $\|\eta\| = 1$ such that $R_x(\eta) \in \mathcal{S}$.*

Assumption 4.3.1. *The objective function f is twice continuously differentiable.*

Assumption 4.3.2. *There exist $r > 0$ and $\rho > 0$ such that, for each $y \in R_{x^*}(\mathbb{B}(0_{x^*}, r)) =: \tilde{\Omega}$, we have $R_y(\mathbb{B}(0_y, \rho)) \supset \tilde{\Omega}$ and $R_y(\cdot)$ is a diffeomorphism on $\mathbb{B}(0_y, \rho)$. The iterates x_k stay continuously in $\tilde{\Omega}$, meaning that $R_{x_k}(t\eta_k) \in \tilde{\Omega}$ for all $t \in [0, \alpha_k]$. Moreover, f is strongly retraction-convex with respect to the retraction R in the closure of $\tilde{\Omega}$.*

Lemma 4.3.1. *If Assumption 4.3.1 holds, then f is retraction-convex with respect to the exponential mapping in an open set $\mathcal{S}$ if and only if $\operatorname{Hess} f(x)$ is positive definite for all $x \in \mathcal{S}$. Moreover, f is strongly retraction-convex with respect to the exponential mapping in an open set $\mathcal{S}$ if and only if there exists a constant $a_0 > 0$ such that $\operatorname{Hess} f(x) - a_0 \operatorname{id}$ is positive definite for all $x \in \mathcal{S}$.*

Proof. First, suppose f is retraction-convex with respect to the exponential mapping in a set $\mathcal{S}$. Therefore, $\frac{d^2 \tilde{m}_{x,\eta}(0)}{dt^2} \geq 0$ and since $\frac{d^2 \tilde{m}_{x,\eta}(0)}{dt^2} = g(\operatorname{Hess} f(x)\eta, \eta)$, we have that

$$g(\operatorname{Hess} f(x)\eta, \eta) \geq 0,$$

for all $\eta \in \mathrm{T}_x \mathcal{M}$. Thus $\operatorname{Hess} f(x)$ is positive definite.

Second, suppose f is strongly retraction-convex with respect to the exponential mapping in a set $\mathcal{S}$. By the same idea, we have

$$\frac{d^2 \tilde{m}_{x,\eta}(0)}{dt^2} - a_0 \geq 0.$$

Since by definition we have $g((\operatorname{Hess} f(x) - a_0 \operatorname{id})\eta, \eta) = g(\operatorname{Hess} f(x)\eta, \eta) - a_0$ and when using the exponential mapping $\frac{d^2 \tilde{m}_{x,\eta}(0)}{dt^2} = g(\operatorname{Hess} f(x)\eta, \eta)$ it follows that

$$g((\operatorname{Hess} f(x) - a_0 \operatorname{id})\eta, \eta) \geq 0$$

and $\operatorname{Hess} f(x) - a_0 \operatorname{id}$ is positive definite.

Conversely, suppose $\operatorname{Hess} f(x)$ is positive definite for all $x \in \mathcal{S}$. Note that for the exponential mapping, we have

$$\operatorname{Exp}_x(t\eta) = \operatorname{Exp}_y(P_\gamma^{1\leftarrow 0}((t - t_0)\eta)),$$

where $x \in \mathcal{M}$, $\eta \in \operatorname{T}_x \mathcal{M}$, $y = \operatorname{Exp}_x(t_0\eta)$, γ is the geodesic from x to y such that $\gamma(0) = x$ and $\gamma(1) = y$, and $P_\gamma^{1\leftarrow 0}\mu$ denotes parallel translation of $\mu \in \operatorname{T}_x \mathcal{M}$ along the geodesic. The curve on the manifold defined by

$$\tilde{m}_{x,\eta}(t) = f(\operatorname{Exp}_x(t\eta)) = f(\operatorname{Exp}_y(P_\gamma^{1\leftarrow 0}((t - t_0)\eta))) = \tilde{m}_{y,P_\gamma^{1\leftarrow 0}(\eta)}(t - t_0)$$

satisfies

$$\frac{d^2 \tilde{m}_{x,\eta}(t)}{dt^2}|_{t=t_0} = \frac{d^2 \tilde{m}_{y,P_\gamma^{1\leftarrow 0}(\eta)}(t)}{dt^2}|_{t=0} = g(\operatorname{Hess} f(y) P_\gamma^{1\leftarrow 0}(\eta), P_\gamma^{1\leftarrow 0}(\eta)) \geq 0.$$

Since t_0 can be arbitrary such that $R_x(t_0\eta) \in \mathcal{S}$, the proof of retraction-convexity is complete.

Furthermore, if $\operatorname{Hess} f(x) - a_0 \operatorname{id}$ is positive definite for all $x \in \mathcal{S}$, then

$$\frac{d^2 \tilde{m}_{x,\eta}(t)}{dt^2}|_{t=t_0} - a_0 = \frac{d^2 \tilde{m}_{y,P_\gamma^{1\leftarrow 0}(\eta)}(t)}{dt^2}|_{t=0} - a_0 = g((\operatorname{Hess} f(y) - a_0 \operatorname{id}) P_\gamma^{1\leftarrow 0}(\eta), P_\gamma^{1\leftarrow 0}(\eta)) \geq 0$$

completing the proof for strong retraction-convexity. □

Lemma 4.3.2. *Suppose Assumption 4.3.1 holds and* $\operatorname{Hess} f(x^*)$ *is positive definite. Define* $\tilde{m}_{x,\eta}(t) = f(R_x(t\eta))$*, where* $\|\eta\| = 1$*. Then there exists a neighbor* $\mathcal{N}$ *of* x^* *and two constants* $0 < a_0 < a_1$ *such that*

$$a_0 \leq \frac{d^2 \tilde{m}_{x,\eta}}{dt^2}(t) \leq a_1,$$

for all $x \in \mathcal{N}$ *and* t *which satisfies* $R_x(t\eta) \in \mathcal{N}$*.*

Proof. First, the left inequality is proved. Define $u(\hat{x}, \hat{\xi}_x) = \hat{f}(R_{\hat{x}}(\hat{\xi}_x))$ where the coordinate expressions are chosen by using orthonormal vector fields. Therefore, the matrix expression G_x of the metric at x is identity. Since x^* is a critical point, the Euclidean $\operatorname{Hess}_2 u(\hat{x}^*, 0)$ is equivalent to the Riemannian $\operatorname{Hess} f(x^*)$ that is assumed to be positive definite, where $\operatorname{Hess}_2 u(\hat{x}, \hat{\xi})$ denotes the Hessian respect to the second variable. In addition, f is assumed to be twice continuously differentiable, therefore, $\operatorname{Hess}_2 u(\hat{x}, \hat{\xi}_x)$ is continuous and there exists a neighborhood $\hat{\mathcal{N}} \times \hat{\mathcal{V}}$ of $(\hat{x}^*, 0)$ such that the smallest eigenvalue of $\operatorname{Hess}_2 u(\hat{x}, \hat{\eta}_x)$ is uniformly greater than some positive number a_0. By Lemmas

3.3.1 and 3.3.3, $\hat{\mathcal{N}}$ can be made small enough so that $\hat{\zeta} \in \hat{\mathcal{V}}$ where $\zeta = R_{x_1}^{-1}(x_2)$ for any $\hat{x}_1, \hat{x}_2 \in \hat{\mathcal{N}}$. Let $\mathcal{N}$ denote the non-coordinate expression of $\hat{\mathcal{N}}$ and let $\tilde{m}_{x,\eta}(t)$ denote $f(R_x(t\eta)) = u(\hat{x}, t\hat{\eta})$. By computing the second derivative for $\tilde{m}_{x,\eta}(t)$ and noticing $\|\hat{\eta}\|_2 = \|G_x^{1/2}\hat{\eta}\|_2 = \|\eta\| = 1$, we have

$$\frac{d^2\tilde{m}_{x,\eta}}{dt^2}(t) = \frac{d^2u(\hat{x}, t\hat{\eta})}{dt^2}(t) = \hat{\eta}^T \operatorname{Hess}_2 u(\hat{x}, t\hat{\eta})\hat{\eta} \geq a_0,$$

for all t such that $R_x(t\eta) \in \mathcal{N}$.

To prove the right inequality, we note that $\hat{\mathcal{N}}$ can be chosen to be bounded and then $\mathcal{N}$ is also bounded. Therefore, the closure of $\mathcal{N}$, $\bar{\mathcal{N}}$, is compact and the largest eigenvalue of $\operatorname{Hess}_2 u(\hat{x}, \hat{\xi})$ for all $x \in \bar{\mathcal{N}}$ and $R_x(\xi) \in \bar{\mathcal{N}}$ is bounded by a number a_1. Thus,

$$\frac{d^2\tilde{m}_{x,\eta}}{dt^2}(t) = \hat{\eta}^T \operatorname{Hess}_2 u(\hat{x}, t\hat{\eta})\hat{\eta} \leq a_1,$$

for all t such that $R_x(t\eta) \in \mathcal{N}$. □

4.3.2 Preliminary Lemmas

The lemmas in this section provide the results needed to show global convergence stated in Theorem 4.3.1. The strategy generalizes that for the Euclidean case in [BNY87]. Where appropriate, comments are included indicating important adaptations of the reasoning to Riemannian manifolds. The two main difficulties are the lack of Taylor's Theorem for a function defined on a Riemannian manifold and the use in some Euclidean proofs of an average Hessian on a straight line that generalizes only to lines defined by the exponential mapping and parallel translation and not to lines defined by retractions and vector transports.

The first result, Lemma 4.3.3, is used to prove Lemma 4.3.5.

Lemma 4.3.3. *If Assumptions 4.3.1 and 4.3.2 hold then*

$$\frac{1}{2}a_0\|s_k\|^2 \leq (c_1 - 1)\alpha_k g(\operatorname{grad} f(x_k), \eta_k). \tag{4.3.1}$$

Proof. In Euclidean space, Taylor's Theorem is used to characterize a function around a point. However, there is no Taylor's Theorem for a function f on Riemannian manifold due to the lack of addition. This difficulty is overcome by defining a function on a curve on the manifold and applying Taylor's Theorem. Define $\tilde{m}_k(t) = f(R_{x_k}(t\eta_k/\|\eta_k\|))$. Since $f \in C^2$ is strongly retraction-convex on a compact set, there exist constants $0 < a_0 < a_1$ such that

$$a_0 \leq \frac{d^2\tilde{m}_{x,\eta}(t)}{dt^2} \leq a_1.$$

From Taylor's theorem, we know

$$\begin{aligned} f(x_{k+1}) - f(x_k) = \tilde{m}_k(\alpha_k\|\eta_k\|) - \tilde{m}_k(0) &= \frac{d\tilde{m}_k(0)}{dt}\alpha_k\|\eta_k\| + \frac{1}{2}\frac{d^2\tilde{m}_k(p)}{dt^2}(\alpha_k\|\eta_k\|)^2 \\ &= g(\operatorname{grad} f(x_k), \alpha_k\eta_k) + \frac{1}{2}\frac{d^2\tilde{m}_k(p)}{dt^2}(\alpha_k\|\eta_k\|)^2 \\ &\geq g(\operatorname{grad} f(x_k), \alpha_k\eta_k) + \frac{1}{2}a_0(\alpha_k\|\eta_k\|)^2, \end{aligned} \tag{4.3.2}$$

where $0 \leq p \leq \alpha_k\|\eta_k\|$. Using (4.3.2), the first Wolfe condition (4.2.1) and that $\|s_k\| = \alpha_k\|\eta_k\|$, we obtain

$$(c_1 - 1)g(\operatorname{grad} f(x_k), \alpha_k\eta_k) \geq \frac{1}{2}a_0\|s_k\|^2$$

completing the proof. □

Lemma 4.3.4 generalizes [BNY87, (2.4)].

Lemma 4.3.4. *If Assumptions 4.3.1 and 4.3.2 hold then there exists two constants* $0 < a_0 \leq a_1$ *such that*

$$a_0 g(s_k, s_k) \leq g(s_k, y_k) \leq a_1 g(s_k, s_k), \tag{4.3.3}$$

for all k.

Proof. In the Euclidean case of [BNY87, (2.4)], the proof follows easily from the convexity of the cost function and the resulting positive definiteness of the Hessian over the entire relevant set. The Euclidean proof exploits the relationship $y_k = \bar{G}_k s_k$, where $\bar{G}_k$ is the average Hessian and that $\bar{G}_k$ must be positive definite to bound the inner product $s_k^T y_k$ using the largest and smallest eigenvalues that can in turn be bounded on the relevant set. We do not have this property on a Riemannian manifold but the locking condition, retraction-convexity and replacing the average Hessian with a quantity derived from a function defined on a curve on the manifold allows the generalization.

Define $\tilde{m}_k(t) = f(R_{x_k}(t\eta_k/\|\eta_k\|))$. Using the locking condition (4.2.8) and Taylor's Theorem yields

$$g(s_k, y_k) = \alpha_k\|\eta_k\|(\frac{d\tilde{m}(\alpha_k\|\eta_k\|)}{dt} - \frac{d\tilde{m}(0)}{dt}) = \alpha_k\|\eta_k\| \int_0^{\alpha_k\|\eta_k\|} \frac{d^2\tilde{m}}{dt^2}(s)ds$$

and since $g(s_k, s_k) = \alpha_k^2\|\eta_k\|^2$, we have

$$\frac{g(s_k, y_k)}{g(s_k, s_k)} = \frac{1}{\alpha_k\|\eta_k\|} \int_0^{\alpha_k\|\eta_k\|} \frac{d^2\tilde{m}}{dt^2}(s)ds.$$

By Assumption 4.3.2, it follows that

$$a_0 \leq \frac{g(s_k, y_k)}{g(s_k, s_k)} \leq a_1.$$

□

Lemma 4.3.5 generalizes [BNY87, Lemma 2.1].

Lemma 4.3.5. *Suppose Assumptions 4.3.1 and 4.3.2 hold. Then there exist two constants* $0 < a_2 < a_3$ *such that*

$$a_2 \| \operatorname{grad} f(x_k) \| \cos\theta_k \leq \|s_k\| \leq a_3 \| \operatorname{grad} f(x_k) \| \cos\theta_k, \tag{4.3.4}$$

for all k*, where* $\cos\theta_k = \frac{-g(\operatorname{grad} f(x_k), \eta_k)}{\| \operatorname{grad} f(x_k) \| \|\eta_k\|}$.

Proof. Define $\tilde{m}_k(t) = f(R_{x_k}(t\eta_k / \|\eta_k\|))$. By (4.2.7) of Lemma 4.2.1, we have

$$g(s_k, y_k) \geq \alpha_k (c_2 - 1) g(\operatorname{grad} f(x_k), \eta_k) = \alpha_k (1 - c_2) \| \operatorname{grad} f(x_k) \| \|\eta_k\| \cos\theta_k.$$

Using (4.3.3) and noticing $\|\alpha_k \eta_k\| = \|s_k\|$, we know

$$\|s_k\| \geq a_2 \| \operatorname{grad} f(x_k) \| \cos\theta_k$$

where $a_2 = (1 - c_2)/a_1$ proving the left inequality.

By (4.3.1) of Lemma 4.3.3, we have

$$(c_1 - 1) g(\operatorname{grad} f(x_k), \alpha_k \eta_k) \geq \frac{1}{2} a_0 \|s_k\|^2.$$

Noting that $\|s_k\| = \alpha_k \|\eta_k\|$ and by the definition of $\cos\theta_k$, we have

$$\|s_k\| \leq a_3 \| \operatorname{grad} f(x_k) \| \cos\theta_k,$$

where $a_3 = 2(1 - c_1)/a_0$. □

Lemma 4.3.6 is needed to prove Lemmas 4.3.7 and 4.3.10. Lemma 4.3.6 gives a Lipschitz-like relationship between two related vector transports applied to the same tangent vector.

Lemma 4.3.6. *Let* $\mathcal{M}$ *be a Riemannian manifold endowed with two vector transports* $\mathcal{T}_1 \in C^0$ *and* $\mathcal{T}_2 \in C^\infty$ *where* $\mathcal{T}_1$ *satisfies (4.2.4) and (4.2.5) and both transports are associated with a same* *retraction R. Then for any $\bar{x} \in \mathcal{M}$ there exists a constant $a_4 > 0$ and a neighborhood of $\bar{x}$, $\mathcal{U}$, such that for all $x, y \in \mathcal{U}$*

$$\|\mathcal{T}_{1_\eta}\xi - \mathcal{T}_{2_\eta}\xi\| \leq a_4 \|\xi\| \|\eta\|,$$

where $\eta = R_x^{-1} y$ and $\xi \in T_x$.

Proof. $L_R(\hat{x}, \hat{\eta})$ and $L_2(\hat{x}, \hat{\eta})$ denote coordinate form of $\mathcal{T}_{R_\eta}$ and $\mathcal{T}_{2_\eta}$ respectively. We have

$$\begin{aligned}
\|\mathcal{T}_{1_\eta}\xi - \mathcal{T}_{2_\eta}\xi\| &= \|\mathcal{T}_{1_\eta}\xi - \mathcal{T}_{R_\eta}\xi + \mathcal{T}_{R_\eta}\xi - \mathcal{T}_{2_\eta}\xi\| \\
&\leq b_0 \|\eta\| \|\xi\| + \|\mathcal{T}_{R_\eta}\xi - \mathcal{T}_{2_\eta}\xi\| \\
&\leq b_0 \|\eta\| \|\xi\| + b_1 \|(L_R(\hat{x}, \hat{\eta}) - L_2(\hat{x}, \hat{\eta}))\hat{\xi}\|_2 \\
&\leq b_0 \|\eta\| \|\xi\| + b_1 \|\hat{\xi}\|_2 \|L_R(\hat{x}, \hat{\eta}) - L_2(\hat{x}, \hat{\eta})\|_2 \\
&\leq b_0 \|\eta\| \|\xi\| + b_2 \|\hat{\xi}\|_2 \|\hat{\eta}\|_2 \text{ (since } L_R(\hat{x}, 0) = L_2(\hat{x}, 0)) \\
&= b_3 \|\xi\| \|\eta\|
\end{aligned}$$

where b_0, b_1, b_2, b_3 are positive constants. □

Lemma 4.3.7 is a consequence of Lemma 4.3.6.

Lemma 4.3.7. *Let $\mathcal{M}$ be a Riemannian manifold endowed with a retraction R whose differentiated retraction is denoted $\mathcal{T}_R$. Let $\bar{x} \in \mathcal{M}$. Then there is a neighborhood $\mathcal{U}$ of $\bar{x}$ and constant $\tilde{a}_4 > 0$ such that for all $x, y \in \mathcal{U}$, any $\xi \in \mathrm{T}_x \mathcal{M}$ with $\|\xi\| = 1$, the effect of the differentiated retraction is bounded with*

$$|\|\mathcal{T}_{R_\eta}\xi\| - 1| \leq \tilde{a}_4 \|\eta\|,$$

where $\eta = R_x^{-1} y$.

Proof. Applying Lemma 4.3.6 with $\mathcal{T}_1 = \mathcal{T}_R$ and $\mathcal{T}_2$ be isometric, we have

$$\|\mathcal{T}_{R_\eta}\xi - \mathcal{T}_{2_\eta}\xi\| \leq b_0 \|\xi\| \|\eta\|,$$

where b_0 is a positive constant. Noticing $\|\xi\| = 1$ and $\| \cdot \|$ is the induced norm, we have

$$b_0 \|\eta\| \geq \|\mathcal{T}_{R_\eta}\xi - \mathcal{T}_{2_\eta}\xi\| \geq \|\mathcal{T}_{R_\eta}\xi\| - \|\mathcal{T}_{2_\eta}\xi\| = \|\mathcal{T}_{R_\eta}\xi\| - 1.$$

Similarly, we have

$$b_0 \|\eta\| \geq \|\mathcal{T}_{2_\eta}\xi - \mathcal{T}_{R_\eta}\xi\| \geq \|\mathcal{T}_{2_\eta}\xi\| - \|\mathcal{T}_{R_\eta}\xi\| = 1 - \|\mathcal{T}_{R_\eta}\xi\|.$$

to complete the proof. □

Lemma 4.3.8 generalizes [BNY87, (2.13)] and implies a generalization of the Zoutendijk Condition [NW06, Theorem 3.2], i.e., if $\cos\theta_k$ does not approach 0, then according to this lemma, the algorithm is convergent.

Lemma 4.3.8. *Suppose Assumptions 4.3.1 and 4.3.2 hold. Then there exists a constant* $a_5 > 0$ *such that for all* k

$$f(x_{k+1}) - f(x^*) \leq (1 - a_5 \cos^2\theta_k)(f(x_k) - f(x^*)),$$

where $\cos\theta_k = \frac{-g(\operatorname{grad} f(x_k), \eta_k)}{\|\operatorname{grad} f(x_k)\| \|\eta_k\|}$.

Proof. The original proof in [BNY87, (2.13)] uses the average Hessian. As when proving Lemma 4.3.3, this is replaced by considering a function defined on a curve on the manifold. Let $z_k = \|R_{x^*}^{-1} x_k\|$ and $\zeta_k = (R_{x^*}^{-1} x_k)/z_k$. Define $m_k(t) = f(R_{x^*}(t\zeta_k))$. From Taylor's Theorem, we have

$$m_k(0) - m_k(z_k) = \frac{dm_k(z_k)}{dt}(0 - z_k) + \frac{1}{2}\frac{d^2 m_k(p)}{dt^2}(0 - z_k)^2, \tag{4.3.5}$$

where p is some number between 0 and z_k. Notice that x^* is the minimizer, so $m_k(0) - m_k(z_k) \leq 0$. According to Assumption 4.3.2, we have

$$\frac{dm_k(z_k)}{dt} \geq \frac{1}{2} a_0 z_k. \tag{4.3.6}$$

Still using (4.3.5) and noticing that $\frac{d^2 m_k(p)}{dt^2}(0 - z_k)^2 \geq 0$, we have

$$f(x_k) - f(x^*) \leq \frac{dm_k(z_k)}{dt} z_k. \tag{4.3.7}$$

Combining (4.3.6) and (4.3.7) and noticing that $\frac{dm_k(z_k)}{dt} = g(\operatorname{grad} f(x_k), \mathcal{T}_{R_{z_k\zeta_k}} \zeta_k)$, we have

$$f(x_k) - f(x^*) \leq \frac{2}{a_0} g^2(\operatorname{grad} f(x_k), \mathcal{T}_{R_{z_k\zeta_k}} \zeta_k).$$

and

$$\begin{aligned} f(x_k) - f(x^*) &\leq \frac{2}{a_0} \|\operatorname{grad} f(x_k)\|^2 \|\mathcal{T}_{R_{z_k\zeta_k}} \zeta_k\|^2 \\ &\leq b_0 \|\operatorname{grad} f(x_k)\|^2, \text{ (by Lemma 4.3.7)} \end{aligned} \tag{4.3.8}$$

where b_0 is a positive constant. Using (4.3.4), the first Wolfe condition (4.2.1) and the definition of $\cos\theta_k$, we obtain

$$f(x_{k+1}) - f(x_k) \leq -b_1 \|\operatorname{grad} f(x_k)\|^2 \cos^2\theta_k,$$

where b_1 is some positive constant. Using (4.3.8), we obtain

$$f(x_{k+1}) - f(x^*) \leq (1 - a_5 \cos^2 \theta_k)(f(x_k) - f(x^*)).$$

where $a_5 = b_1/b_0$ is a positive constant. □

Lemma 4.3.9 generalizes [BNY87, Lemma 2.2].

Lemma 4.3.9. *Suppose Assumptions 4.3.1 and 4.3.2 hold. Then there exist two constants* $0 < a_6 < a_7$ *such that*

$$a_6 \frac{g(s_k, \tilde{\mathcal{B}}_k s_k)}{\|s_k\|^2} \leq \alpha_k \leq a_7 \frac{g(s_k, \tilde{\mathcal{B}}_k s_k)}{\|s_k\|^2}, \tag{4.3.9}$$

for all k.

Proof. Note that this proof does not depend on the use of the average Hessian as in original proof of [BNY87, Lemma 2.2] since Lemma 4.3.3 is applied. We have

$$\begin{aligned}
(1 - c_2) g(s_k, \tilde{\mathcal{B}}_k s_k) &= (1 - c_2) g(\alpha_k \eta_k, \alpha_k \mathcal{B}_k \eta_k) \\
&= (1 - c_2) g(\alpha_k \eta_k, \alpha_k \mathcal{B}_k \eta_k) \\
&= (1 - c_2) g(\alpha_k \eta_k, \alpha_k \mathcal{B}_k \eta_k) \\
&= (c_2 - 1) \alpha_k^2 g(\eta_k, \operatorname{grad} f(x_k)) \\
&\leq \alpha_k g(s_k, y_k) \text{ (by (4.2.7) of Lemma 4.2.1)} \\
&\leq \alpha_k a_1 \|s_k\|^2 \text{ (by (4.3.3))}.
\end{aligned}$$

Therefore,

$$\alpha_k \geq a_6 \frac{g(s_k, \tilde{\mathcal{B}}_k s_k)}{\|s_k\|^2},$$

where $a_6 = (1 - c_2)/a_1$ giving the left inequality.

By (4.3.1) of Lemma 4.3.3 and since $\|s_k\| = \alpha_k \|\eta_k\|$, we have

$$(c_1 - 1) \alpha_k g(\operatorname{grad} f(x_k), \eta_k) \geq \frac{1}{2} a_0 \|s_k\|^2.$$

By Step 3 of Algorithm 3,

$$\begin{aligned}
(c_1 - 1) \alpha_k g(\operatorname{grad} f(x_k), \eta_k) &= (1 - c_1) g(\mathcal{B}_k \eta_k, \alpha_k \eta_k) \\
&= \frac{1 - c_1}{\alpha_k} g(s_k, \tilde{\mathcal{B}}_k s_k).
\end{aligned}$$

Therefore,

$$\alpha_k \leq a_7 \frac{g(s_k, \tilde{\mathcal{B}}_k s_k)}{\|s_k\|^2},$$

where $a_7 = (1 - c_1)/a_0$ giving the right inequality. □

Lemma 4.3.10 generalizes [BNY87, Lemma 3.1, Equation (3.3)].

Lemma 4.3.10. *Suppose Assumption 4.3.1 holds. Then, for all k there exists a constant $a_9 > 0$ such that*

$$g(y_k, y_k) \leq a_9 g(s_k, y_k). \tag{4.3.10}$$

Proof. Define $y_k^{\mathrm{P}} = \operatorname{grad} f(x_{k+1}) - P_{\gamma_k}^{1 \leftarrow 0} \operatorname{grad} f(x_k)$, where P is parallel translation and $\gamma_k(t) = R_{x_k}(t\alpha_k\eta_k)$, i.e., the retraction line from x_k to x_{k+1}. From Lemma 3.3.8, we have

$$\|P_{\gamma_k}^{0 \leftarrow 1} y_k^{\mathrm{P}} - \bar{H}_k \alpha_k \eta_k\| \leq b_0 \|\alpha_k \eta_k\|^2 = b_0 \|s_k\|^2,$$

where $\bar{H}_k = \int_0^1 P_{\gamma_k}^{0 \leftarrow t} \operatorname{Hess} f(\gamma_k(t)) P_{\gamma_k}^{t \leftarrow 0} dt$ and $b_0 > 0$. It follows that

$$\begin{aligned}
\|y_k\| &\leq \|y_k - y_k^{\mathrm{P}}\| + \|y_k^{\mathrm{P}}\| = \|y_k - y_k^{\mathrm{P}}\| + \|P_{\gamma_k}^{0 \leftarrow 1} y_k^{\mathrm{P}}\| \\
&= \|y_k - y_k^{\mathrm{P}}\| + \|P_{\gamma_k}^{0 \leftarrow 1} y_k^{\mathrm{P}} - \bar{H}_k \alpha_k \eta_k\| + \|\bar{H}_k \alpha_k \eta_k\| \\
&= \|\operatorname{grad} f(x_{k+1})/\beta_k - \mathcal{T}_{S_{\alpha_k \eta_k}} \operatorname{grad} f(x_k) - \operatorname{grad} f(x_{k+1}) + P_{\gamma_k}^{1 \leftarrow 0} \operatorname{grad} f(x_k)\| \\
&\quad + \|\bar{H}_k \alpha_k \eta_k\| + b_0 \|s_k\|^2 \\
&\leq \|\operatorname{grad} f(x_{k+1})/\beta_k - \operatorname{grad} f(x_{k+1})\| + \|P_{\gamma_k}^{1 \leftarrow 0} \operatorname{grad} f(x_k) - \mathcal{T}_{S_{\alpha_k \eta_k}} \operatorname{grad} f(x_k)\| \\
&\quad + \|\bar{H}_k \alpha_k \eta_k\| + b_0 \|s_k\|^2 \\
&\leq b_1 \|s_k\|, \text{ (by the continuity of Hessian and Lemmas 4.3.6 and 4.3.7)}
\end{aligned}$$

where $b_1 > 0$. Therefore,

$$\begin{aligned}
\frac{g(y_k, y_k)}{g(s_k, y_k)} &\leq \frac{g(y_k, y_k)}{a_0 g(s_k, s_k)} \text{ (by Lemma 4.3.4)} \\
&\leq \frac{b_1^2}{a_0}
\end{aligned}$$

giving the desired result. □

Lemma 4.3.11 generalizes [BNY87, Lemma 3.1] and as with the earlier lemmas the proof does not use an average Hessian.

Lemma 4.3.11. *Suppose Assumptions 4.3.1 and 4.3.2 hold. Then there exist constants* $a_{10} > 0$, $a_{11} > 0$, $a_{12} > 0$ *such that*

$$\frac{g(s_k, \tilde{\mathcal{B}}_k s_k)}{g(s_k, y_k)} \leq a_{10}\alpha_k \tag{4.3.11}$$

$$\frac{\|\tilde{\mathcal{B}}_k s_k\|^2}{g(s_k, \tilde{\mathcal{B}}_k s_k)} \geq a_{11}\frac{\alpha_k}{\cos^2\theta_k} \tag{4.3.12}$$

$$\frac{|g(y_k, \tilde{\mathcal{B}}_k s_k)|}{g(y_k, s_k)} \leq a_{12}\frac{\alpha_k}{\cos\theta_k} \tag{4.3.13}$$

for all k.

Proof. By (4.2.7) of Lemma 4.2.1, we have

$$g(s_k, y_k) \geq (c_2 - 1)g(\operatorname{grad} f(x_k), \alpha_k\eta_k).$$

So by the Step 3 of Algorithm 3, we obtain

$$g(s_k, y_k) \geq \frac{(1-c_2)}{\alpha_k} g(s_k, \tilde{\mathcal{B}} s_k)$$

and therefore

$$\frac{g(s_k, \tilde{\mathcal{B}}_k s_k)}{g(s_k, y_k)} \leq a_{10}\alpha_k,$$

where $a_{10} = 1/(1-c_2)$ proving (4.3.11).

Inequality (4.3.12) follows from

$$\begin{aligned} \frac{\|\tilde{\mathcal{B}}_k s_k\|^2}{g(s_k, \tilde{\mathcal{B}}_k s_k)} &= \frac{\alpha_k^2\|\operatorname{grad} f(x_k)\|^2}{\alpha_k\|s_k\|\|\operatorname{grad} f(x_k)\|\cos\theta_k} \text{ (by Step 3 of Algorithm 3 and the definition of } \cos\theta_k\text{)} \\ &= \frac{\alpha_k\|\operatorname{grad} f(x_k)\|}{\|s_k\|\cos\theta_k} \\ &\geq a_{11}\frac{\alpha_k}{\cos^2\theta_k}, \text{ (by (4.3.4))} \end{aligned}$$

where $a_{11} > 0$.

Finally, inequality (4.3.13) follows from

$$\begin{aligned} \frac{|g(y_k, \tilde{\mathcal{B}}_k s_k)|}{g(s_k, y_k)} &\leq \frac{\alpha_k\|y_k\|\|\operatorname{grad} f(x_k)\|}{g(s_k, y_k)} \text{ (by Step 3 of Algorithm 3)} \\ &\leq \frac{a_9^{1/2}\alpha_k\|\operatorname{grad} f(x_k)\|}{g^{1/2}(s_k, y_k)} \text{ (by (4.3.10))} \\ &\leq \frac{a_9^{1/2}\alpha_k\|\operatorname{grad} f(x_k)\|}{a_0^{1/2}\|s_k\|} \text{ (by (4.3.3))} \\ &\leq a_{12}\frac{\alpha_k}{\cos\theta_k}, \text{ (by (4.3.4))} \end{aligned}$$

where a_{12} is a positive constant. □

Lemma 4.3.12 generalizes [BNY87, Lemma 3.2].

Lemma 4.3.12. *Suppose Assumptions 4.3.1 and 4.3.2 hold.* $\phi_k \in [0,1]$. *Then there exists a constant* $a_{13} > 0$ *such that*

$$\prod_{j=1}^{k} \alpha_j \geq a_{13}^{k}, \tag{4.3.14}$$

for all $k \geq 1$.

Proof. The major difference between the Euclidean and Riemannian proofs is that in the Riemannian case, we have two operators $\mathcal{B}_k$ and $\tilde{\mathcal{B}}_k$ as opposed to a single operator in the Euclidean case. Once we have proven that they have the same trace and determinant, the proof unfolds similarly to the Euclidean proof. The details are given for the reader's convenience.

Use hat to denote the coordinates expression of the operators $\mathcal{B}_k$ and $\tilde{\mathcal{B}}_k$ in Algorithm 3 and consider trace($\hat{\mathcal{B}}$) and det($\hat{\mathcal{B}}$). Since they are independent of basis, we know they are well defined. Since $\mathcal{T}_S$ is an isometric vector transport, we have that $\mathcal{T}_{S_{\alpha_k \eta_k}}$ is invertible for all k, and thus

$$\begin{aligned} \operatorname{trace}(\hat{\tilde{\mathcal{B}}}_k) &= \operatorname{trace}(\hat{\mathcal{T}}_{S_{\alpha_k \eta_k}} \hat{\mathcal{B}}_k \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^{-1}) = \operatorname{trace}(\hat{\mathcal{B}}_k), \\ \det(\hat{\tilde{\mathcal{B}}}_k) &= \det(\hat{\mathcal{T}}_{S_{\alpha_k \eta_k}} \hat{\mathcal{B}}_k \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^{-1}) = \det(\hat{\mathcal{B}}_k). \end{aligned}$$

From the update formula of $\mathcal{B}_k$ in Algorithm 3, the trace of update formula is

$$\begin{aligned} \operatorname{trace}(\hat{\mathcal{B}}_{k+1}) = \operatorname{trace}(\hat{\mathcal{B}}_k) &+ \frac{\|y_k\|^2}{g(y_k, s_k)} + \phi_k \frac{\|y_k\|^2}{g(y_k, s_k)} \frac{g(s_k, \tilde{\mathcal{B}}_k s_k)}{g(y_k, s_k)} \\ &- (1 - \phi_k) \frac{\|\tilde{\mathcal{B}}_k s_k\|^2}{g(s_k, \tilde{\mathcal{B}}_k s_k)} - 2\phi_k \frac{g(y_k, \tilde{\mathcal{B}}_k s_k)}{g(y_k, s_k)}. \end{aligned} \tag{4.3.15}$$

Recall that $\phi_k g(s_k, \tilde{\mathcal{B}}_k s_k) \geq 0$. If we choose a particular basis such that the expression of the metric is the identity, then the Broyden update equation (4.2.3) is exactly the classical Broyden update equation, except that B_k is replaced by $\hat{\tilde{\mathcal{B}}}_k$, and by [BNY87, (3.9)] we have

$$\det(\hat{\mathcal{B}}_{k+1}) \geq \det(\hat{\mathcal{B}}_k) \frac{g(y_k, s_k)}{g(s_k, \tilde{\mathcal{B}}_k s_k)}. \tag{4.3.16}$$

Since det and $g(\cdot,\cdot)$ are independent of the basis, it follows that (4.3.16) holds regardless of the chosen basis. Using (4.3.10), (4.3.11), (4.3.12) and (4.3.13) for (4.3.15), we obtain

$$\operatorname{trace}(\hat{\mathcal{B}}_{k+1}) \leq \operatorname{trace}(\hat{\mathcal{B}}_k) + a_9 + \phi_k a_9 a_{10} \alpha_k - \frac{a_{11}(1 - \phi_k)\alpha_k}{\cos^2 \theta_k} + \frac{2\phi_k a_{12} \alpha_k}{\cos \theta_k} \tag{4.3.17}$$

Notice that

$$\begin{aligned}
\frac{\alpha_k}{\cos\theta_k} &= \frac{\alpha_k \| \operatorname{grad} f(x_k)\| \|\eta_k\|}{-g(\operatorname{grad} f(x_k), \eta_k)} \\
&= \frac{\alpha_k \|\tilde{\mathcal{B}}_k s_k\| \|s_k\|}{g(s_k, \tilde{\mathcal{B}}_k s_k)} \\
&= \frac{\|\tilde{\mathcal{B}}_k s_k\|}{\|s_k\|} \frac{\alpha_k \|s_k\|^2}{g(s_k, \tilde{\mathcal{B}}_k s_k)} \\
&\leq a_7 \frac{\|\tilde{\mathcal{B}}_k s_k\|}{\|s_k\|} \text{ (by (4.3.9)).} \qquad (4.3.18)
\end{aligned}$$

Since the fourth term in (4.3.17) is always negative and $\cos\theta_k \leq 1$, (4.3.18) and (4.3.17) imply that

$$\operatorname{trace}(\hat{\mathcal{B}}_{k+1}) \leq \operatorname{trace}(\hat{\mathcal{B}}_k) + a_9 + (\phi_k a_9 a_{10} + 2\phi_k a_{12} a_7) \frac{\|\tilde{\mathcal{B}}_k s_k\|}{\|s_k\|}.$$

Consider $\|\hat{\mathcal{B}}_k\|_g$, where $\|\cdot\|_g$ denotes the induce norm from the vector norm of $\|u\|_g = \sqrt{u^T G u}$. It follows that

$$\begin{aligned}
\frac{\|\tilde{\mathcal{B}}_k s_k\|}{\|s_k\|} &\leq \|\tilde{\mathcal{B}}_k\| = \|\mathcal{B}_k\| = \|\hat{\mathcal{B}}_k\|_g = \frac{\|\hat{\mathcal{B}}_k v\|_g}{\|v\|_g} \text{ (there exists a } v \text{ such that this equality holds)} \\
&= \sqrt{\frac{v^T \hat{\mathcal{B}}_k^T G_k \hat{\mathcal{B}}_k v}{v^T G_k v}} = \sqrt{\frac{v^T G_k \hat{\mathcal{B}}_k \hat{\mathcal{B}}_k v}{v^T G_k v}} \text{ (since } \mathcal{B}_k \text{ is self-adjoint, then } \hat{\mathcal{B}}_k^T G_k = G_k \hat{\mathcal{B}}_k) \\
&= \sqrt{\frac{\tilde{v}^T G_k^{1/2} \hat{\mathcal{B}}_k G_k^{-1/2} G_k^{1/2} \hat{\mathcal{B}}_k G_k^{-1/2} \tilde{v}}{\tilde{v}^T \tilde{v}}} = \sqrt{\frac{\tilde{v}^T G_k^{-1/2} G_k \hat{\mathcal{B}}_k G_k^{-1/2} G_k^{1/2} \hat{\mathcal{B}}_k G_k^{-1/2} \tilde{v}}{\tilde{v}^T \tilde{v}}} \\
&= \sqrt{\frac{\tilde{v}^T G_k^{-1/2} \hat{\mathcal{B}}_k^T G_k^{1/2} G_k^{1/2} \hat{\mathcal{B}}_k G_k^{-1/2} \tilde{v}}{\tilde{v}^T \tilde{v}}} = \sqrt{\frac{\tilde{v}^T M^T M \tilde{v}}{\tilde{v}^T \tilde{v}}} \\
&\leq \|M\|_2 \leq \operatorname{trace}(M) = \operatorname{trace}(\hat{\mathcal{B}}_k)
\end{aligned}$$

where G_k is the matrix expression of inner product of $\mathrm{T}_{x_k} \mathcal{M}$, $\tilde{v} = G_k^{1/2} v$, $M = G_k^{1/2} \hat{\mathcal{B}}_k^T G_k^{-1/2}$. Therefore,

$$\operatorname{trace}(\hat{\mathcal{B}}_{k+1}) \leq a_9 + (1 + \phi_k a_9 a_{10} + 2\phi_k a_{12} a_7) \operatorname{trace}(\hat{\mathcal{B}}_k).$$

This inequality implies that there exists constant $b_0 > 0$ such that

$$\operatorname{trace}(\hat{\mathcal{B}}_{k+1}) \leq b_0^k. \qquad (4.3.19)$$

Using (4.3.11) and (4.3.16), we have

$$\det(\hat{\mathcal{B}}_{k+1}) \geq \det(\hat{\mathcal{B}}_k) \frac{1}{a_{10}\alpha_k} \geq \det(\hat{\mathcal{B}}_1) \prod_{j=1}^{k} \frac{1}{a_{10}\alpha_j}. \qquad (4.3.20)$$

From the geometric/arithmetic mean inequality[2] applied to the eigenvalues of $\hat{\mathcal{B}}_{k+1}$, we know

$$\det(\hat{\mathcal{B}}_{k+1}) \leq (\frac{\text{trace}(\hat{\mathcal{B}}_{k+1})}{d})^d,$$

where d is the dimension of manifold $\mathcal{M}$. Therefore, by (4.3.19) and (4.3.20),

$$\prod_{j=1}^{k} \frac{1}{a_{10}\alpha_j} \leq \frac{1}{\det(\hat{\mathcal{B}}_1)}(\frac{\text{trace}(\hat{\mathcal{B}}_{k+1})}{d})^d \leq \frac{1}{\det(\hat{\mathcal{B}}_1)d^d}(b_0^d)^k.$$

Thus there exists a constant $a_{13} > 0$ such that

$$\prod_{j=1}^{k} \alpha_j \geq a_{13}^k,$$

for all $k \geq 1$. □

4.3.3 Main Convergence Result

With the preliminary lemmas in place, the main convergence result can be stated and proven in a manner that closely follows the Euclidean proof of [BNY87, Theorem 3.1].

Theorem 4.3.1. *Suppose Assumptions 4.3.1 and 4.3.2 hold and $\phi_k \in [0, 1-\delta]$. Then the sequence $\{x_k\}$ generated by Algorithm 3 converges to a minimizer x^* of f.*

Proof. Inequality (4.3.17) can be written as

$$\text{trace}(\hat{\mathcal{B}}_{k+1}) \leq \text{trace}(\hat{\mathcal{B}}_k) + a_9 + t_k\alpha_k, \tag{4.3.21}$$

where

$$t_k = \phi_k a_9 a_{10} - \frac{a_{11}(1-\phi_k)}{\cos^2\theta_k} + \frac{2\phi_k a_{12}}{\cos\theta_k}.$$

The proof is by contradiction. Assume $\cos\theta_k \to 0$, then $t_k \to -\infty$. So there exists a constant $K_0 > 0$ such that $t_k < -2a_9/a_{13}$ for all $k \geq K_0$. Using (4.3.21) and that $\hat{\mathcal{B}}_{k+1}$ is positive definite, we have

$$\begin{aligned} 0 < \text{trace}(\hat{\mathcal{B}}_{k+1}) &\leq \text{trace}(\hat{\mathcal{B}}_{K_0}) + a_9(k+1-K_0) + \sum_{j=K_0}^{k} t_k\alpha_k \\ &< \text{trace}(\hat{\mathcal{B}}_{K_0}) + a_9(k+1-K_0) - \frac{2a_9}{a_{13}}\sum_{j=K_0}^{k}\alpha_k. \end{aligned} \tag{4.3.22}$$

[2] For $x_i \geq 0$, $(\prod_{i=1}^{d} x_i)^{1/d} \leq \sum_{i=1}^{d} x_i/d$.

Applying the geometric/arithmetic mean inequality to (4.3.14), we get

$$\sum_{j=1}^{k} \alpha_j \geq k a_{13}$$

and therefore

$$\sum_{j=K_0}^{k} \alpha_j \geq k a_{13} - \sum_{j=1}^{K_0} \alpha_j. \tag{4.3.23}$$

Plugging (4.3.23) into (4.3.22), we obtain

$$\begin{aligned} 0 &< \operatorname{trace}(\hat{\mathcal{B}}_{K_0}) + a_9(k + 1 - K_0) - \frac{2a_9}{a_{13}} k a_{13} + \frac{2a_9}{a_{13}} \sum_{j=1}^{K_0-1} \alpha_k \\ &= \operatorname{trace}(\hat{\mathcal{B}}_{K_0}) + a_9(1 - k - K_0) + \frac{2a_9}{a_{13}} \sum_{j=1}^{K_0-1} \alpha_k. \end{aligned}$$

For large enough k, the right-hand side of the inequality is negative, which contradicts the assumption that $\cos\theta_k \to 0$. Therefore there exists a constant δ and a subsequence such that $\cos\theta_{k_j} > \delta > 0$ for all j, i.e., there is a subsequence that does not converge to 0. Applying Lemma 4.3.8 completes the proof. □

4.4 Constructing Isometric Vector Transport or Retraction

In order to apply an algorithm in the RBroyden family method, we must specify a retraction R, an isometric vector transport $\mathcal{T}_S$ and β that satisfy (4.2.6). Exponential mapping and parallel translation satisfy condition (4.2.6) with $\beta = 1$. However, for some manifolds, we do not have the analytical form of exponential mapping and parallel translation. Even if a form is known its evaluation may be unacceptably expensive. Two methods of constructing an isometric vector given a retraction and a method for constructing a retraction for any isometric vector transport are discussed in this section. In practice, the choice of the pair must also consider if an efficient implementation is possible.

4.4.1 Method 1 of Constructing an Isometric Vector Transport

Given a retraction R, if an associated isometric vector transport,$\mathcal{T}_{\mathrm{I}}$, for which there is an efficient implementation, is known then $\mathcal{T}_{\mathrm{I}}$ can be modified so that it satisfies condition (4.2.6). Consider $x \in \mathcal{M}$, $\eta \in \mathrm{T}_x \mathcal{M}$, $y = R_x(\eta)$ and define the tangent vectors $\xi_1 = \mathcal{T}_{\mathrm{I}_\eta} \eta$ and $\xi_2 = \beta \mathcal{T}_{R_\eta} \eta$ with the normalizing scalar $\beta = \frac{\|\eta\|}{\|\mathcal{T}_{R_\eta}\eta\|}$. We need P_y, a linear isometric operator on $\mathrm{T}_y \mathcal{M}$, such that $\xi_2 = P_y \xi_1$. Given P_y, the operator

$$\mathcal{T}_S = P_y \mathcal{T}_{\mathrm{I}}. \tag{4.4.1}$$

clearly satisfies condition (4.2.6). The natural idea is to use a Householder reflector, i.e.,

$$P_y = \mathrm{id} - \frac{2\nu\nu^\flat}{\nu^\flat\nu},$$

where $\nu = \xi_1 - \xi_2$. Unfortunately, $\mathcal{T}_S$ defined with a Householder reflector for P_y does not satisfy the consistency condition of vector transport.

P_y can be defined by using two Householder reflectors. Let $\omega \in \mathrm{T}_y \mathcal{M}, \|\omega\| = \|\xi_1\| = \|\xi_2\|$, be some tangent vector and define

$$P_y = (\mathrm{id} - \frac{2\nu_2\nu_2^\flat}{\nu_2^\flat\nu_2})(\mathrm{id} - \frac{2\nu_1\nu_1^\flat}{\nu_1^\flat\nu_1}),$$

where $\nu_1 = \xi_1 - \omega$ and $\nu_2 = \omega - \xi_2$. ω could be any tangent vector in $\mathrm{T}_y \mathcal{M}$ which satisfies $\|\omega\| = \|\xi_1\| = \|\xi_2\|$. If $\omega = -\xi_1$ or $-\xi_2$ then P_y is the well-known direct rotation from ξ_1 to ξ_2 in the inner product that defines the $\flat$. The use negative sign avoids numerical cancelation as ξ_1 approaches ξ_2, i.e., near convergence.

Since P_y approaches id when x approaches y, it is easy to check that $\mathcal{T}_S$ satisfies consistency condition of vector transport. Recall, that we have relaxed the definition of vector transport for $\mathcal{T}_S$ by requiring conditions (4.2.4) and (4.2.5) rather than smoothness. This is verified in Theorem 4.4.1.

Theorem 4.4.1. *The isometry*

$$\mathcal{T}_S = (\mathrm{id} - \frac{2\nu_2\nu_2^\flat}{\nu_2^\flat\nu_2})(\mathrm{id} - \frac{2\nu_1\nu_1^\flat}{\nu_1^\flat\nu_1})\mathcal{T}_{\mathrm{I}}, \tag{4.4.2}$$

where $\nu_1 = \xi_1 - \omega$ and $\nu_2 = \omega - \xi_2$ and ω, ξ_1 and ξ_2 are as defined above, satisfies (4.2.4) and (4.2.5). Specifically, for any $\bar{x} \in \mathcal{M}$, there exists a neighborhood of $\bar{x}$, $\mathcal{U}$, a constant c_0 such that for all $x, y \in \mathcal{U}$

$$\|\mathcal{T}_{S_\eta} - \mathcal{T}_{R_\eta}\| \leq c_0\|\eta\|$$

$$\|\mathcal{T}_{S_\eta}^{-1} - \mathcal{T}_{R_\eta}^{-1}\| \leq c_0\|\eta\|$$

where $\eta = R_x^{-1}(y)$.

Proof. By Lemma 4.3.7 there exists a constant $b_0 > 0$ such that

$$|\frac{1}{\beta(\eta)} - 1| \leq b_0\|\eta\|. \tag{4.4.3}$$

Since $\beta(\eta) = \frac{\|\eta\|}{\|\mathcal{T}_{R_\eta}\eta\|} = \frac{\|\mathcal{T}_{R_\eta}^{-1}\mathcal{T}_{R_\eta}\eta\|}{\|\mathcal{T}_{R_\eta}\eta\|}$ and $\mathcal{T}_{R_\eta}^{-1}$ is a smooth function, by using the same idea as Lemma 4.3.7, there exists a constant $b_1 > 0$ such that

$$|\beta(\eta) - 1| \leq b_1\|\eta\|. \tag{4.4.4}$$

Also, since

$$\begin{aligned}\xi_2 &= \beta(\eta)\mathcal{T}_{R_\eta}\mathcal{T}_{\mathrm{I}_\eta}^{-1}\xi_1\\ \xi_1 &= \frac{1}{\beta(\eta)}\mathcal{T}_{\mathrm{I}_\eta}\mathcal{T}_{R_\eta}^{-1}\xi_2,\end{aligned}$$

we have

$$\begin{aligned}\xi_2 - \xi_1 &= (\beta(\eta)\mathcal{T}_{R_\eta}\mathcal{T}_{\mathrm{I}_\eta}^{-1} - \mathrm{id})\xi_1\\ \xi_1 - \xi_2 &= (\frac{1}{\beta(\eta)}\mathcal{T}_{\mathrm{I}_\eta}\mathcal{T}_{R_\eta}^{-1} - \mathrm{id})\xi_2.\end{aligned}$$

Since $\mathcal{T}_R$, $\mathcal{T}_R^{-1}$, $\mathcal{T}_{\mathrm{I}}$ and $\mathcal{T}_{\mathrm{I}}^{-1}$ are all C^1, by (4.4.3), (4.4.4), and the definitions of ν_1 and ν_2, we have

$$\|\nu_2 + \nu_1\| \leq b_2\|\eta\|\|\nu_1\| \tag{4.4.5}$$

$$\|\nu_2 + \nu_1\| \leq b_2\|\eta\|\|\nu_2\| \tag{4.4.6}$$

Consider the difference of $\mathcal{T}_S$ and $\mathcal{T}_I$,

$$\begin{aligned}&\|\mathcal{T}_{S_\eta} - \mathcal{T}_{I_\eta}\|\\ &= \|(\frac{4\nu_2\nu_2^\flat\nu_1\nu_1^\flat}{\nu_2^\flat\nu_2\nu_1^\flat\nu_1} - \frac{2\nu_2\nu_2^\flat}{\nu_2^\flat\nu_2} - \frac{2\nu_1\nu_1^\flat}{\nu_1^\flat\nu_1})\mathcal{T}_{I_\eta}\|\\ &\leq b_3(\|\frac{2\nu_2\nu_2^\flat\nu_1\nu_1^\flat - 2\nu_2\nu_2^\flat\nu_1^\flat\nu_1}{\nu_2^\flat\nu_2\nu_1^\flat\nu_1}\| + \|\frac{2\nu_2\nu_2^\flat\nu_1\nu_1^\flat - 2\nu_1\nu_1^\flat\nu_2^\flat\nu_2}{\nu_2^\flat\nu_2\nu_1^\flat\nu_1}\|) \text{ (by Lemma 4.3.7)}\\ &= b_3(\|\frac{2\|\nu_2\|\|(\nu_2\nu_1^\flat - \nu_1\nu_2^\flat)\nu_1\|}{\nu_2^\flat\nu_2\nu_1^\flat\nu_1}\| + \|\frac{2\|\nu_1\|\|(\nu_2\nu_1^\flat - \nu_1\nu_2^\flat)\nu_2\|}{\nu_2^\flat\nu_2\nu_1^\flat\nu_1}\|) \text{ (since } \|\cdot\| \text{ is an induced norm)}\\ &\leq 2b_3(\frac{\|\nu_1\|\|(\nu_2\nu_1^\flat + \nu_2\nu_2^\flat - \nu_2\nu_2^\flat - \nu_1\nu_2^\flat)\|\|\nu_2\|}{\nu_2^\flat\nu_2\nu_1^\flat\nu_1} + \frac{\|\nu_1\|\|(\nu_2\nu_1^\flat + \nu_2\nu_2^\flat - \nu_2\nu_2^\flat - \nu_1\nu_2^\flat)\|\|\nu_2\|}{\nu_2^\flat\nu_2\nu_1^\flat\nu_1})\\ &\leq 2b_3(\frac{\|\nu_1\|(\|\nu_2\nu_1^\flat + \nu_2\nu_2^\flat\| + \|\nu_2\nu_2^\flat + \nu_1\nu_2^\flat\|)\|\nu_2\|}{\|\nu_1\|^2\|\nu_2\|^2} + \frac{\|\nu_1\|(\|\nu_2\nu_1^\flat + \nu_2\nu_2^\flat\| + \|\nu_2\nu_2^\flat + \nu_1\nu_2^\flat\|)\|\nu_2\|}{\|\nu_1\|^2\|\nu_2\|^2})\\ &= 2b_3(\frac{\|\nu_1\|(\|\nu_2\|\|\nu_1 + \nu_2\| + \|\nu_2\|\|\nu_2 + \nu_1\|)\|\nu_2\|}{\|\nu_1\|^2\|\nu_2\|^2} + \frac{\|\nu_1\|(\|\nu_2\|\|\nu_1 + \nu_2\| + \|\nu_2\|\|\nu_2 + \nu_1\|)\|\nu_2\|}{\|\nu_1\|^2\|\nu_2\|^2})\\ &\leq b_4\|\eta\| \text{ (by (4.4.5) and (4.4.6))}\end{aligned}$$

where b_3 and b_4 are positive constants. Finally, we have

$$\begin{aligned}\|\mathcal{T}_{S_\eta} - \mathcal{T}_{R_\eta}\| &\leq \|\mathcal{T}_{S_\eta} - \mathcal{T}_{I_\eta}\| + \|\mathcal{T}_{I_\eta} - \mathcal{T}_{R_\eta}\| \\ &\leq b_4\|\eta\| + b_5\|\eta\| \text{ (by Lemma 4.3.6)}\end{aligned}$$

proving the first inequality. The second inequality follows from a similar argument. □

4.4.2 Method 2 of Constructing an Isometric Vector Transport

Method 1 modifies a given isometric vector transport. In this section, a method is presented to construct directly an isometric vector transport that satisfies the condition (4.2.6). Method 2 requires a function that constructs an orthonormal basis of $\mathrm{T}_x \mathcal{M}$. Let d denote the dimension of manifold $\mathcal{M}$ and let the function giving a basis of $\mathrm{T}_x \mathcal{M}$ be $B : x \to B(x)$. For our earlier results that relax the smoothness requirement on vector transport, B need only be C^1.

Consider $x \in \mathcal{M}$, $\eta, \xi \in \mathrm{T}_x \mathcal{M}$, $y = R_x(\eta)$, $B_1 = B(x)$ and $B_2 = B(y)$. Then the isometric vector transport we want can be written as

$$\mathcal{T}_{S_\eta}\xi = B_2 T B_1^\flat \xi, \tag{4.4.7}$$

where T is an orthogonal matrix constructed so that $\mathcal{T}_S$ satisfies the desired condition.

Since B_1 and B_2 are bases, we know

$$\eta = B_1 c_1, \beta \mathcal{T}_{R_\eta}\eta = B_2 c_2,$$

where $c_1, c_2 \in \mathbb{R}^d$ contain the coordinate of η and $\beta \mathcal{T}_{R_\eta}\eta$ relative to the basis formed by the columns of B_1, B_2. By (4.2.6): $\beta \mathcal{T}_{R_\eta}\eta = \mathcal{T}_{S_\eta}\eta$, we have

$$B_2 c_2 = B_2 T B_1^\flat B_1 c_1,$$

which is

$$c_2 = T c_1. \tag{4.4.8}$$

For any orthogonal matrix T satisfying (4.4.8), $\mathcal{T}_S$ defined by (4.4.7) satisfies condition (4.2.6). Using the same idea as Method 1, we obtain the desired isometric vector transport

$$\mathcal{T}_{S_\eta}\xi = B_2(I - \frac{2 v_2 v_2^T}{v_2^T v_2})(I - \frac{2 v_1 v_1^T}{v_1^T v_1}) B_1^\flat \xi, \tag{4.4.9}$$

where $v_1 = B_1^\flat \eta - w$, $v_2 = w - \beta B_2^\flat \mathcal{T}_{R_\eta} \eta$. w can be any vector such that $\|w\| = \|c_1\| = \|c_2\|$ and choosing $w = -c_1$ or $-c_2$ yields a direct rotation.

The problem therefore becomes how to build the function B. Absil et al. [AMS08, p. 37] give an approach based on $(\mathcal{U}, \varphi)$, a chart of the manifold $\mathcal{M}$ that yields a smooth B. E_i, the i-th coordinate vector field of $(\mathcal{U}, \varphi)$on $\mathcal{U}$ defined by

$$(E_i f)(x) := \partial_i (f \circ \varphi^{-1})(\varphi(x)) = D(f \circ \varphi^{-1})(\varphi(x))[e_i].$$

These coordinate vector fields are smooth and every vector field ξ admits the decomposition

$$\xi = \sum_i (\xi \varphi_i) E_i$$

on $\mathcal{U}$. The function

$$\tilde{B} : x \to \tilde{B}(x) = \{E_2, E_2, \dots, E_d\}$$

is a smooth function that builds basis on $\mathcal{U}$. Finally, any orthogonalization method, such as Gram-Schmidt algorithm or a QR decomposition, can be used to get an orthonormal basis giving the function

$$B : x \to B(x) = \tilde{B}(x) Q(x),$$

where $Q(x)$ is a upper triangle matrix with positive diagonal terms.

4.4.3 Constructing a Retraction

In this section, it is assumed that an efficient isometric vector transport is given. This vector transport may have been derived from a particular choice of associated retraction but that combination may not satisfy condition (4.2.6). However, a vector transport may be associated with more than one retraction and we propose a method to construct a retraction so that the combination satisfies condition (4.2.6).

Given $\mathcal{T}_S$ consider the differential equation

$$\begin{aligned} \frac{d}{dt} R_x(t\eta) &= \mathcal{T}_{S_{t\eta}} \eta, \\ R_x(0) &= x. \end{aligned} \tag{4.4.10}$$

The solution $R_x(t\eta)$ is a retraction such that its differentiated retraction satisfies the locking condition with $\mathcal{T}_S$. This is easily seen by letting η be an arbitrary vector in $\mathrm{T}_x \mathcal{M}$. By definition we have

$$\frac{d}{dt}R_x(t\eta)|_{t=\tau_0} = \frac{d}{d\tau}R_x(\tau_0\eta + \tau\eta))|_{\tau=0}.$$

The right side is by definition vector transport of η by differentiated retraction and we therefore have

$$\frac{d}{d\tau}R_x(\tau_0\eta + \tau\eta))|_{\tau=0} = \mathcal{T}_{R_{\tau_0\eta}}\eta.$$

The locking condition (4.2.6) of differentiated retraction and $\mathcal{T}_S$ is

$$\mathcal{T}_{R_{\tau_0\eta}}\eta = \mathcal{T}_{S_{\tau_0\eta}}\eta$$

which is the same as (4.4.10) with $\beta \equiv 1$.

For an embedded manifold, $\frac{d}{dt}R_x(t\eta) \in \mathbb{R}^n$ is the vector whose components are the derivatives of the coefficient of $R_x(t\eta)$ with respect to some chosen basis of $\mathcal{R}^n$. In this case, the retraction R can be obtained by solving the differential equation (4.4.10). Note that η and x are given and fixed, $\mathcal{T}_{S_{t\eta}}\eta$ is then a function of $R_x(t\eta)$ and therefore it can be rewritten as $F(R_x(t\eta))$.

If $F(R_x(t\eta))$ is a linear function, then there exists a constant matrix M such that $F(R_x(t\eta)) = MR_x(t\eta)$. Thus, (4.4.10) becomes

$$\begin{aligned} \frac{d}{dt}R_x(t\eta) &= MR_x(t\eta), \\ R_x(0) &= x, \end{aligned} \tag{4.4.11}$$

which has a closed solution $R_x(t\eta) = \exp(tM)x$. An isometric vector transport that satisfies (4.4.11) exists for many manifolds. For example, isometric vector transport by parallelization (9.2.19)

$$\mathcal{T}_{S_\eta}\xi = B_2 B_1^\flat \xi, \tag{4.4.12}$$

is such a choice for the compact Stiefel manifold, where η, ξ, B_1 and B_2 are the same as those in (4.4.7). Note that the corresponding matrix M for this choice has substantial structure that can results in the retraction and transport having computational complexity similar to those based on orthogonal factorization. This choice is also easily adapted to the Grassmann manifold when it is represented as a submanifold of the compact Stiefel. For quotient manifolds constructing a retraction must carefully consider the representation and horizontal distribution used. Since our experiments are on an embedded manifold we discuss quotient manifold no further here.

4.5 Limited-memory RBFGS

In the form of RBroyden family methods discussed above explicit representations are needed for the operators $\mathcal{B}_k$, $\tilde{\mathcal{B}}_k$, $\mathcal{T}_{S_{\alpha_k \eta_k}}$, and $\mathcal{T}_{S_{\alpha_k \eta_k}}^{-1}$. These may not be available. Furthermore, even if explicit expressions are known, applying them may be unacceptably expensive computationally, e.g., the matrix multiplication required in the update of $\mathcal{B}_k$. Generalizations of the Euclidean limited-memory BFGS method can solve this problem for RBFGS. The idea of limited-memory RBFGS (LRBFGS) is to store some number of the most recent s_k and y_k and to transport those vectors to the new tangent space rather than the entire matrix $\mathcal{H}_k$.

For RBFGS, the inverse update formula is

$$\mathcal{H}_{k+1} = \mathcal{V}_k^\flat \tilde{\mathcal{H}}_k \mathcal{V}_k + \rho_k s_k s_k^\flat,$$

where

$$\rho_k = \frac{1}{g(y_k, s_k)} \text{ and } \mathcal{V}_k = \mathrm{id} - \rho_k y_k s_k^\flat.$$

If the $m+1$ most recent s_k and y_k are stored then we have

$$\begin{aligned}
\mathcal{H}_{k+1} = {} & \tilde{\mathcal{V}}_k^\flat \tilde{\mathcal{V}}_{k-1}^\flat \cdots \tilde{\mathcal{V}}_{k-m}^\flat \tilde{\mathcal{H}}_{k+1}^0 \tilde{\mathcal{V}}_{k-m} \cdots \tilde{\mathcal{V}}_{k-1} \tilde{\mathcal{V}}_k \\
& + \rho_{k-m} \tilde{\mathcal{V}}_k^\flat \tilde{\mathcal{V}}_{k-1}^\flat \cdots \tilde{\mathcal{V}}_{k-m+1}^\flat s_{k-m}^{(k+1)} {s_{k-m}^{(k+1)}}^\flat \tilde{\mathcal{V}}_{k-m+1} \cdots \tilde{\mathcal{V}}_{k-1} \tilde{\mathcal{V}}_k \\
& + \cdots \\
& + \rho_k s_k^{(k+1)} s_k^{(k+1)},
\end{aligned}$$

where $\tilde{\mathcal{V}}_i = \mathrm{id} - \rho_i y_i^{(k+1)} {s_i^{(k+1)}}^\flat$ and $\tilde{\mathcal{H}}_{k+1}^0$ is the initial Hessian approximation for step $k+1$. Note that $\tilde{\mathcal{H}}_{k+1}^0$ is not necessarily $\tilde{\mathcal{H}}_{k-m}$. It can be any positive definite self-adjoint operator. Similar to the Euclidean case, we use

$$\mathcal{H}_{k+1}^0 = \frac{g(s_k, y_k)}{g(y_k, y_k)} \mathrm{id}\,. \tag{4.5.1}$$

It is easily seen that $\eta_{k+2} = -\mathcal{H}_{k+1} \operatorname{grad} f(x_{k+1})$ which yields the Step 3 to Step 12 of Algorithm 4 and the explicit form of $\mathcal{T}_S$ is not required.

The following is a limited-memory algorithm based on this idea.

Note Step 17 of Algorithm 4. The vector $s_{k-m}^{(k+1)}$ is obtained by transporting $s_{k-m}^{(k-m+1)}$ m times. If the vector transport is insensitive to finite precision then the approach is acceptable. Otherwise, $s_{k-m}^{(k+1)}$ may be not in $\mathrm{T}_{x_{k+1}} \mathcal{M}$. Care must be taken to avoid this situation. One possibility is to

Algorithm 4 LRBFGS

Input: Riemannian manifold $\mathcal{M}$ with Riemannian metric g; a retraction R; isometric vector transport $\mathcal{T}_S$ that satisfies (4.2.6); smooth function f on $\mathcal{M}$; initial iterate $x_0 \in \mathcal{M}$; an integer $m > 0$.

```
1:  $k = 0$, $\varepsilon > 0$, $0 < c_1 < c_2 < 1$, $\gamma_0 = 1$, $l = 0$.
2:  $\mathcal{H}_k^0 = \gamma_k \operatorname{id}$. Obtain $\eta_k \in \mathrm{T}_{x_k}\mathcal{M}$ by the following algorithm:
  3:  $q \leftarrow \operatorname{grad} f(x_k)$
  4:  for $i = k-1, k-2, \dots, l$ do
  5:      $\xi \leftarrow \rho_i g(s_i^{(k)}, q)$;
  6:      $q \leftarrow q - \xi_i y_i^{(k)}$;
  7:  end for
  8:  $r \leftarrow \mathcal{H}_k^0 q$;
  9:  for $i = l, l+1, \dots, k-1$ do
  10:     $\omega \leftarrow \rho_i g(y_i^{(k)}, r)$;
  11:     $r \leftarrow r + s_i^{(k)}(\xi_i - \omega)$;
  12: end for
13: set $\eta_k = -r$;
14: find $\alpha_k$ that satisfies Wolfe conditions
        $f(x_{k+1}) \leq f(x_k) + c_1\alpha_k g(\operatorname{grad} f(x_k), \eta_k)$
        $\frac{d}{dt} f(R_x(t\eta_k))|_{t=\alpha_k} \geq c_2 \frac{d}{dt} f(R_x(t\eta_k)|_{t=0}$
15: Set $x_{k+1} = R_{x_k}(\alpha_k\eta_k)$. If $\|\operatorname{grad} f(x_{k+1})\| > \varepsilon$, then break.
16: Define $\rho_k = 1/g(s_k, y_k)$, $s_k^{(k+1)} = \mathcal{T}_{S_{\alpha_k\eta_k}} \alpha_k\eta_k$ and $y_k^{(k+1)} = \operatorname{grad} f(x_{k+1})/\beta_k - \mathcal{T}_{S_{\alpha_k\eta_k}} \operatorname{grad} f(x_k)$,
    $\rho_k = 1/g(s_k^{(k+1)}, y_k^{(k+1)})$ and $\gamma_{k+1} = g(s_k^{(k+1)}, y_k^{(k+1)})/\|y_k^{(k+1)}\|^2$, where $\beta_k = \frac{\|\alpha_k\eta_k\|}{\|\mathcal{T}_{R_{\alpha_k\eta_k}}\alpha_k\eta_k\|}$.
17: Let $l = \max\{k-m, 0\}$. Add $s_k^{(k+1)}$, $y_k^{(k+1)}$ and $\rho_k$ into storage and if $k > m$, then
    discard vector pair $\{s_{l-1}^{(k)}, y_{l-1}^{(k)}\}$ and scaler $\rho_{l-1}$ from storage; Transport $s_l^{(k)}, s_{l+1}^{(k)}, \dots, s_{k-1}^{(k)}$
    and $y_l^{(k)}, y_{l+1}^{(k)}, \dots, y_{k-1}^{(k)}$ from $\mathrm{T}_{x_k}\mathcal{M}$ to $\mathrm{T}_{x_{k+1}}\mathcal{M}$ by $\mathcal{T}_S$, then get $s_l^{(k+1)}, s_{l+1}^{(k+1)}, \dots, s_{k-1}^{(k+1)}$ and
    $y_l^{(k+1)}, y_{l+1}^{(k+1)}, \dots, y_{k-1}^{(k+1)}$.
18: $k = k+1$, goto 2.
```

project $s_i^{(k+1)}, y_i^{(k+1)}, i = l, l+1, \dots, k-1$ to tangent space $T_{x_{k+1}}\mathcal{M}$ after every transport. The other possibility is to store $x_{l+1}, x_{l+2}, \dots, x_k$, $s_l^{(l+1)}, s_{l+1}^{(l+2)}, \dots, s_k^{(k+1)}$ and $y_l^{(l+1)}, y_{l+1}^{(l+2)}, \dots, y_k^{(k+1)}$ and transport $s_i^{(i+1)}, y_i^{(i+1)}$ from $T_{x_{i+1}}\mathcal{M}$ to $T_{x_{k+1}}\mathcal{M}$ during Steps 3 to 12.

4.6 Ring and Wirth's RBFGS Update Formula

In Ring and Wirth's RBFGS [RW12] for infinite dimensional Riemannian manifolds, the direction vector η_k is chosen to define a function on $\mathrm{T}_{x_k}\mathcal{M}$ that satisfies the equation

$$\mathcal{B}_k(\eta_k, \xi) = \mathrm{D}\, f_{R_{x_k}}(0)[\xi] = g(\operatorname{grad} f(x_k), \xi)$$

for all $\xi \in \mathrm{T}_{x_k}\mathcal{M}$.

Ring and Wirth's update is

$$\mathcal{B}_{k+1}(\mathcal{T}_{S_{\alpha_k \eta_k}}\zeta, \mathcal{T}_{S_{\alpha_k \eta_k}}\xi) = \mathcal{B}_k(\zeta, \xi) - \frac{\mathcal{B}_k(s_k, \zeta)\mathcal{B}_k(s_k, \xi)}{\mathcal{B}_k(s_k, s_k)} + \frac{y_k(\zeta) y_k(\xi)}{y_k(s_k)},$$

where $s_k = R_{x_k}^{-1}(x_{k+1}) \in \mathrm{T}_{x_k}\mathcal{M}$ and $y_k = \mathrm{D}\, f_{R_{x_k}}(s_k) - \mathrm{D}\, f_{R_{x_k}}(0)$ is a cotangent vector of at x_k, i.e., $\mathrm{D}\, f_{R_{x_k}}(s_k)[\xi] = g(\operatorname{grad} f(R_{x_k}(s_k)), \mathcal{T}_{R_{s_k}}\xi)$ for all $\xi \in \mathrm{T}_{x_k}\mathcal{M}$. If the dimension of the manifold is finite then using coordinates yields the following form of the update to the matrix that defines the bilinear function

$$\hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^T \hat{\mathcal{B}}_{k+1} \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}} = \hat{\mathcal{B}}_k - \frac{\hat{\mathcal{B}}_k \hat{s}_k \hat{s}_k^T \hat{\mathcal{B}}_k}{\hat{s}_k^T \hat{\mathcal{B}}_k \hat{s}_k} + \frac{\hat{y}_k \hat{y}_k^T}{\hat{y}_k^T \hat{s}_k}, \tag{4.6.1}$$

where $\hat{y}_k$ satisfies $y_k(\eta) = \hat{y}_k^T \hat{\eta}$ for all $\eta \in \mathrm{T}_{x_k}\mathcal{M}$ and $\mathcal{B}_k(\zeta, \xi) = \hat{\zeta}^T \hat{\mathcal{B}}_k \hat{\xi}$ for all $\zeta, \xi \in \mathrm{T}_{x_k}\mathcal{M}$. In contrast, the bilinear function defined by our methods is $g(\zeta, \mathcal{B}_k \xi) = \hat{\zeta}^T G_k \hat{\mathcal{B}}_k \hat{\xi}$ where G_k is the matrix expression of the metric in $\mathrm{T}_{x_k}\mathcal{M}$. Note that $\hat{\mathcal{B}}_k$ in Ring and Wirth's update plays the same role as $G_k \hat{\mathcal{B}}_k$ in our RBFGS. Ring and Wirth's expressions absorb the matrix G_k, implicit in the definition of their inner product, into the definitions of $\hat{\mathcal{B}}_k$ and y_k. Therefore, their RBFGS update and our RBFGS update forms are the same. The difference between the algorithms is the definitions of s_k and y_k.

Ring and Wirth do not derive an inverse Hessian approximation for their RBFGS. However, the following coordinate expression of the inverse Hessian approximation update can be derived from (4.6.1)

$$\hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^{-1} \hat{\mathcal{H}}_{k+1} (\hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^T)^{-1} = (I - \frac{\hat{s}_k \hat{y}_k^T}{\hat{y}_k^T \hat{s}_k}) \hat{\mathcal{H}}_k (I - \frac{\hat{y}_k \hat{s}_k^T}{\hat{y}_k^T \hat{s}_k}) + \frac{\hat{s}_k \hat{s}_k^T}{\hat{y}_k^T \hat{s}_k}. \tag{4.6.2}$$

Finite dimensional versions of Ring and Wirth's RBFGS based on these two updates are easily implemented. Our experiments include the more efficient inverse Hessian approximation form.

4.7 Property of RBroyden Family Method

The Euclidean Broyden family algorithms with different ϕ_k are equivalent if the line search algorithm is exact [NW06]. However, Riemannian Broyden family does not have this property in general. The next theorem shows that the property holds if and only if $\beta_k \equiv 1$.

Theorem 4.7.1. *If the line search is exact and a same local minimizer is chosen when search directions are the same, same $\mathcal{B}_0$ and x_0 is used, x_0 or x_1 is not a local minimizer, $\mathcal{T}_{S_{\alpha_k \eta_k}} \operatorname{grad} f(x_k)$ is not equal to $\operatorname{grad} f(x_{k+1})$, then all methods in RBroyden family method with $\phi_k \geq \phi_k^c$ generate the same sequence of iterates if and only if $\beta_k \equiv 1$.*

Proof. Without losing generality, we analyze the update formula $\mathcal{H}_k = \mathcal{B}_k^{-1}$ rather than $\mathcal{B}_k$. The update formula is

$$\mathcal{H}_{k+1} = \tilde{\mathcal{H}}_k - \frac{\tilde{\mathcal{H}}_k y_k (\tilde{\mathcal{H}}_k^* y_k)^\flat}{(\tilde{\mathcal{H}}_k^* y_k)^\flat y_k} + \frac{s_k s_k^\flat}{s_k^\flat y_k} + \tilde{\phi}_k g(y_k, \tilde{\mathcal{H}}_k y_k) u_k u_k^\flat, \tag{4.7.1}$$

where $\tilde{\mathcal{H}}_k = \mathcal{T}_{S_{\alpha_k \eta_k}} \circ \mathcal{H}_k \circ \mathcal{T}_{S_{\alpha_k \eta_k}}^{-1}$,

$$u_k = \frac{s_k}{g(s_k, y_k)} - \frac{\tilde{\mathcal{H}}_k y_k}{g(y_k, \tilde{\mathcal{H}}_k y_k)}, \tag{4.7.2}$$

$\tilde{\phi}_k$ is an arbitrary number greater than $\tilde{\phi}_k^c$ where $\tilde{\phi}_k^c = 1/(1 - \nu_k)$, and

$$\nu_k = (g(y_k, \tilde{\mathcal{H}}_k y_k) g(s_k, \tilde{\mathcal{H}}_k^{-1} s_k)) / g(y_k, s_k)^2.$$

If the line search is exact, we have that

$$\frac{d}{dt} f(R_x(t\eta_k))|_{t=\alpha_k} = 0,$$

which is

$$g(\operatorname{grad} f(x_{k+1}), s_k) = 0. \tag{4.7.3}$$

Use g_{k+1} to denote $g(\operatorname{grad} f(x_{k+1}), \tilde{\mathcal{H}}_k \operatorname{grad} f(x_{k+1}))$ and h_k to denote $g(\operatorname{grad} f(x_k), \mathcal{H}_k \operatorname{grad} f(x_k))$. Noticing that

$$y_k = \operatorname{grad} f(x_{k+1})/\beta_k - \mathcal{T}_{S_{\alpha_k \eta_k}} \operatorname{grad} f(x_k) \tag{4.7.4}$$

$$s_k = -\alpha_k \tilde{\mathcal{H}}_k \mathcal{T}_{S_{\alpha_k \eta_k}} \operatorname{grad} f(x_k), \tag{4.7.5}$$

we have

$$g(\operatorname{grad} f(x_{k+1}), \tilde{\mathcal{H}}_k \mathcal{T}_{S_{\alpha_k \eta_k}} \operatorname{grad} f(x_k)) = 0 \tag{4.7.6}$$

$$g(s_k, y_k) = \alpha_k h_k \tag{4.7.7}$$

$$g(\tilde{\mathcal{H}}_k y_k, y_k) = g_{k+1}/\beta^2 + h_k. \tag{4.7.8}$$

Using (4.7.1), we have

$$\begin{aligned}
-\eta_{k+1} &= \mathcal{H}_{k+1} \operatorname{grad} f(x_{k+1}) \\
&= \tilde{\mathcal{H}}_k \operatorname{grad} f(x_{k+1}) - \frac{\tilde{\mathcal{H}}_k y_k g(\tilde{\mathcal{H}}_k y_k, \operatorname{grad} f(x_{k+1}))}{g(\tilde{\mathcal{H}}_k y_k, y_k)} + \frac{s_k g(s_k, \operatorname{grad} f(x_{k+1}))}{g(s_k, y_k)} \\
&\quad + \tilde{\phi}_k g(y_k, \tilde{\mathcal{H}}_k y_k) u_k g(u_k, \operatorname{grad} f(x_{k+1}))
\end{aligned}$$

Using (4.7.3), (4.7.2), (4.7.5), (4.7.7) and (4.7.4), we obtain

$$\begin{aligned}
-\eta_{k+1} = \tilde{\mathcal{H}}_k \operatorname{grad} f(x_{k+1}) &- \frac{\tilde{\mathcal{H}}_k y_k g_{k+1}}{g(\tilde{\mathcal{H}}_k y_k, y_k)} \\
&+ \tilde{\phi}_k g(y_k, \tilde{\mathcal{H}}_k y_k) g(u_k, \operatorname{grad} f(x_{k+1}))[\frac{-\alpha_k \tilde{\mathcal{H}}_k \mathcal{T}_{S_{\alpha_k \eta_k}} \operatorname{grad} f(x_k)}{\alpha_k h_k} \\
&- \frac{\tilde{\mathcal{H}}_k (\operatorname{grad} f(x_{k+1})/\beta_k - \mathcal{T}_{S_{\alpha_k \eta_k}} \operatorname{grad} f(x_k))}{g(y_k, \tilde{\mathcal{H}}_k y_k)}]
\end{aligned}$$

Using (4.7.8) and (4.7.4), we obtain

$$\begin{aligned}
-\eta_{k+1} = &\frac{g_{k+1}/\beta_k^2 - g_{k+1}/\beta_k + h_k}{g(\tilde{\mathcal{H}}_k y_k, y_k)} \tilde{\mathcal{H}}_k \operatorname{grad} f(x_{k+1}) + \frac{g_{k+1}}{g(\tilde{\mathcal{H}}_k y_k, y_k)} \tilde{\mathcal{H}}_k \mathcal{T}_{S_{\alpha_k \eta_k}} \operatorname{grad} f(x_k) \\
&- \tilde{\phi}_k g(y_k, \tilde{\mathcal{H}}_k y_k) g(u_k, \operatorname{grad} f(x_{k+1}))[\frac{\tilde{\mathcal{H}}_k \mathcal{T}_{S_{\alpha_k \eta_k}} \operatorname{grad} f(x_k)}{h_k} \\
&+ \frac{\tilde{\mathcal{H}}_k (\operatorname{grad} f(x_{k+1})/\beta_k - \mathcal{T}_{S_{\alpha_k \eta_k}} \operatorname{grad} f(x_k))}{g(y_k, \tilde{\mathcal{H}}_k y_k)}]
\end{aligned}$$

Finally, we use (4.7.8) and obtain,

$$-\eta_{k+1} = \frac{1}{g(\tilde{\mathcal{H}}_k y_k, y_k)} (g_{k+1} \tilde{\mathcal{H}}_k \mathcal{T}_{S_{\alpha_k \eta_k}} \operatorname{grad} f(x_k) + (g_{k+1}/\beta_k^2 - g_{k+1}/\beta_k + h_k) \tilde{\mathcal{H}}_k \operatorname{grad} f(x_{k+1})) \tag{4.7.9}$$

$$- \tilde{\phi}_k \frac{g(u_k, \operatorname{grad} f(x_{k+1}))}{\beta_k^2 h_k} (g_{k+1} \tilde{\mathcal{H}}_k \mathcal{T}_{S_{\alpha_k \eta_k}} \operatorname{grad} f(x_k) + \beta_k h_k \tilde{\mathcal{H}}_k \operatorname{grad} f(x_{k+1})). \tag{4.7.10}$$

We can see that once $\beta_k \neq 1$, the vectors in parenthesis of (4.7.9) and (4.7.10) are different since $(g_{k+1}/\beta_k^2 - g_{k+1}/\beta_k + h_k) = \beta_k h_k$ requires $g_{k+1}/\beta_k^2 + h_k = 0$. However, $g_{k+1}/\beta_k^2 + h_k = g(\tilde{\mathcal{H}}_k y_k, y_k) \neq$

0 if the iterates do not terminate. Once $\tilde{\phi}_k$ changes, the direction η_{k+1} changes. Therefore, iterates change for different $\tilde{\phi}_k$.

Once $\beta_k \equiv 1$, the vectors in parenthesis of (4.7.9) and (4.7.10) are the same. The different $\tilde{\phi}_k$ only make the length of η_{k+1} different and the direction are the same. Since the same minimizer is used by assumption, the identical iterates are obtained for all RBroyden family methods with $\tilde{\phi}_k \geq \tilde{\phi}_k^c$. □

CHAPTER 5

RIEMANNIAN DENNIS-MORÉ CONDITIONS

5.1 Introduction

Superlinear convergence analyses of Euclidean quasi-Newton methods for the problems of finding zeros of a vector field and optimizing a cost function are based on versions of the well-known Dennis-Moré condition [DM74, DM77, DS83]. The relevant two Euclidean forms of Dennis-Moré condition are given in Theorems 5.1.1 and 5.1.2.

Theorem 5.1.1 (Euclidean Dennis-Moré Condition 1 (EDM1) [DM74, Theorem 2.2])**.** *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be continuously differentiable in the open, convex set D in $\mathbb{R}^n$, and assume that for some $\bar{x}$ in D, $F'(\bar{x})$ is nonsingular. Let $\{B_k\}$ be a sequence of nonsingular matrices and suppose that for some x_0 in D the sequence $\{x_k\}$ where*

$$x_{k+1} = x_k - B_k^{-1} F(x_k)$$

remains in D and converges to $\bar{x}$. Then $\{x_k\}$ converges q-superlinearly to $\bar{x}$ and $F(\bar{x}) = 0$ if and only if

$$\lim_{k \to \infty} \frac{\|(B_k - F'(\bar{x}))s_k\|}{\|s_k\|} = 0.$$

where $s_k = x_{k+1} - x_k$.

Theorem 5.1.2 (Euclidean Dennis Moré Condition 2 (EDM2) [DM77, Theorem 6.4])**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable in an open set D and consider the iteration $x_{k+1} = x_k + \alpha_k p_k$, where p_k is a descent direction and α_k satisfies the Wolfe conditions. If the sequence $\{x_k\}$ converges to a point $\bar{x}$ such that $\operatorname{Hess} f(\bar{x})$ is positive definite, and if the search direction satisfies*

$$\lim_{k \to \infty} \frac{\| \operatorname{grad} f(x_k) + \operatorname{Hess} f(x_k) p_k\|}{\|p_k\|} = 0,$$

then there is an index $k_0 \geq 0$ such that $\alpha_k = 1$ is admissible for $k \geq k_0$. Moreover, $\operatorname{grad} f(\bar{x}) = 0$ and if $\alpha_k = 1$ for all $k \geq k_0$, then $\{x_k\}$ converges superlinearly to $\bar{x}$.

EDM1 is a necessary and sufficient condition for superlinear convergence of a sequence of a particular form to a zero of a vector field [1]. EDM2 is the corresponding form for optimizing a real-valued function on a Euclidean space [2].

Both of these Euclidean conditions have been generalized. Gallivan et. al. [GQA12] generalize EDM1 to a Riemannian manifold. In fact, their requirement of $F \in C^2$ can be relaxed to $F \in C^1$. Ring and Wirth [RW12] prove a Riemannian version of EDM2 by considering the lifted function $\hat{f}_x(\eta) = f(R_x(\eta))$ where R is a retraction and $\eta \in \mathrm{T}_x \mathcal{M}$.

In this chapter, a Riemannian version of EDM1 that subsumes the version in [GQA12] is derived. The generalization of Euclidean Hessian to a Riemannian manifold is not unique and the choice made determines the Riemannian version of EDM2. One possibility is to use the Euclidean Hessian of the lifted function, $\operatorname{Hess} \hat{f}_x(\eta)$, as Ring and Wirth used. The more rigorous and more natural possibility used here is the uniquely defined Riemannian Hessian, $\operatorname{Hess} f(x)$ discussed in Chapter 1 for which, in general, $\operatorname{Hess} \hat{f}_{x_k}(0) \neq \operatorname{Hess} f(x)$. In Chapter 6, our Riemannian version of EDM2 is used to prove the superlinear convergence of the RBroyden family method for optimization of Chapter 4.

Chapter 5 is organized as follows. Section 5.2.1 derives a Riemannian Dennis-Moré condition for finding the zeros of a vector field, Theorem 5.2.2, and associated Riemannian and Euclidean results. Section 5.2.2 derives a Riemannian Dennis-Moré condition for optimization, Theorem 5.2.4.

5.2 Riemannian Dennis-Moré Conditions

5.2.1 Riemannian Dennis-Moré Conditions for a Vector Field

For completeness, we state the Riemannian Dennis Moré condition given by Gallivan et al. [GQA12]. Note that the generalizations of B_k and $F'(\bar{x})$ on a Riemannian manifold are linear operators on tangent spaces and vector transport is required to move tangent vectors into the same tangent space when comparing or performing linear combination.

[1] The proof of EDM1 in [DM74] was incorrect. The authors fixed the problem in [DM77] by requiring the vector field satisfy $F \in C^1$. The proof in [DS83, Theorem 8.2.4] requires F to be Lipschitz continuously differentiable, i.e., $F \in C^1$ and the derivative is Lipschitz continuous. This is not necessary since [DS83, Eq. 8.2.18] in the proof can be replaced by $o(\|e_k\| + \|e_{k+1}\|)$, which does not require Lipschitz continuously differentiable, and the proof still holds.

[2] The statement of [DM77, Theorem 6.4] is not complete since it does not mention choosing $\alpha_k = 1$ when $k \geq k_0$. The statement of [NW99, Theorem 3.5] requires $f \in C^3$ which is too strong. In addition, $\operatorname{grad} f(x^*) = 0$ is given as an assumption, but it should be a consequence. The first problem is fixed in [NW06, Theorem 3.6] but the second remains. [DS83, Theorem 6.3.4] requires $\operatorname{Hess} f$ to be Lipschitz continuous. Requiring $f \in C^2$ is enough, which is the original condition given in [DM77, Theorem 6.4].

Theorem 5.2.1 (Riemannian Dennis Moré Condition 1 (RDM1) [GQA12, Theorem 14.1])**.** *Let $\mathcal{M}$ be a Riemannian manifold endowed with a C^2 vector transport $\mathcal{T}$ and an associated retraction R. Let F be a C^2 tangent vector field on $\mathcal{M}$. Also let $\mathcal{M}$ be endowed with an affine connection ∇. Let $\mathbb{D}F(x)$ denote the linear transformation of $\mathrm{T}_x\,\mathcal{M}$ defined by $\mathbb{D}F(x)[\xi_x] = \nabla_{\xi_x}F$ for all tangent vector ξ_x to $\mathcal{M}$ at x. Let $\{\mathcal{B}_k\}$ be a sequence of nonsingular linear transformations of $\mathrm{T}_{x_k}\,\mathcal{M}$, where $k = 0, 1, \ldots, x_{k+1} = R_{x_k}(\eta_k)$, and $\eta_k = -\mathcal{B}_k^{-1}F(x_k)$. Assume that $\mathbb{D}F(\bar{x})$ is nonsingular, $x_k \neq \bar{x}, \forall k$, and $\lim_{k\to\infty} x_k = \bar{x}$. Then $\{x_k\}$ converges superlinearly to $\bar{x}$ and $F(\bar{x}) = 0$ if and only if*

$$\lim_{k\to\infty} \frac{\|[\mathcal{B}_k - \mathcal{T}_{\xi_k}\mathbb{D}F(\bar{x})\mathcal{T}_{\xi_k}^{-1}]\eta_k\|}{\|\eta_k\|} = 0, \tag{5.2.1}$$

where $\xi_k \in \mathrm{T}_{\bar{x}}\,\mathcal{M}$ is defined by $\xi_k = R_{\bar{x}}^{-1}(x_k)$, i.e. $R_{\bar{x}}(\xi_k) = x_k$.

Our main Riemannian result for vector fields is given in Theorem 5.2.2.

Theorem 5.2.2 (Riemannian Dennis-Moré Condition 2 (RDM2))**.** *Let $\mathcal{M}$ be a Riemannian manifold endowed with a vector transport $\mathcal{T}$ and an associated retraction R. Let F be a C^1 tangent vector field on $\mathcal{M}$. Also let $\mathcal{M}$ be endowed with an affine connection ∇. Let $\mathbb{D}F(x)$ denote the linear transformation of $\mathrm{T}_x\,\mathcal{M}$ defined by $\mathbb{D}F(x)[\xi_x] = \nabla_{\xi_x}F$ for all tangent vector ξ_x to $\mathcal{M}$ at x. Let $\{\eta_k \in \mathrm{T}_{x_k}\,\mathcal{M}\}$ be a sequence of nonzero tangent vectors and $x_{k+1} = R_{x_k}(\eta_k)$. Assume that $\mathbb{D}F(\bar{x})$ is nonsingular, $x_k \neq \bar{x}, \forall k$, and $\lim_{k\to\infty} x_k = \bar{x}$. Then $\{x_k\}$ converges superlinearly to $\bar{x}$ and $F(\bar{x}) = 0$ if and only if*

$$\lim_{k\to\infty} \frac{\|F(x_k) + \mathcal{T}_{\xi_k}\mathbb{D}F(\bar{x})\mathcal{T}_{\xi_k}^{-1}\eta_k\|}{\|\eta_k\|} = 0, \tag{5.2.2}$$

where $\xi_k \in \mathrm{T}_{\bar{x}}\,\mathcal{M}$ is defined by $\xi_k = R_{\bar{x}}^{-1}(x_k)$, i.e. $R_{\bar{x}}(\xi_k) = x_k$.

Proof. The proof essentially follows the proof in [GQA12, Theorem 14.1]. We simplify the procedures of getting (5.2.8) and (5.2.10).

Assume first that (5.2.2) holds. We first show that $\lim_{k\to\infty} \frac{\|\mathcal{T}_{\eta_k}^{-1}F(x_{k+1})\|}{\|\eta_k\|} = 0$, and then we show that this implies superlinear convergence of the sequence $\{x_k\}$. Adding and subtracting terms to $\mathcal{T}_{R_{\eta_k}}^{-1}F(x_{k+1})$ yields

$$\begin{aligned} \mathcal{T}_{\eta_k}^{-1}F(x_{k+1}) =& (F(x_k) + \mathcal{T}_{\xi_k}\mathbb{D}F(\bar{x})\mathcal{T}_{\xi_k}^{-1}\eta_k) - (-\mathcal{T}_{\eta_k}^{-1}F(x_{k+1}) + F(x_k) + \tilde{\mathbb{D}}F(x_k)\eta_k) \\ & - (\mathcal{T}_{\xi_k}\tilde{\mathbb{D}}F(\bar{x})\mathcal{T}_{\xi_k}^{-1} - \tilde{\mathbb{D}}F(x_k))\eta_k - \mathcal{T}_{\xi_k}(\mathbb{D}F(\bar{x}) - \tilde{\mathbb{D}}F(\bar{x}))\mathcal{T}_{\xi_k}^{-1}\eta_k, \end{aligned} \tag{5.2.3}$$

where $\tilde{\mathbb{D}}F(x)$ denotes the derivative at 0_x of the function $\mathrm{T}_x\,\mathcal{M} \to \mathrm{T}_x\,\mathcal{M} : \zeta \to \mathcal{T}_{\zeta}^{-1} F(R_x(\zeta))$. By Taylor's Theorem, we have

$$\lim_{k\to\infty} \frac{\|(-\mathcal{T}_{\eta_k}^{-1} F(x_{k+1}) + F(x_k) + \tilde{\mathbb{D}}F(x_k)\eta_k)\|}{\|\eta_k\|} = 0. \tag{5.2.4}$$

Because F is C^1, we have

$$\lim_{k\to\infty} \frac{\|(\mathcal{T}_{\xi_k}\tilde{\mathbb{D}}F(\bar{x})\mathcal{T}_{\xi_k}^{-1} - \tilde{\mathbb{D}}F(x_k))\eta_k\|}{\|\eta_k\|} = 0. \tag{5.2.5}$$

Since $\lim_{k\to\infty} x_k = \bar{x}$, we have $\lim_{k\to\infty}\|\eta_k\| = 0$ by Lemma 3.3.3 and thus

$$\lim_{k\to\infty} \|\mathcal{T}_{\xi_k}\mathbb{D}F(\bar{x})\mathcal{T}_{\xi_k}^{-1}\eta_k\| = 0.$$

From (5.2.2), we know $\lim_{k\to\infty}\|F(x_k)\| = 0$. So $F(\bar{x}) = 0$. Since $F(\bar{x}) = 0$, we have that $\tilde{\mathbb{D}}F(\bar{x}) = \mathbb{D}F(\bar{x})$, (by [AMS08, Page 96]), hence

$$\lim_{k\to\infty} \frac{\|\mathcal{T}_{\xi_k}(\mathbb{D}F(\bar{x}) - \tilde{\mathbb{D}}F(\bar{x}))\mathcal{T}_{\xi_k}^{-1}\eta_k\|}{\|\eta_k\|} = 0. \tag{5.2.6}$$

Applying (5.2.4), (5.2.5), (5.2.6) and (5.2.2) to (5.2.3) yields

$$\lim_{k\to\infty} \frac{\|\mathcal{T}_{\eta_k}^{-1} F(x_{k+1})\|}{\|\eta_k\|} = 0 \tag{5.2.7}$$

and it follows that

$$\begin{aligned}
\|\mathcal{T}_{\eta_k}^{-1} F(x_{k+1})\| &= \|\mathcal{T}_{\eta_k}^{-1} F(x_{k+1})\| - \|F(x_{k+1})\| + \|F(x_{k+1})\| \\
&\geq \|F(x_{k+1})\| - \left|\|\mathcal{T}_{\eta_k}^{-1} F(x_{k+1})\| - \|F(x_{k+1})\|\right| \\
&\geq \|F(x_{k+1})\|(1 - b_0 \operatorname{dist}(x_k, x_{k+1})) \text{ (by [GQA12, Lemma 14.3])} \\
&\geq b_1\|\xi_{k+1}\|(1 - b_0 \operatorname{dist}(x_k, x_{k+1})), \text{ (by Lemmas 3.3.7 and 3.3.3)}
\end{aligned} \tag{5.2.8}$$

where b_0, b_1 are some positive constants. We also have, by Lemma 3.3.3

$$b_2\|\eta_k\| \leq \operatorname{dist}(x_k, x_{k+1}) \leq \operatorname{dist}(x_k, \bar{x}) + \operatorname{dist}(x_{k+1}, \bar{x}) \leq b_3(\|\xi_k\| + \|\xi_{k+1}\|)$$
$$\|\eta_k\| \leq \frac{b_3}{b_2}(\|\xi_k\| + \|\xi_{k+1}\|),$$

where b_2 and b_3 are positive constants. Therefore, we obtain

$$\begin{aligned}
0 = \lim_{k\to\infty} \frac{\|\mathcal{T}_{\eta_k}^{-1} F(x_{k+1})\|}{\|\eta_k\|} &\geq \lim_{k\to\infty} \frac{b_1\|\xi_{k+1}\|}{\|\eta_k\|}(1 - b_0 \operatorname{dist}(x_k, x_{k+1})) \\
\geq \lim_{k\to\infty} \frac{b_1\|\xi_{k+1}\|}{2\|\eta_k\|} &\geq \lim_{k\to\infty} \frac{b_1\|\xi_{k+1}\|}{2b_3/b_2(\|\xi_k\| + \|\xi_{k+1}\|)} \\
= \lim_{k\to\infty} \frac{b_1\|\xi_{k+1}\|/\|\xi_k\|}{2b_3/b_2(1 + \|\xi_{k+1}\|/\|\xi_k\|)}&.
\end{aligned} \tag{5.2.9}$$

Thus, we have

$$\lim_{k\to\infty} \frac{\|\xi_{k+1}\|}{\|\xi_k\|} = 0,$$

which implies the superlinear convergence.

Conversely, assume that $\{x_k\}$ converges superlinearly to $\bar{x}$ and $F(\bar{x}) = 0$. By using (5.2.3), (5.2.4), (5.2.5) and (5.2.6), we need only to show that

$$\lim_{k\to\infty} \frac{\|\mathcal{T}_{\eta_k}^{-1} F(x_{k+1})\|}{\|\eta_k\|} = 0.$$

Noting that

$$\begin{aligned}
\|\eta_k\| &\geq \frac{1}{b_2} \operatorname{dist}(x_k, x_{k+1}) \geq \frac{1}{b_2} (\operatorname{dist}(x_k, \bar{x}) - \operatorname{dist}(x_{k+1}, \bar{x})) \geq \frac{b_3}{b_2} (\|\xi_k\| - \|\xi_{k+1}\|) \\
\|\eta_k\| &\geq \frac{1}{b_2} \operatorname{dist}(x_k, x_{k+1}) \geq \frac{1}{b_2} (\operatorname{dist}(x_{k+1}, \bar{x}) - \operatorname{dist}(x_k, \bar{x})) \geq \frac{b_3}{b_2} (\|\xi_{k+1}\| - \|\xi_k\|),
\end{aligned}$$

we have

$$\|\eta_k\| \geq \frac{b_3}{b_2} \left| \|\xi_{k+1}\| - \|\xi_k\| \right|.$$

It follows

$$\begin{aligned}
\|\mathcal{T}_{\eta_k}^{-1} F(x_{k+1})\| &= \|\mathcal{T}_{\eta_k}^{-1} F(x_{k+1})\| - \|F(x_{k+1})\| + \|F(x_{k+1})\| \\
&\leq \|F(x_{k+1})\| + \left| \|\mathcal{T}_{\eta_k}^{-1} F(x_{k+1})\| - \|F(x_{k+1})\| \right| \\
&\leq \|F(x_{k+1})\| (1 + b_4 \operatorname{dist}(x_k, x_{k+1})) \text{ (by [GQA12, Lemma 14.3])} \\
&\leq b_5 \|\xi_{k+1}\| (1 + b_4 \operatorname{dist}(x_k, x_{k+1})), \text{ (by Lemmas 3.3.7 and Lemma 3.3.3)} \qquad (5.2.10)
\end{aligned}$$

where b_4 and b_5 are some constants. We obtain

$$\begin{aligned}
\lim_{k\to\infty} \frac{\|\mathcal{T}_{\eta_k}^{-1} F(x_{k+1})\|}{\|\eta_k\|} &\leq \lim_{k\to\infty} \frac{b_5 \|\xi_{k+1}\| (1 + b_4 \operatorname{dist}(x_k, x_{k+1}))}{\frac{b_3}{b_2} \left| \|\xi_k\| - \|\xi_{k+1}\| \right|} \\
&\leq \lim_{k\to\infty} \frac{2 b_5 \|\xi_{k+1}\|}{\frac{b_3}{b_2} (\|\xi_k\| - \|\xi_{k+1}\|)} \\
&\leq \lim_{k\to\infty} \frac{2 b_5 b_2}{b_3} \frac{\|\xi_{k+1}\| / \|\xi_k\|}{1 - \|\xi_{k+1}\| / \|\xi_k\|} = 0
\end{aligned}$$

completing the proof. □

When $F(x_k) = -\mathcal{B}_k \eta_k$, e.g., for a quasi-Newton method, RDM2 implies RDM1 but is, in fact, more general since only $F \in C^1$ is assumed. If RDM2 is placed in a Euclidean setting the Dennis-Moré condition given in Theorem 5.2.3 results.

Theorem 5.2.3 (Euclidean Dennis Moré Condition 3 (EDM3)). *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be continuously differentiable in the open, convex set D in $\mathbb{R}^n$, and assume that for some $\bar{x}$ in D, and $F'(\bar{x})$ is nonsingular. Suppose that for some x_0 in D the sequence $\{x_k\}$ where*

$$x_{k+1} = x_k + s_k$$

remains in D and converges to $\bar{x}$. Then $\{x_k\}$ converges q-superlinearly to $\bar{x}$ and $F(\bar{x}) = 0$ if and only if

$$\lim_{k \to \infty} \frac{\|F(x_k) + F'(\bar{x}) s_k\|}{\|s_k\|} = 0.$$

If s_k in Theorem 5.2.3 is chosen to be $-B_k^{-1} F(x_k)$, Theorem 5.2.3 reduces to Theorem 5.1.1. Also note that since $F \in C^1$, $F'(\bar{x})$ can be replaced by $F'(x_k)$ by the triangle inequality of the norm to give an inexact Newton result for zeros of a Euclidean vector field. Finally, since F is a C^1 vector field and $\mathcal{T}$ is smooth, a similar Riemannian generalization for inexact Newton for zeros of a vector field given in the following corollary of Theorem 5.2.3 follows.

Corollary 5.2.1 (Riemannian Dennis-Moré Condition 3 (RDM3)). *Let $\mathcal{M}$ be a Riemannian manifold endowed with a retraction R. Let F be a C^1 tangent vector field on $\mathcal{M}$. Also let $\mathcal{M}$ be endowed with an affine connection ∇. Let $\mathbb{D}F(x)$ denote the linear transformation of $\mathrm{T}_x \mathcal{M}$ defined by $\mathbb{D}F(x)[\xi_x] = \nabla_{\xi_x} F$ for all tangent vector ξ_x to $\mathcal{M}$ at x. Let $\{\eta_k \in \mathrm{T}_{x_k} \mathcal{M}\}$ be a sequence of nonzero tangent vectors and $x_{k+1} = R_{x_k}(\eta_k)$. Assume that $\mathbb{D}F(\bar{x})$ is nonsingular, $x_k \neq \bar{x}, \forall k$, and* $\lim_{k \to \infty} x_k = \bar{x}$. *Then $\{x_k\}$ converges superlinearly to $\bar{x}$ and $F(\bar{x}) = 0$ if and only if*

$$\lim_{k \to \infty} \frac{\|F(x_k) + \mathbb{D}F(x_k)\eta_k\|}{\|\eta_k\|} = 0,$$

where $\xi_k \in \mathrm{T}_{\bar{x}} \mathcal{M}$ is defined by $\xi_k = R_{\bar{x}}^{-1}(x_k)$, i.e. $R_{\bar{x}}(\xi_k) = x_k$.

5.2.2 Riemannian Dennis-Moré Condition for a Real-valued Function

Our work needs a generalization of Lipschitz continuously differentiable on a manifold. Absil et. al. provides two generalizations, i.e., [AMS08, Definitions 7.4.1 and 7.4.3]. [AMS08, Definition 7.4.1] defines radially Lipschitz continuously differentiable for a function on the tangent bundle of a manifold. [AMS08, Definition 7.4.3] defines Lipschitz continuously differentiable for a function on a manifold. The latter relies on the exponential mapping and parallel translation. We propose Definition 5.2.1 which is more general than [AMS08, Definition 7.4.3] in the sense that an arbitrary pair of a retraction and a vector transport is used.

Definition 5.2.1 (Lipschitz continuously differentiable with respect to a vector transport)**.** *Let $\mathcal{M}$ be a Riemannian manifold endowed with a vector transport $\mathcal{T}$ and an associated retraction R. A function $f : \mathcal{M} \to \mathbb{R}$ is Lipschitz continuously differentiable with respect to $\mathcal{T}$ in $\mathcal{U} \subset \mathcal{M}$ if it is differentiable and if there exists a number $\kappa > 0$ such that, for all $x, y \in \mathcal{U}$, it holds that*

$$\| \operatorname{grad} f(y) - \mathcal{T}_\xi \operatorname{grad} f(x) \| \leq \kappa \|\xi\|,$$

where $\xi = R_x^{-1} y$.

Lemma 5.2.1 shows that a twice continuously differentiable function is Lipschitz continuously differentiable with respect to a given vector transport locally. This property is similar to the Euclidean definition.

Lemma 5.2.1. *If a function $f : \mathcal{M} \to \mathbb{R}$ is twice continuously differentiable, then for $\bar{x} \in \mathcal{M}$, and for any given vector transport $\mathcal{T}$, there exists a neighborhood of $\bar{x}$, $\mathcal{U}$, such that f is Lipschitz continuously differentiable with respect to $\mathcal{T}$ in $\mathcal{U}$.*

Proof. Choose $\mathcal{U}$ to be small enough such that $\mathcal{U}$ is a subset of a totally retractive neighborhood of $\bar{x}$. Therefore, for any $x, y \in \mathcal{U}$, $R_x^{-1} y$ is well-defined.

Define $z^{\mathrm{P}} = \operatorname{grad} f(y) - P_\gamma^{1 \leftarrow 0} \operatorname{grad} f(x)$, where P is parallel translation and $\gamma(t) = R_x(t\xi)$, i.e., the retraction line from x to y. From Lemma 3.3.8, we have

$$\|P_\gamma^{0 \leftarrow 1} z^{\mathrm{P}} - \bar{H}\xi\| \leq b_0 \|\xi\|^2,$$

which yields

$$\|z^{\mathrm{P}}\| \leq \|\bar{H}\xi\| + b_0 \|\xi\|^2,$$

where $\bar{H} = \int_0^1 P_\gamma^{0 \leftarrow t} \operatorname{Hess} f(\gamma(t)) P_\gamma^{t \leftarrow 0} dt$ and b_0 is a positive constant. It follows that

$$\begin{aligned}
\| \operatorname{grad} f(y) - \mathcal{T}_\xi \operatorname{grad} f(x) \| &= \|z^{\mathrm{P}} + P_\gamma^{1 \leftarrow 0} \operatorname{grad} f(x) - \mathcal{T}_\xi \operatorname{grad} f(x)\| \\
&\leq \|z^{\mathrm{P}}\| + \|P_\gamma^{1 \leftarrow 0} \operatorname{grad} f(x) - \mathcal{T}_\xi \operatorname{grad} f(x)\| \\
&\leq \|\bar{H}\xi\| + b_0 \|\xi\|^2 + b_1 \|\xi\| \| \operatorname{grad} f(x)\| \text{ (by Lemma 4.3.6)} \\
&\leq b_2 \|\xi\| + b_0 \|\xi\|^2 + b_1 b_3 \|\xi\| \text{ (since } \bar{H} \text{ and } \| \operatorname{grad} f(x)\| \text{ is bounded)}
\end{aligned}$$

which completes the proof. □

Lemma 5.2.2 is needed for the proof of Theorem 5.2.4.

Lemma 5.2.2. *Let $\mathcal{M}$ be a Riemannian manifold endowed with an isometric vector transport $\mathcal{T}_S$ and an associated retraction R. Assume that $\mathcal{T}_S \in C^0$ satisfies* (4.2.4) *and* (4.2.5) *and, along with $\mathcal{T}_R$, the differentiated retraction of R, satisfies the locking condition* (4.2.6). *If a function $f : \mathcal{M} \to \mathbb{R}$ is generalized Lipschitz continuously differentiable in $\mathcal{U}$, then there exists a number $\kappa_1 > 0$ such that, for all $x, y \in \mathcal{U}$, it holds that*

$$\|\beta^{-1} \operatorname{grad} f(y) - \mathcal{T}_{S_\xi} \operatorname{grad} f(x)\| \leq \kappa_1 \|\xi\|,$$

where $\xi = R_x^{-1} y$ and $\beta = \frac{\|\xi\|}{\|\mathcal{T}_{R_\xi} \xi\|}$.

Proof. We have

$$\begin{aligned}
\|\beta^{-1} \operatorname{grad} f(y) - \mathcal{T}_{S_\xi} \operatorname{grad} f(x)\| &\leq \|\beta^{-1} \operatorname{grad} f(y) - \operatorname{grad} f(y)\| + \|\operatorname{grad} f(y) - \mathcal{T}_{R_\xi} \operatorname{grad} f(x)\| \\
&\quad + \|\mathcal{T}_{R_\xi} \operatorname{grad} f(x) - \mathcal{T}_{S_\xi} \operatorname{grad} f(x)\| \\
&\leq b_0 \|\xi\| \|\operatorname{grad} f(y)\| \text{ (by Lemma 4.3.7)} \\
&\quad + b_1 \|\xi\| \text{ (since } T_R \text{ is a regular vector transport)} \\
&\quad + b_2 \|\xi\| \|\operatorname{grad} f(x)\| \text{ (by (4.2.4))} \\
&\leq b_3 \|\xi\| \text{ (since } \operatorname{grad} f(x) \text{ is bounded)}
\end{aligned}$$

where b_0, b_1, b_2 and b_3 are positive constants. □

The main result can now be stated and proven.

Theorem 5.2.4 (Riemannian Dennis Moré Condition 4 (RDM4))**.** *Let $\mathcal{M}$ be a Riemannian manifold endowed with a retraction R. Let $f : \mathcal{M} \to \mathbb{R}$ be twice continuously differentiable and consider the iteration $x_{k+1} = R_{x_k}(\alpha_k \eta_k), \eta_k \in \mathrm{T}_{x_k} \mathcal{M}$, where η_k is a descent direction and α_k satisfies the Wolfe conditions* (4.2.1) *and* (4.2.2). *Assume the sequence $\{x_k\}$ converges to a point $\bar{x}$ such that* $\operatorname{Hess} f(\bar{x})$ *is positive definite. If the search direction satisfies*

$$\lim_{k \to \infty} \frac{\|\operatorname{grad} f(x_k) + \operatorname{Hess} f(x_k) \eta_k\|}{\|\eta_k\|} = 0, \tag{5.2.11}$$

then there is an index $k_0 \geq 0$ such that $\alpha_k = 1$ is admissible for $k \geq k_0$. Moreover, $\operatorname{grad} f(\bar{x}) = 0$ *and if $\alpha_k = 1$ for all $k \geq k_0$, then $\{x_k\}$ converges superlinearly to $\bar{x}$.*

Proof. The proof is generalized from [DS83, Theorem 6.3.3 and Theorem 6.3.4]. Lemmas 5.2.1 and 5.2.2 show that if f is C^2, then it is Lipschitz continuously differentiable with respect to a given vector transport. This property reduces to well-known fact in the Euclidean setting.

First, we show $\lim_{k\to\infty}(g(\operatorname{grad} f(x_k), \eta_k))/(\|\eta_k\|) = 0$. Applying the first Wolfe condition (4.2.1), we have

$$-\infty < f(x_j) - f(x_0) = \sum_{k=0}^{j-1}(f(x_{k+1}) - f(x_k)) \leq c_1 \sum_{k=0}^{j-1} g(\operatorname{grad} f(x_k), \alpha_k \eta_k) < 0,$$

which yields

$$-\sum_{k=0}^{\infty} g(\operatorname{grad} f(x_k), \alpha_k \eta_k) < \infty. \tag{5.2.12}$$

By results in Chapter 4, there exists an isometric vector transport $\mathcal{T}_S \in C^0$ associated with R that satisfies (4.2.4) and (4.2.5) and along with $\mathcal{T}_R$, the differentiated retraction, satisfies the locking condition (4.2.6). Define $s_k = \mathcal{T}_{S_{\alpha_k \eta_k}} \alpha_k \eta_k$ and $y_k = \beta_k^{-1} \operatorname{grad} f(x_{k+1}) - \mathcal{T}_{S_{\alpha_k \eta_k}} \operatorname{grad} f(x_k)$, where $\beta_k = \frac{\|\alpha_k \eta_k\|}{\|\mathcal{T}_{R_{\alpha_k \eta_k}} \alpha_k \eta_k\|}$. From the second Wolfe condition (4.2.2) and (4.2.8), we have

$$(c_2 - 1)\alpha_k g(\operatorname{grad} f(x_k), \eta_k) \leq g(s_k, y_k)$$

and combining with the Cauchy Schwarz inequality, it follows that

$$(c_2 - 1)\alpha_k g(\operatorname{grad} f(x_k), \eta_k) \leq \|s_k\| \|y_k\|. \tag{5.2.13}$$

By Lemmas 5.2.1 and 5.2.2, we have $\|y_k\| \leq b_0 \|s_k\|$, where b_0 is a positive constant. Plugging it into (5.2.13), we have

$$(c_2 - 1)\alpha_k g(\operatorname{grad} f(x_k), \eta_k) \leq b_0 \|s_k\|^2.$$

From (5.2.12) and above inequality, we have

$$0 = \lim_{k\to\infty} g(\operatorname{grad} f(x_k), \alpha_k \eta_k) \leq \lim_{k\to\infty} \frac{c_2 - 1}{b_0} (\frac{g(\operatorname{grad} f(x_k), \alpha_k \eta_k)}{\|s_k\|})^2 \leq 0.$$

Since $\|s_k\| = \|\alpha_k \eta_k\|$, we have

$$\lim_{k\to\infty} \frac{g(\operatorname{grad} f(x_k), \eta_k)}{\|\eta_k\|} = 0,$$

which is desired result.

The next step is to show $\|\eta_k\| \to 0$. We have

$$-\frac{g(\operatorname{grad} f(x_k), \eta_k)}{\|\eta_k\|} = \frac{g(\eta_k, \operatorname{Hess} f(x_k)\eta_k)}{\|\eta_k\|} - \frac{g(\eta_k, \operatorname{grad} f(x_k) + \operatorname{Hess} f(x_k)\eta_k)}{\|\eta_k\|}.$$

Let $\mu^{-1} = \|\operatorname{Hess} f(\bar{x})^{-1}\|$. When x_k close to $\bar{x}$ enough, we have $\mu^{-1}/2 \leq \|\operatorname{Hess} f(x_k)^{-1}\| \leq 2\mu^{-1}$. Using (5.2.11) and k large enough,

$$-\frac{g(\operatorname{grad} f(x_k), \eta_k)}{\|\eta_k\|} \geq (\frac{1}{2}\mu - \frac{\|\operatorname{grad} f(x_k) + \operatorname{Hess} f(x_k)\eta_k\|}{\|\eta_k\|})\|\eta_k\| \geq \frac{1}{4}\mu\|\eta_k\|, \tag{5.2.14}$$

which implies $\|\eta_k\| \to 0$.

Now, we can show the main results of the theorem. Consider the lifting function. $\hat{f}_k$ denotes $f \circ R_{x_k}$ and $\hat{f}_*$ denotes $f \circ R_{\bar{x}}$. By Taylor's Theorem, we have

$$\hat{f}_k(\eta_k) - \hat{f}_k(0) = g(\operatorname{grad} f(x_k), \eta_k) + \frac{1}{2}g(\eta_k, \operatorname{Hess} \hat{f}_k(p_k)[\eta_k])$$

where $p_k = t\eta_k$, for some $t \in [0,1]$. Let ξ_k denote $R_{x_k}^{-1}\bar{x}$. It follows that

$$\begin{aligned}
&\hat{f}_k(\eta_k) - \hat{f}_k(0) - \frac{1}{2}g(\operatorname{grad} f(x_k), \eta_k) \\
&= \frac{1}{2}g(\operatorname{grad} f(x_k), \eta_k) + \frac{1}{2}g(\eta_k, \operatorname{Hess} \hat{f}_k(p_k)[\eta_k]) \\
&= \frac{1}{2}g(\operatorname{grad} f(x_k) + \operatorname{Hess} f(x_k)\eta_k, \eta_k) \\
&\quad + \frac{1}{2}g((\mathcal{T}_{S_{\xi_k}} \operatorname{Hess} f(\bar{x})\mathcal{T}_{S_{\xi_k}}^{-1} - \operatorname{Hess} f(x_k))[\eta_k], \eta_k) \\
&\quad + \frac{1}{2}g((\operatorname{Hess} \hat{f}_k(p_k) - \mathcal{T}_{S_{\xi_k}} \operatorname{Hess} \hat{f}_*(0)\mathcal{T}_{S_{\xi_k}}^{-1})[\eta_k], \eta_k) \text{ (by [AMS08, Proposition 5.5.6])} \\
&\leq \frac{1}{2}(\frac{\|\operatorname{grad} f(x_k) + \operatorname{Hess} f(x_k)\eta_k\|}{\|\eta_k\|} + o_1(\|\xi_k\|) + o_1(\|\eta_k\|) + o_1(\|\xi_k\|))\|\eta_k\|^2,
\end{aligned} \tag{5.2.15}$$

where $o_1(t)$ denotes $o(1)$ with respect to t, i.e., $\lim_{t\to 0} o_1(t) = 0$. Since $\operatorname{Hess} f(x)$ and $\operatorname{Hess} \hat{f}_x(\eta_x)$ are continuous, (5.2.15) holds. Choosing k_0 large enough so that for all $k \geq k_0$, (5.2.14) and

$$\frac{\|\operatorname{grad} f(x_k) + \operatorname{Hess} f(x_k)\eta_k\|}{\|\eta_k\|} + o_1(\|\xi_k\|) + o_1(\|\eta_k\|) + o_1(\|\xi_k\|) \leq \frac{1}{4}\mu \min(c_2, 1 - 2c_1) \tag{5.2.16}$$

hold and using (5.2.15), we have

$$\begin{aligned}
f(R_{x_k}(\eta_k)) - f(x_k) &= \hat{f}_k(\eta_k) - \hat{f}_k(0) \\
&\leq \frac{1}{2}g(\operatorname{grad} f(x_k), \eta_k) + \frac{1}{8}\mu(1 - 2c_1)\|\eta_k\|^2 \text{ (by (5.2.16))} \\
&\leq \frac{1}{2}(1 - (1 - 2c_1))g(\operatorname{grad} f(x_k), \eta_k) \text{ (by (5.2.14))} \\
&= c_1 g(\operatorname{grad} f(x_k), \eta_k),
\end{aligned}$$

which means $\alpha_k = 1$ satisfies (4.2.1). Similarly, we have

$$\begin{aligned}
&\frac{d}{dt} f(R_{x_k}(t\eta_k))|_{t=1} \\
&= \frac{d}{dt} \hat{f}_k(t\eta_k)|_{t=1} \\
&= \frac{d}{dt} \hat{f}_k(t\eta_k)|_{t=0} + g(\eta_k, \operatorname{Hess} \hat{f}_k(a\eta_k))[\eta_k]) \text{ where } a \in [0,1] \\
&= g(\operatorname{grad} f(x_k), \eta_k) + g(\eta_k, \operatorname{Hess} \hat{f}_k(a\eta_k)[\eta_k]) \\
&\leq (\frac{\| \operatorname{grad} f(x_k) + \operatorname{Hess} f(x_k)\eta_k \|}{\|\eta_k\|} + o_1(\|\xi_k\|) + o_1(\|\eta_k\|) + o_1(\|\xi_k\|))\|\eta_k\|^2 \text{ (similar to (5.2.15))} \\
&\leq \frac{\mu c_2}{4} \|\eta_k\|^2 \text{ (by (5.2.16))} \\
&\leq -c_2 g(\operatorname{grad} f(x_k), \eta_k) \text{ (by (5.2.14))} \\
&= -c_2 \frac{d}{dt} f(R_{x_k}(t\eta_k))|_{t=0}.
\end{aligned}$$

Therefore, $\alpha_k = 1$ satisfies the Wolfe conditions eventually. Superlinear convergence can be obtained by applying Corollary 5.2.1 with F taken to be $\operatorname{grad} f$. □

CHAPTER 6

CONVERGENCE RATE ANALYSIS OF THE RIEMANNIAN BROYDEN FAMILY METHOD

6.1 Introduction

In the Euclidean setting, the history of the investigation of quasi-Newton method is rich and there are many important papers such as Dennis and Moré [DM74] [DM77], Stoer [Sto75], Powell [Pow76] [Pow86], Schnabel [Sch78], Ritter [Rit79] [Rit81], Stachursky [Sta81], Griewank and Toint [GT82], Byrd, Nocedal and Yuan [BNY87] and Byrd, Liu and Nocedal [BLN92]. However, in the Riemannian setting, the literature on convergence analysis of quasi-Newton methods is still limited. Riemannian quasi-Newton methods have been used for various applications (see Chapter 4) without systematic convergence analysis. There are two recent attempts to provide a complete analysis of the convergence of Riemannian quasi-Newton methods. Qi [Qi11] analyzes the convergence of RBFGS with exponential mapping and parallel translation and Ring and Wirth [RW12] provide convergence analysis for their particular version of RBFGS.

Since the global convergence is shown in Chapter 4, the analyses in this chapter assume the iteration is converging to an isolated minimizer x^*. Additional assumptions need only hold in a neighborhood of x^*, denoted by $\mathcal{S}$. The following notation is added to that of Chapter 4:

$$\epsilon_k = \max(\mathrm{dist}(x_{k+1}, x^*), \mathrm{dist}(x_k, x^*)), \;\; H_* = \mathrm{Hess}\, f(x^*), \;\; \zeta_k = R_{x^*}^{-1} x_k,$$
$$H_k = \mathcal{T}_{S_{\zeta_k}} H_* \mathcal{T}_{S_{\zeta_k}}^{-1}, \;\; \bar{s}_k = H_{k+1}^{1/2} s_k, \;\; \bar{y}_k = H_{k+1}^{-1/2} y_k,$$
$$\bar{\mathcal{B}}_k = H_k^{-1/2} \mathcal{B}_k H_k^{-1/2}, \;\; \mathcal{C}_k = H_{k+1}^{-1/2} \tilde{\mathcal{B}}_k H_{k+1}^{-1/2}, \;\; \cos \bar{\theta}_k = \frac{g(\bar{s}_k, \mathcal{C}_k \bar{s}_k)}{\|\bar{s}_k\| \|\mathcal{C}_k \bar{s}_k\|},$$

where $H_k^{1/2}$ denotes a linear operator on $\mathrm{T}_{x_k} \mathcal{M}$ that satisfies $H_k^{1/2} H_k^{1/2} = H_k$ and is self-adjoint. The existent of $H_k^{1/2}$ follows by Lemma 6.2.3.

This chapter contains two main results on the rate of convergence of Algorithm 3. In Section 6.2.1, it is shown to be R-linear convergent. Using this result and a slightly strengthened assumption on the continuity of the Hessian, superlinear convergence is shown in Section 6.2.2.

6.2 The RBroyden Family Convergence Rate Analysis

6.2.1 R-Linear Convergence Analysis

Since we consider a general retraction, a generalization of the Euclidean triangle inequality in $\mathcal{S}$ must be assumed. As shown in Lemma 6.2.1, choosing the exponential mapping for R implies Assumption 6.2.1.

Assumption 6.2.1. *There is a constant c_3 such that for all $x, y \in \mathcal{S}$,*

$$\max_{t\in[0,1]} \operatorname{dist}(R_x(t\eta), x^*) \leq c_3 \max(\operatorname{dist}(x, x^*), \operatorname{dist}(y, x^*)),$$

where $\eta = R_x^{-1} y$.

Lemma 6.2.1. *Let $\mathcal{U}$ be an open set of a Riemannian manifold $\mathcal{M}$ such that for any $p, q \in \mathcal{U}$, there exists a unique minimum geodesic γ from p to q and $\gamma \subset \mathcal{U}$. If x, y and z are $\mathcal{U}$, then the inequality*

$$\max_{t\in[0,1]} \operatorname{dist}(\operatorname{Exp}_x(t\eta), z) \leq 2\max(\operatorname{dist}(x, z), \operatorname{dist}(y, z))$$

holds, where $\eta = \operatorname{Exp}_x^{-1} y$.

Proof. Let $p(t)$ denote $\operatorname{Exp}_x(t\eta)$. Since the distance function satisfies the triangle inequality, we have

$$\begin{aligned}
\operatorname{dist}(p(t), z) &\leq \operatorname{dist}(p(t), x) + \operatorname{dist}(x, z) \\
\operatorname{dist}(p(t), z) &\leq \operatorname{dist}(p(t), y) + \operatorname{dist}(y, z).
\end{aligned}$$

By adding above inequalities, we have

$$\begin{aligned}
\operatorname{dist}(p(t), z) &\leq \frac{1}{2}(\operatorname{dist}(p(t), x) + \operatorname{dist}(x, z) + \operatorname{dist}(p(t), y) + \operatorname{dist}(y, z)) \\
&= \frac{1}{2}(\operatorname{dist}(x, y) + \operatorname{dist}(x, z) + \operatorname{dist}(y, z)) \text{ (since } p(t) \text{ is on the shortest geodesic.)} \\
&\leq \frac{1}{2}(\operatorname{dist}(x, z) + \operatorname{dist}(z, y) + \operatorname{dist}(x, z) + \operatorname{dist}(y, z)) \\
&= \operatorname{dist}(x, z) + \operatorname{dist}(y, z) \\
&\leq 2\max(\operatorname{dist}(x, z), \operatorname{dist}(y, z))
\end{aligned}$$

□

The R-linear convergence analysis in the Euclidean case, given in [BNY87, §4], relies on the change of variables from x to $x^* + (\operatorname{Hess} f(x^*))^{1/2}(x - x^*)$. Such a change of variables is not legitimate when the Euclidean space is replaced by a general Riemannian manifold. For this and other reasons, notably the presence of $\tilde{\mathcal{B}}_k$ rather than $\mathcal{B}_k$ in the Broyden update equation (4.2.3), the generalization of the analysis in [BNY87, §4] to the Riemannian case requires considerably more effort than a mere "mutatis mutandis" modification. The differences are highlighted in the following proofs.

Lemma 6.2.2 generalizes [BNY87, (4.3)].

Lemma 6.2.2. *If Assumptions 4.3.1 and 6.2.1 hold, then equation*

$$\|y_k - H_{k+1}s_k\| = \|s_k\| o_1(\epsilon_k)$$

holds, where $o_1(t)$ denotes $o(1)$ with respect to t, i.e., $\lim_{t\to 0} o_1(t) = 0$.

Proof. Define $y_k^{\mathrm{P}} = \operatorname{grad} f(x_{k+1}) - P_{\gamma_k}^{1\leftarrow 0} \operatorname{grad} f(x_k)$, where P is parallel transport and γ_k is the retraction line from x_k to x_{k+1}. From Lemma 3.3.8, we have

$$\|P_{\gamma_k}^{0\leftarrow 1} y_k^{\mathrm{P}} - \bar{H}_k \alpha_k \eta_k\| \leq b_0 \|\alpha_k \eta_k\|^2 = b_0 \|s_k\|^2,$$

where $\bar{H}_k = \int_0^1 P_{\gamma_k}^{0\leftarrow t} \operatorname{Hess} f(\gamma_k(t)) P_{\gamma_k}^{t\leftarrow 0} dt$ and b_0 is a positive constant. It follows that

$$\begin{aligned}
&\|y_k - H_{k+1}s_k\| \\
&\leq \|y_k - y_k^{\mathrm{P}}\| + \|P_{\gamma_k}^{0\leftarrow 1} y_k^{\mathrm{P}} - \bar{H}_k \alpha_k \eta_k\| + \|P_{\gamma_k}^{1\leftarrow 0} \bar{H}_k P_{\gamma_k}^{0\leftarrow 1} P_{\gamma_k}^{1\leftarrow 0} \alpha_k \eta_k - H_{k+1} \mathcal{T}_{S_{\alpha_k \eta_k}} \alpha_k \eta_k\| \\
&\leq \| \operatorname{grad} f(x_{k+1})/\beta_k - \operatorname{grad} f(x_{k+1})\| + \|P_{\gamma_k}^{1\leftarrow 0} \operatorname{grad} f(x_k) - \mathcal{T}_{S_{\alpha_k \eta_k}} \operatorname{grad} f(x_k)\| \\
&\quad + b_0 \|s_k\|^2 + \|P_{\gamma_k}^{1\leftarrow 0} \bar{H}_k P_{\gamma_k}^{0\leftarrow 1} P_{\gamma_k}^{1\leftarrow 0} \alpha_k \eta_k - P_{\gamma_k}^{1\leftarrow 0} \bar{H}_k P_{\gamma_k}^{0\leftarrow 1} \mathcal{T}_{S_{\alpha_k \eta_k}} \alpha_k \eta_k\| \\
&\quad + \|P_{\gamma_k}^{1\leftarrow 0} \bar{H}_k P_{\gamma_k}^{0\leftarrow 1} \mathcal{T}_{S_{\alpha_k \eta_k}} \alpha_k \eta_k - H_{k+1} \mathcal{T}_{S_{\alpha_k \eta_k}} \alpha_k \eta_k\| \\
&= \| \operatorname{grad} f(x_{k+1})\| |1/\beta_k - 1| \text{ (using Lemmas 3.3.7 and 4.3.7)} \\
&\quad + \|P_{\gamma_k}^{1\leftarrow 0} \operatorname{grad} f(x_k) - \mathcal{T}_{S_{\alpha_k \eta_k}} \operatorname{grad} f(x_k)\| \text{ (using Lemmas 3.3.7 and 4.3.6)} \\
&\quad + b_0 \|s_k\|^2 + \|\bar{H}_k\| \|P_{\gamma_k}^{1\leftarrow 0} \alpha_k \eta_k - \mathcal{T}_{S_{\alpha_k \eta_k}} \alpha_k \eta_k\| \text{ (using Lemma 4.3.6)} \\
&\quad + \|P_{\gamma_k}^{1\leftarrow 0} \bar{H}_k P_{\gamma_k}^{0\leftarrow 1} - H_{k+1}\| \|s_k\| \text{ (using Assumption 6.2.1)} \\
&\leq b_1 \epsilon_k \|s_k\| + b_2 \epsilon_k \|s_k\| + b_3 \epsilon_k \|s_k\| + o_1(\epsilon_k) \|s_k\| \qquad (6.2.1) \\
&= o_1(\epsilon_k) \|s_k\|
\end{aligned}$$

where b_1, b_2, b_3 are positive constants. Therefore, we have

$$\|y_k - H_{k+1}s_k\| = \|s_k\| o_1(\epsilon_k),$$

□

If a matrix B is symmetric positive definite for Euclidean metric, then it is easily seen that there exists a symmetric positive definite matrix $B^{1/2}$ such that $B = B^{1/2}B^{1/2}$. Lemma 6.2.3 shows that this property holds as well for an arbitrary metric. This property allows a decomposition of the Hessian at x^*, i.e., $H_* = H_*^{1/2}H_*^{1/2}$, $H_*^{1/2}$ is self-adjoint.

Lemma 6.2.3. *Let $\langle u, v\rangle = u^T G v$ be an inner product of $\mathbb{R}^d$, where G is a symmetric positive definite matrix. If B is positive definite and self-adjoint with respect to this inner product, in other words,*

$$\langle Bu, v\rangle = \langle u, Bv\rangle \text{ and } \langle Bu, u\rangle > 0$$

for all u, v, then there exists a matrix A such that $B = AA$ and A is self-adjoint.

Proof. Let $P = GB$. Since B is positive definite and self-adjoint with respect to the inner product, P is a symmetric positive definite matrix. Therefore, there exists a matrix L such that $L^T L = P$. Because G is a symmetric positive definite matrix, there exists a symmetric matrix $G^{1/2}$ such that $G^{1/2}G^{1/2} = G$. Let U and V be from singular value decomposition: $LG^{-1/2} = USV^T$. We will show that

$$A = G^{-1/2}VU^T L.$$

First, we have

$$\begin{aligned} AA &= G^{-1/2}VU^T LG^{-1/2}VU^T L = G^{-1/2}VU^T USV^T VU^T L = G^{-1/2}VSU^T L \\ &= G^{-1/2}G^{-1/2}L^T L = G^{-1}P = B. \end{aligned}$$

In order to show A is self-adjoint, we only need to show GA is symmetric. Noticing that $U^T L = SV^T G^{1/2}$, we have

$$GA - A^T G = G^{1/2}VU^T L - L^T UV^T G^{1/2} = G^{1/2}VSV^T G^{1/2} - G^{1/2}VSV^T G^{1/2} = 0.$$

□

Lemma 6.2.4 generalizes [BNY87, Lemma 4.1].

Lemma 6.2.4. *Suppose Assumptions 4.3.1, 4.3.2 and 6.2.1 hold. For any $0 < \epsilon \leq 1$, there is a neighborhood $N(x^*)$ of x^* such that if x_k and x_{k+1} generated by Algorithm 3 are in $N(x^*)$, then*

$$\frac{g(\bar{y}_k, \bar{y}_k)}{g(\bar{y}_k, \bar{s}_k)} \frac{g(\bar{s}_k, \mathcal{C}_k \bar{s}_k)}{g(\bar{y}_k, \bar{s}_k)} - 2\frac{g(\bar{y}_k, \mathcal{C}_k \bar{s}_k)}{g(\bar{y}_k, \bar{s}_k)} \leq \frac{a_{14}\epsilon\alpha_k}{\cos\bar{\theta}_k},$$

where a_{14} is a positive constant.

Proof. From Lemma 6.2.2, we know that

$$\|\bar{y}_k - \bar{s}_k\| = o_1(\epsilon_k)\|\bar{s}_k\|, \tag{6.2.2}$$

Therefore, we have

$$\|\bar{y}_k\| = (1 + o_1(\epsilon_k))\|\bar{s}_k\|. \tag{6.2.3}$$

By squaring (6.2.2) and using (6.2.3), we have

$$(1 + o_1(\epsilon_k))^2\|\bar{s}_k\|^2 - 2g(\bar{y}_k, \bar{s}_k) + \|\bar{s}_k\|^2 = \|\bar{y}_k\|^2 - 2g(\bar{y}_k, \bar{s}_k) + \|\bar{s}_k\|^2 = (o_1(\epsilon_k))^2\|\bar{s}_k\|^2,$$

and therefore

$$g(\bar{y}_k, \bar{s}_k) = (1 + o_1(\epsilon_k))\|\bar{s}_k\|^2. \tag{6.2.4}$$

Thus, we know

$$\frac{g(\bar{y}_k, \bar{s}_k)}{\|\bar{s}_k\|^2} = 1 + o_1(\epsilon_k) \tag{6.2.5}$$

Combining (6.2.3) and (6.2.5), we obtain

$$\frac{\|\bar{y}_k\|^2}{g(\bar{y}_k, \bar{s}_k)} = 1 + o_1(\epsilon_k). \tag{6.2.6}$$

Let $N(\bar{x})$ be sufficiently small, meaning take ϵ_k small enough so that

$$\frac{g(\bar{y}_k, \bar{y}_k)}{g(\bar{y}_k, \bar{s}_k)} \leq 1 + \epsilon \text{ and } \frac{\|\bar{y}_k - \bar{s}_k\|}{\|\bar{s}_k\|} \leq \epsilon \tag{6.2.7}$$

Observe that $\mathcal{C}_k$ is positive definite, given the positive-definiteness of $\tilde{\mathcal{B}}_k$. It follows that

$$
\begin{aligned}
&\frac{g(\bar{y}_k, \bar{y}_k)}{g(\bar{y}_k, \bar{s}_k)} \frac{g(\bar{s}_k, \mathcal{C}_k \bar{s}_k)}{g(\bar{y}_k, \bar{s}_k)} - 2 \frac{g(\bar{y}_k, \mathcal{C}_k \bar{s}_k)}{g(\bar{y}_k, \bar{s}_k)} \\
&= (\frac{g(\bar{y}_k, \bar{y}_k)}{g(\bar{y}_k, \bar{s}_k)} - 2) \frac{g(\bar{s}_k, \mathcal{C}_k \bar{s}_k)}{g(\bar{y}_k, \bar{s}_k)} - 2 \frac{g(\bar{y}_k - \bar{s}_k, \mathcal{C}_k \bar{s}_k)}{g(\bar{y}_k, \bar{s}_k)} \\
&\leq (\epsilon - 1) \frac{g(\bar{s}_k, \mathcal{C}_k \bar{s}_k)}{g(\bar{y}_k, \bar{s}_k)} + \frac{2\epsilon \|\bar{s}_k\| \|\mathcal{C}_k \bar{s}_k\|}{a_0 \|\bar{s}_k\|^2} \text{ (by (6.2.7) and Lemma 4.3.4)} \\
&\leq \frac{2\epsilon \|\mathcal{C}_k \bar{s}_k\|}{a_0 \|\bar{s}_k\|} \text{ (since } \epsilon \leq 1, \mathcal{C}_k \text{ is positive definite and } g(\bar{s}_k, \bar{y}_k) = g(s_k, y_k) > 0) \\
&\leq \frac{2\|H_*^{-1/2}\| \epsilon \alpha_k \| \operatorname{grad} f(x_k)\|}{a_0 \|\bar{s}_k\|} \text{ (by definitions of } \mathcal{C}_k \text{ and } \bar{s}_k) \\
&= \frac{a_{14} \epsilon \alpha_k}{\cos \bar{\theta}_k}, \text{ (by Lemma 4.3.5)}
\end{aligned}
$$

where $a_{14} = 2\|H_*^{-1/2}\| / (a_0 a_2)$. □

Lemma 6.2.5 proves the same results as Lemma 3.3.10 but does not require the vector transport to be smooth.

Lemma 6.2.5. *The isometric vector transport $\mathcal{T}_S \in C^0$ and $\mathcal{T}_S$ satisfies* (4.2.4) *and* (4.2.5)*. Let $\bar{x} \in \mathcal{M}$. Then there is a neighborhood $\mathcal{U}$ of $\bar{x}$ and $\tilde{a}_{14}$ such that for all $x, y \in \mathcal{U}$,*

$$
\begin{aligned}
\| \operatorname{id} - \mathcal{T}_{S_\xi}^{-1} \mathcal{T}_{S_\eta}^{-1} \mathcal{T}_{S_\zeta} \| &\leq \tilde{a}_{14} \max(\operatorname{dist}(x, \bar{x}), \operatorname{dist}(y, \bar{x})), \\
\| \operatorname{id} - \mathcal{T}_{S_\zeta}^{-1} \mathcal{T}_{S_\eta} \mathcal{T}_{S_\xi} \| &\leq \tilde{a}_{14} \max(\operatorname{dist}(x, \bar{x}), \operatorname{dist}(y, \bar{x})),
\end{aligned}
$$

where $\xi = R_{\bar{x}}^{-1} x$, $\eta = R_x^{-1} y$ and $\zeta = R_{\bar{x}}^{-1} y$.

Proof. By applying Lemma 3.3.10 for the differentiated retraction $\mathcal{T}_R$, we obtain

$$
\| \operatorname{id} - \mathcal{T}_{R_\xi}^{-1} \mathcal{T}_{R_\eta}^{-1} \mathcal{T}_{R_\zeta} \| \leq b_0 \max(\operatorname{dist}(x, \bar{x}), \operatorname{dist}(y, \bar{x})).
$$

where b_0 is a positive constant. It follows that

$$
\begin{aligned}
&\| \operatorname{id} - \mathcal{T}_{S_\xi}^{-1} \mathcal{T}_{S_\eta}^{-1} \mathcal{T}_{S_\zeta} \| \\
&\leq \| \operatorname{id} - \mathcal{T}_{R_\xi}^{-1} \mathcal{T}_{R_\eta}^{-1} \mathcal{T}_{R_\zeta} \| + \| \mathcal{T}_{R_\xi}^{-1} \mathcal{T}_{R_\eta}^{-1} \mathcal{T}_{R_\zeta} - \mathcal{T}_{S_\xi}^{-1} \mathcal{T}_{R_\eta}^{-1} \mathcal{T}_{R_\zeta} \| \\
&\quad + \| \mathcal{T}_{S_\xi}^{-1} \mathcal{T}_{R_\eta}^{-1} \mathcal{T}_{R_\zeta} - \mathcal{T}_{S_\xi}^{-1} \mathcal{T}_{S_\eta}^{-1} \mathcal{T}_{R_\zeta} \| + \| \mathcal{T}_{S_\xi}^{-1} \mathcal{T}_{S_\eta}^{-1} \mathcal{T}_{R_\zeta} - \mathcal{T}_{S_\xi}^{-1} \mathcal{T}_{S_\eta}^{-1} \mathcal{T}_{S_\zeta} \| \\
&\leq b_0 \max(\operatorname{dist}(x, \bar{x}), \operatorname{dist}(y, \bar{x})) + b_1 \|\xi\| + b_2 \|\eta\| + b_3 \|\zeta\| \text{ (by (4.2.4) and (4.2.5))} \\
&\leq b_4 \max(\operatorname{dist}(x, \bar{x}), \operatorname{dist}(y, \bar{x}))
\end{aligned}
$$

where b_0, b_1, b_2, b_3 and b_4 are positive constants and the first inequality follows. The second is shown by a similar argument. □

In the R-linear convergence analysis, the Frobenius norm is used. However, this norm is not independent of basis of the tangent space. Lemma 6.2.6 shows that the Frobenius norm is equivalent to the norm induced by the Riemannian metric of the manifold in a compact set on the manifold.

Lemma 6.2.6. *Let $\mathcal{M}$ be a Riemannian manifold endowed with a metric g. Let $\bar{x} \in \mathcal{M}$ and $\mathcal{U}$ be a compact neighborhood of $\bar{x}$. Then there exist constants $M > m > 0$ such that for all $x \in \mathcal{U}$ and all linear transformations $\mathcal{A}_x$ of $\mathrm{T}_x\,\mathcal{M}$, we have*

$$m\|\mathcal{A}_x\| \leq \|\hat{\mathcal{A}}_x\|_F \leq M\|\mathcal{A}_x\|.$$

Moreover, if the chosen basis of $\mathrm{T}_x\,\mathcal{M}$ is orthonormal with respect to the Riemannian metric g (so that the matrix expression of the metric at x is the identity), then the bounds hold with $m = 1$ and $M = \sqrt{d}$.

Proof. Let G_x denote the matrix expression of inner production, i.e. $g(w_x, v_x) = \hat{w}_x^T G_x \hat{v}_x$. For any $v_x \in \mathrm{T}_x\,\mathcal{M}$, we have

$$\begin{aligned}\frac{\|\mathcal{A}_x v_x\|^2}{\|v_x\|^2} &= \frac{\mathrm{trace}(\hat{v}_x^T \hat{\mathcal{A}}_x^T G_x \hat{\mathcal{A}}_x \hat{v}_x)}{\mathrm{trace}(\hat{v}_x^T G_x \hat{v}_x)} \\ &= \frac{\mathrm{trace}(\hat{u}_x^T G_x^{-T/2} \hat{\mathcal{A}}_x^T G_x^{T/2} G_x^{1/2} \hat{\mathcal{A}}_x G_x^{-1/2} \hat{u}_x)}{\mathrm{trace}(\hat{u}_x^T \hat{u}_x)} \quad (\text{letting } \hat{u}_x = G_x^{1/2} \hat{v}_x) \\ &= \frac{\|G_x^{1/2} \hat{\mathcal{A}}_x G_x^{-1/2} \hat{u}_x\|_2^2}{\|\hat{u}_x\|_2^2}\end{aligned}$$

On one hand, let v_x satisfy $\frac{\|\mathcal{A}_x v_x\|}{\|v_x\|} = \|\mathcal{A}_x\|$. Using $\|\hat{\mathcal{A}}_x\|_2 \leq \|\hat{\mathcal{A}}_x\|_F$ ([GV96, (2.3.7)]), we have

$$\begin{aligned}\|\mathcal{A}_x\| = \frac{\|\mathcal{A}_x v_x\|}{\|v_x\|} &= \frac{\|G_x^{1/2} \hat{\mathcal{A}}_x G_x^{-1/2} \hat{u}_x\|_2}{\|\hat{u}_x\|_2} \leq \|G_x^{1/2} \hat{\mathcal{A}}_x G_x^{-1/2}\|_2 \\ &\leq \|G_x^{1/2}\|_2 \|\hat{\mathcal{A}}_x\|_2 \|G_x^{-1/2}\|_2 \leq \|G_x^{1/2}\|_2 \|G_x^{-1/2}\|_2 \|\hat{\mathcal{A}}_x\|_F \leq \frac{1}{m} \|\hat{\mathcal{A}}_x\|_F,\end{aligned}$$

where $m = \frac{1}{\max_{x \in \mathcal{U}}(\|G_x^{1/2}\|_2 \|G_x^{-1/2}\|_2)}$. On the other hand, let u_x satisfy

$$\frac{\|G_x^{1/2} \hat{\mathcal{A}}_x G_x^{-1/2} \hat{u}_x\|_2}{\|\hat{u}_x\|_2} = \|G_x^{1/2} \hat{\mathcal{A}}_x G_x^{-1/2}\|_2.$$

Using $\|\hat{\mathcal{A}}_x\|_F \le \sqrt{d}\|\hat{\mathcal{A}}_x\|_2$ ([GV96, (2.3.7)]), we have

$$\begin{aligned}\|\hat{\mathcal{A}}_x\|_F &\le \sqrt{d}\|\hat{\mathcal{A}}_x\|_2 \le \sqrt{d}\|G_x^{-1/2}\|_2\|G_x^{1/2}\hat{\mathcal{A}}_x G_x^{-1/2}\|_2\|G_x^{1/2}\|_2 \\ &= \sqrt{d}\|G_x^{-1/2}\|_2\|G_x^{1/2}\|_2\frac{\|G_x^{1/2}\hat{\mathcal{A}}_x G_x^{-1/2}\hat{u}_x\|_2}{\|\hat{u}_x\|_2} = \sqrt{d}\|G_x^{-1/2}\|_2\|G_x^{1/2}\|_2\frac{\|\mathcal{A}_x v_x\|}{\|v_x\|} \\ &\le \sqrt{d}\|G_x^{-1/2}\|_2\|G_x^{1/2}\|_2\|\mathcal{A}_x\| \le M\|\mathcal{A}_x\|,\end{aligned}$$

where $M = \sqrt{d}\max_{x\in\mathcal{U}}(\|G_x^{1/2}\|_2\|G_x^{-1/2}\|_2)$. □

We can now show R-linear convergence as stated in the following theorem that generalizes [BNY87, Lemma 4.2].

Theorem 6.2.1 (R-linear convergence)**.** *Suppose Assumptions 4.3.1, 4.3.2 and 6.2.1 hold.* $\phi_k \in [0, 1-\delta]$*. Then there is a constant* $0 \le a_{15} < 1$ *such that*

$$f(x_{k+1}) - f(x^*) \le a_{15}^k(f(x_1) - f(x^*)), \tag{6.2.8}$$

holds for all sufficiently large k*.*

Proof. By pre- and post- multiplying the update formula in Algorithm 3 by $H_{k+1}^{-1/2}$, we have

$$\begin{aligned}\bar{\mathcal{B}}_{k+1} &= H_{k+1}^{-1/2}\tilde{\mathcal{B}}_k H_{k+1}^{-1/2} - \frac{H_{k+1}^{-1/2}\tilde{\mathcal{B}}_k s_k(\tilde{\mathcal{B}}_k^* s_k)^\flat H_{k+1}^{-1/2}}{(\tilde{\mathcal{B}}_k^* s_k)^\flat s_k} \\ &\quad + \frac{H_{k+1}^{-1/2} y_k y_k^\flat H_{k+1}^{-1/2}}{y_k^\flat s_k} + \phi_k g(s_k, \tilde{\mathcal{B}}_k s_k) H_{k+1}^{-1/2} v_k v_k^\flat H_{k+1}^{-1/2} \\ &= H_{k+1}^{-1/2}\tilde{\mathcal{B}}_k H_{k+1}^{-1/2} - \frac{H_{k+1}^{-1/2}\tilde{\mathcal{B}}_k s_k(H_{k+1}^{-1/2}\tilde{\mathcal{B}}_k^* s_k)^\flat}{(\tilde{\mathcal{B}}_k^* s_k)^\flat s_k} \\ &\quad + \frac{H_{k+1}^{-1/2} y_k (H_{k+1}^{-1/2} y_k)^\flat}{y_k^\flat s_k} + \phi_k g(s_k, \tilde{\mathcal{B}}_k s_k) H_{k+1}^{-1/2} v_k v_k^\flat H_{k+1}^{-1/2} \\ &= H_{k+1}^{-1/2}\tilde{\mathcal{B}}_k H_{k+1}^{-1/2} - \frac{H_{k+1}^{-1/2}\tilde{\mathcal{B}}_k H_{k+1}^{-1/2}\bar{s}_k(H_{k+1}^{-1/2}\tilde{\mathcal{B}}_k^* H_{k+1}^{-1/2}\bar{s}_k)^\flat}{(H_{k+1}^{-1/2}\tilde{\mathcal{B}}_k^* H_{k+1}^{-1/2}\bar{s}_k)^\flat \bar{s}_k} \\ &\quad + \frac{\bar{y}_k\bar{y}_k^\flat}{\bar{y}_k^\flat\bar{s}_k} + \phi_k g(s_k, \tilde{\mathcal{B}}_k s_k) H_{k+1}^{-1/2} v_k (H_{k+1}^{-1/2} v_k)^\flat \\ &= \mathcal{C}_k - \frac{\mathcal{C}_k\bar{s}_k(\mathcal{C}_k^*\bar{s}_k)^\flat}{(\mathcal{C}_k^*\bar{s}_k)^\flat\bar{s}_k} + \frac{\bar{y}_k\bar{y}_k^\flat}{\bar{y}_k^\flat\bar{s}_k} + \phi_k g(\bar{s}_k, \mathcal{C}_k\bar{s}_k)\bar{v}_k\bar{v}_k^\flat,\end{aligned} \tag{6.2.9}$$

where

$$\bar{v}_k = \frac{H_{k+1}^{-1/2} y_k}{g(y_k, s_k)} - \frac{H_{k+1}^{-1/2}\tilde{\mathcal{B}}_k s_k}{g(s_k, \tilde{\mathcal{B}}_k s_k)} = \frac{\bar{y}_k}{g(\bar{y}_k, \bar{s}_k)} - \frac{\mathcal{C}_k\bar{s}_k}{g(\bar{s}_k, \mathcal{C}_k\bar{s}_k)}.$$

Considering the coordinate expression and the trace of (6.2.9), we have

$$\begin{aligned}\operatorname{trace}(\hat{\hat{\mathcal{B}}}_{k+1}) = \operatorname{trace}(\hat{\mathcal{C}}_k) + \frac{\|\bar{y}_k\|^2}{g(\bar{y}_k, \bar{s}_k)} + \phi_k \frac{\|\bar{y}_k\|^2}{g(\bar{y}_k, \bar{s}_k)} \frac{g(\bar{s}_k, \mathcal{C}_k \bar{s}_k)}{g(\bar{y}_k, \bar{s}_k)} \\ - (1 - \phi_k) \frac{\|\mathcal{C}_k \bar{s}_k\|^2}{g(\bar{s}_k, \mathcal{C}_k \bar{s}_k)} - 2\phi_k \frac{g(\bar{y}_k, \mathcal{C}_k \bar{s}_k)}{g(\bar{y}_k, \bar{s}_k)}.\end{aligned} \tag{6.2.10}$$

It follows that

$$\begin{aligned}
&\operatorname{trace}(\hat{\mathcal{C}}_k) - \operatorname{trace}(\hat{\hat{\mathcal{B}}}_k) \\
={}& \operatorname{trace}(\hat{H}_{k+1}^{-1/2} \hat{\hat{\mathcal{B}}}_k \hat{H}_{k+1}^{-1/2}) - \operatorname{trace}(\hat{H}_k^{-1/2} \hat{\mathcal{B}}_k \hat{H}_k^{-1/2}) \\
={}& \operatorname{trace}(\hat{\mathcal{T}}_{S_{\alpha_k \eta_k}} \hat{\mathcal{B}}_k \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^{-1} \hat{H}_{k+1}^{-1}) - \operatorname{trace}(\hat{\mathcal{B}}_k \hat{H}_k^{-1}) \\
={}& \operatorname{trace}(\hat{\mathcal{T}}_{S_{\alpha_k \eta_k}} \hat{\mathcal{B}}_k \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^{-1} \hat{\mathcal{T}}_{S_{\zeta_{k+1}}} \hat{H}_*^{-1} \hat{\mathcal{T}}_{S_{\zeta_{k+1}}}^{-1}) - \operatorname{trace}(\hat{\mathcal{B}}_k \hat{\mathcal{T}}_{S_{\zeta_k}} \hat{H}_*^{-1} \hat{\mathcal{T}}_{S_{\zeta_k}}^{-1}) \\
={}& \operatorname{trace}(\hat{\mathcal{T}}_{S_{\zeta_{k+1}}}^{-1} \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}} \hat{\mathcal{B}}_k \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^{-1} \hat{\mathcal{T}}_{S_{\zeta_{k+1}}} \hat{H}_*^{-1}) - \operatorname{trace}(\hat{\mathcal{T}}_{S_{\zeta_k}}^{-1} \hat{\mathcal{B}}_k \hat{\mathcal{T}}_{S_{\zeta_k}} \hat{H}_*^{-1}) \\
={}& \operatorname{trace}(\hat{\mathcal{T}}_{S_{\zeta_{k+1}}}^{-1} \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}} \hat{\mathcal{B}}_k \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^{-1} \hat{\mathcal{T}}_{S_{\zeta_{k+1}}} \hat{H}_*^{-1}) - \operatorname{trace}(\hat{\mathcal{T}}_{S_{\zeta_k}}^{-1} \hat{\mathcal{B}}_k \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^{-1} \hat{\mathcal{T}}_{S_{\zeta_{k+1}}} \hat{H}_*^{-1}) \\
&+ \operatorname{trace}(\hat{\mathcal{T}}_{S_{\zeta_k}}^{-1} \hat{\mathcal{B}}_k \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^{-1} \hat{\mathcal{T}}_{S_{\zeta_{k+1}}} \hat{H}_*^{-1}) - \operatorname{trace}(\hat{\mathcal{T}}_{S_{\zeta_k}}^{-1} \hat{\mathcal{B}}_k \hat{\mathcal{T}}_{S_{\zeta_k}} \hat{H}_*^{-1}) \\
\leq{}& \|\hat{\mathcal{T}}_{S_{\zeta_{k+1}}}^{-1} \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}} - \hat{\mathcal{T}}_{S_{\zeta_k}}^{-1}\|_F \|\hat{\mathcal{B}}_k \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^{-1} \hat{\mathcal{T}}_{S_{\zeta_{k+1}}} \hat{H}_*^{-1}\|_F + \|\hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^{-1} \hat{\mathcal{T}}_{S_{\zeta_{k+1}}} - \hat{\mathcal{T}}_{S_{\zeta_k}}\|_F \|\hat{H}_*^{-1} \hat{\mathcal{T}}_{S_{\zeta_k}}^{-1} \hat{\mathcal{B}}_k\|_F \\
\leq{}& b_0 (\|\hat{\mathcal{T}}_{S_{\zeta_{k+1}}}^{-1} \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}} - \hat{\mathcal{T}}_{S_{\zeta_k}}^{-1}\| \|\hat{\mathcal{B}}_k \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^{-1} \hat{\mathcal{T}}_{S_{\zeta_{k+1}}} \hat{H}_*^{-1}\| \\
&+ \|\hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^{-1} \hat{\mathcal{T}}_{S_{\zeta_{k+1}}} - \hat{\mathcal{T}}_{S_{\zeta_k}}\| \|\hat{H}_*^{-1} \hat{\mathcal{T}}_{S_{\zeta_k}}^{-1} \hat{\mathcal{B}}_k\|) \text{ (by Lemma 6.2.6)} \\
\leq{}& b_0 (\|\hat{\mathcal{T}}_{S_{\zeta_{k+1}}}^{-1} \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}} \hat{\mathcal{T}}_{S_{\zeta_k}} - I\| \|\hat{\mathcal{B}}_k\| \|\hat{H}_*^{-1}\| + \|\hat{\mathcal{T}}_{S_{\zeta_k}}^{-1} \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^{-1} \hat{\mathcal{T}}_{S_{\zeta_{k+1}}} - I\| \|\hat{H}_*^{-1}\| \|\hat{\mathcal{B}}_k\|) \\
\leq{}& b_1 \epsilon_k, \text{ (by Lemma 6.2.5)}
\end{aligned}$$

where b_0, b_1 are positive constants and $\| \cdot \|_F$ denotes the Frobenius norm. (6.2.10) becomes

$$\begin{aligned}\operatorname{trace}(\hat{\hat{\mathcal{B}}}_{k+1}) \leq \operatorname{trace}(\hat{\hat{\mathcal{B}}}_k) + \frac{\|\bar{y}_k\|^2}{g(\bar{y}_k, \bar{s}_k)} + \phi_k \frac{\|\bar{y}_k\|^2}{g(\bar{y}_k, \bar{s}_k)} \frac{g(\bar{s}_k, \mathcal{C}_k \bar{s}_k)}{g(\bar{y}_k, \bar{s}_k)} \\ - (1 - \phi_k) \frac{\|\mathcal{C}_k \bar{s}_k\|^2}{g(\bar{s}_k, \mathcal{C}_k \bar{s}_k)} - 2\phi_k \frac{g(\bar{y}_k, \mathcal{C}_k \bar{s}_k)}{g(\bar{y}_k, \bar{s}_k)} + b_1 \epsilon_k.\end{aligned} \tag{6.2.11}$$

Take $\epsilon \in (0, 1]$ and, without loss of generality (since the claim to prove is for all sufficiently large k), assume that, for all k, x_k belongs to $N(x^*)$ defined in Lemma 6.2.4. Then, from the above inequality and exploiting the fact that $\epsilon \le 1$, we have

$$\begin{aligned}
\operatorname{trace}(\hat{\tilde{\mathcal{B}}}_{k+1}) &\le \operatorname{trace}(\hat{\tilde{\mathcal{B}}}_k) + 2 + \frac{a_{14}\phi\epsilon\alpha_k}{\cos\bar{\theta}_k} - (1-\phi_k)\frac{\|\mathcal{C}_k\bar{s}_k\|^2}{g(\bar{s}_k, \mathcal{C}_k\bar{s}_k)} + b_1\epsilon_k \\
&\quad \text{(by (6.2.7) with } \epsilon \le 1 \text{ and Lemma 6.2.4)} \\
&= \operatorname{trace}(\hat{\tilde{\mathcal{B}}}_k) + 2 + \frac{a_{14}\phi\epsilon\alpha_k}{\cos\bar{\theta}_k} - (1-\phi_k)\frac{\|\mathcal{C}_k\bar{s}_k\|^2}{\|\bar{s}_k\|\|\mathcal{C}_k\bar{s}_k\|\cos\bar{\theta}_k} + b_1\epsilon_k \\
&= \operatorname{trace}(\hat{\tilde{\mathcal{B}}}_k) + 2 + \frac{a_{14}\phi\epsilon\alpha_k}{\cos\bar{\theta}_k} - (1-\phi_k)\frac{\|H_{k+1}^{-1/2}\tilde{\mathcal{B}}_k s_k\|}{\|H_{k+1}^{1/2}s_k\|\cos\bar{\theta}_k} + b_1\epsilon_k \\
&= \operatorname{trace}(\hat{\tilde{\mathcal{B}}}_k) + 2 + \frac{a_{14}\phi\epsilon\alpha_k}{\cos\bar{\theta}_k} - (1-\phi_k)\frac{\|H_{k+1}^{1/2}\|^2}{\|H_*^{1/2}\|^2}\frac{\|H_{k+1}^{-1/2}\tilde{\mathcal{B}}_k s_k\|}{\|H_{k+1}^{1/2}s_k\|\cos\bar{\theta}_k} + b_1\epsilon_k \\
&\quad \text{(since } \|H_{k+1}\| = \|H_*\|\text{)} \\
&\le \operatorname{trace}(\hat{\tilde{\mathcal{B}}}_k) + 2 + \frac{a_{14}\phi\epsilon\alpha_k}{\cos\bar{\theta}_k} - (1-\phi_k)\frac{1}{\|H_*^{1/2}\|^2}\frac{\|\tilde{\mathcal{B}}_k s_k\|}{\|s_k\|\cos\bar{\theta}_k} + b_1\epsilon_k \qquad (6.2.12)
\end{aligned}$$

Since $\|\cdot\|$ is an induce norm, inequalities $\|H_{k+1}^{1/2}\|\|H_{k+1}^{-1/2}\tilde{\mathcal{B}}_k s_k\| \ge \|\tilde{\mathcal{B}}_k s_k\|$ and $\|H_{k+1}^{1/2}s_k\| \le \|H_{k+1}^{1/2}\|\|s_k\|$ hold. Therefore, (6.2.12) holds. It follows that

$$\begin{aligned}
&\operatorname{trace}(\hat{\tilde{\mathcal{B}}}_k) + 2 + \frac{a_{14}\phi\epsilon\alpha_k}{\cos\bar{\theta}_k} - (1-\phi_k)\frac{1}{\|H_*^{1/2}\|^2}\frac{\|\tilde{\mathcal{B}}_k s_k\|}{\|s_k\|\cos\bar{\theta}_k} + b_1\epsilon_k \\
\le{}& \operatorname{trace}(\hat{\tilde{\mathcal{B}}}_k) + 2 + \frac{a_{14}\phi\epsilon\alpha_k}{\cos\bar{\theta}_k} - (1-\phi_k)b_2\frac{\alpha_k}{\cos^2\theta_k} + b_1\epsilon_k \text{ (by Lemma 4.3.5)} \\
\le{}& \operatorname{trace}(\hat{\tilde{\mathcal{B}}}_k) + 2 + b_1 + \frac{a_{14}\phi\epsilon\alpha_k}{\cos\bar{\theta}_k} - (1-\phi_k)b_2\frac{\alpha_k}{\cos^2\theta_k} \text{ (let } \epsilon_k \le 1\text{)}
\end{aligned}$$

where b_2 is a positive constant. We have

$$0 < \operatorname{trace}(\hat{\tilde{\mathcal{B}}}_{k+1}) \le \operatorname{trace}(\hat{\tilde{\mathcal{B}}}_k) + 2 + b_1 + \frac{\alpha_k}{\cos^2\bar{\theta}_k}(a_{14}\phi_k\epsilon\cos\bar{\theta}_k - (1-\phi_k)b_2). \qquad (6.2.13)$$

We choose ϵ small enough such that $a_{14}(1-\delta)\epsilon\cos\bar{\theta}_k - \delta b_2$ is less than some negative number, denoted $-b_3$, for $k > k_0$ by noting that $a_{14}\phi_k\epsilon\cos\bar{\theta}_k - (1-\phi_k)b_2 \le a_{14}(1-\delta)\epsilon\cos\bar{\theta}_k - \delta b_2 < -b_3$. From (6.2.13), there exists a constant b_4 such that

$$0 < \operatorname{trace}(\hat{\tilde{\mathcal{B}}}_{k+1}) \le \operatorname{trace}(\hat{\tilde{\mathcal{B}}}_1) + (2+b_1)k + b_4 - b_3\sum_{i=1}^{k}\frac{\alpha_i}{\cos^2\bar{\theta}_i}.$$

Therefore, we have

$$\sum_{i=1}^{k}\frac{\alpha_i}{\cos^2\bar{\theta}_i} \le k\ b_5, \qquad (6.2.14)$$

for some positive constant b_5. Using the relationship between the geometric and arithmetic means and (6.2.14), we obtain

$$\prod_{i=1}^{k} \frac{\alpha_i}{\cos^2 \bar{\theta}_i} \leq b_5^k.$$

From Lemma 4.3.12, we get

$$\prod_{i=1}^{k} \cos^2 \bar{\theta}_i \geq (\frac{a_{13}}{b_5})^k.$$

It follows that

$$\begin{aligned}\prod_{i=1}^{k} \cos^2 \theta_i &= \prod_{i=1}^{k} \cos^2 \bar{\theta}_i \prod_{i=1}^{k} (\frac{\cos \theta_i}{\cos \bar{\theta}_i})^2 = \prod_{i=1}^{k} \cos^2 \bar{\theta}_i (\prod_{i=1}^{k} \frac{\|H_{k+1}^{1/2} s_k\| \|H_{k+1}^{-1/2} \tilde{\mathcal{B}}_k s_k\|}{\|s_k\| \|\tilde{\mathcal{B}}_k s_k\|})^2 \\ &= \prod_{i=1}^{k} \cos^2 \bar{\theta}_i (\prod_{i=1}^{k} \frac{\|H_{k+1}^{1/2} s_k\| \|H_{k+1}^{-1/2} \tilde{\mathcal{B}}_k s_k\|}{\|H_{k+1}^{-1/2} H_{k+1}^{1/2} s_k\| \|H_{k+1}^{1/2} H_{k+1}^{-1/2} \tilde{\mathcal{B}}_k s_k\|})^2 \geq (\frac{a_{13}}{b_5})^k (\frac{1}{\|H_*^{1/2}\| \|H_*^{-1/2}\|})^k = b_6^k,\end{aligned}$$

where $b_6 = a_{13}/(b_5 \|H_*^{1/2}\| \|H_*^{-1/2}\|)$. By Lemma 4.3.8, we have

$$f(x_{k+1}) - f(x^*) \leq \prod_{i=1}^{k} (1 - a_5 \cos^2 \theta_i)(f(x_1) - f(x^*)).$$

Using the relationship between the geometric and arithmetic means twice, we obtain

$$\begin{aligned} f(x_{k+1}) - f(x^*) &\leq (\frac{1}{k} \sum_{i=1}^{k} (1 - a_5 \cos^2 \theta_i))^k (f(x_1) - f(x^*)) \\ &\leq (1 - a_5 (\prod_{i=1}^{k} \cos^2 \theta_i)^{\frac{1}{k}})^k (f(x_1) - f(x^*)) = (1 - a_5 b_6)^k (f(x_1) - f(x^*)),\end{aligned}$$

which is

$$f(x_{k+1}) - f(x^*) \leq a_{15}^k (f(x_1) - f(x^*)),$$

where $a_{15} = 1 - a_5 b_6$. □

6.2.2 Superlinear Convergence Analysis

Assumption 6.2.2 generalizes the Euclidean property of twice Hölder continuously differentiability of f at x^* to a Riemannian manifold and it is weaker than Assumption 3.3.3. If the x in Assumption 3.3.3 is restricted to be x^*, then Assumption 3.3.3 is Assumption 6.2.2 with $p = 1$.

Assumption 6.2.2. *There exists a constant c_4 and p such that for all $y \in \mathcal{S}$,*

$$\| \operatorname{Hess} f(y) - \mathcal{T}_{S_\eta} \operatorname{Hess} f(x^*) \mathcal{T}_{S_\eta}^{-1} \| \leq c_4 \|\eta\|^p,$$

where $\eta = R_{x^*}^{-1} y$.

From the R-linear convergence of Theorem 6.2.1, we obtain the following lemma that is used in the superlinear convergence analysis.

Lemma 6.2.7. *Suppose Assumptions 4.3.1, 4.3.2 and 6.2.1 hold. $\phi_k \in [0, 1-\delta]$. The sequence $\{x_k\}$ generated by Algorithm 3 converges to a minimizer x^* of f. Then inequality*

$$\sum_{k=1}^{\infty} (\mathrm{dist}(x_k, x^*))^{\min(1,p)} < \infty$$

holds.

Proof. Define $\tilde{m}_k(t) = f(R_{x^*}(t\zeta_k))$, where $\zeta_k = R_{x^*}^{-1}(x_k)/\|R_{x^*}^{-1}(x_k)\|$. Let $z_k = \|R_{x^*}^{-1}(x_k)\|$. By Taylor's theorem, we have

$$\begin{aligned} f(x_k) - f(x^*) = \tilde{m}_k(z_k) - \tilde{m}_k(0) = \frac{d\tilde{m}_k(0)}{dt} z_k + \frac{d^2\tilde{m}_k(p)}{dt^2} z_k^2 \\ \geq a_0 z_k^2, \text{ (by Assumption 4.3.2 and } x^* \text{ is a minimizer)} \\ \geq b_0 \,\mathrm{dist}(x_k, x^*)^2 \text{ (by Lemma 3.3.3)} \end{aligned}$$

where p is some number between 0 and z_k and b_0 is a positive constant. According to (6.2.8), we have

$$(\mathrm{dist}(x_k, x^*))^{\min(1,p)} \leq a_{15}^{\min((k-1)/2, p(k-1)/2)} \left(\frac{f(x_1) - f(x^*)}{b_0} \right)^{\min(p,1)/2}.$$

Since $a_{15} \in [0, 1)$, we know $(\mathrm{dist}(x_k, x^*))^{\min(1,p)}$ is less than a geometric sequence whose common ratio is less than 1 and, therefore, the summation is finite,

$$\sum_{k=1}^{\infty} (\mathrm{dist}(x_k, x^*))^{\min(1,p)} < \infty.$$

□

Lemma 6.2.8. *Suppose Assumption 4.3.1 holds. Let $\bar{x} \in \mathcal{M}$. Then there is a neighborhood $\mathcal{U}$ of $\bar{x}$ and a_{16} such that for all $x, y \in \mathcal{U}$,*

$$\|H(x) - \mathcal{T}_{S_\eta}^{-1} H(y) \mathcal{T}_{S_\eta}\| \leq a_{16} \max(\mathrm{dist}(x, \bar{x}), \mathrm{dist}(y, \bar{x})),$$

where $H(x) = \mathcal{T}_{S_\xi} \operatorname{Hess} f(\bar{x}) (\mathcal{T}_{S_\xi})^{-1}$, $\xi = R_{\bar{x}}^{-1} x$, $H(y) = \mathcal{T}_{S_\zeta} \operatorname{Hess} f(\bar{x}) (\mathcal{T}_{S_\zeta})^{-1}$, $\zeta = R_{\bar{x}}^{-1} y$ and $\eta = R_x^{-1} y$.

Proof. We have

$$\begin{aligned}\|H(x) - \mathcal{T}_{S_\eta}^{-1} H(y)\mathcal{T}_{S_\eta}\| &= \|\mathcal{T}_{S_\xi} \operatorname{Hess} f(\bar{x})\mathcal{T}_{S_\xi}^{-1} - \mathcal{T}_{S_\eta}^{-1}\mathcal{T}_{S_\zeta} \operatorname{Hess} f(\bar{x})\mathcal{T}_{S_\zeta}^{-1}\mathcal{T}_{S_\eta}\| \\ &= \|\operatorname{Hess} f(\bar{x}) - \mathcal{T}_{S_\xi}^{-1}\mathcal{T}_{S_\eta}^{-1}\mathcal{T}_{S_\zeta} \operatorname{Hess} f(\bar{x})\mathcal{T}_{S_\zeta}^{-1}\mathcal{T}_{S_\eta}\mathcal{T}_{S_\xi}\| \\ &\leq \|(\operatorname{id} - \mathcal{T}_{S_\xi}^{-1}\mathcal{T}_{S_\eta}^{-1}\mathcal{T}_{S_\zeta}) \operatorname{Hess} f(\bar{x})\| \\ &\quad + \|\mathcal{T}_{S_\xi}^{-1}\mathcal{T}_{S_\eta}^{-1}\mathcal{T}_{S_\zeta} \operatorname{Hess} f(\bar{x})(\operatorname{id} - \mathcal{T}_{S_\zeta}^{-1}\mathcal{T}_{S_\eta}\mathcal{T}_{S_\xi})\| \\ &\leq \|\operatorname{Hess} f(\bar{x})\|(\|\operatorname{id} - \mathcal{T}_{S_\xi}^{-1}\mathcal{T}_{S_\eta}^{-1}\mathcal{T}_{S_\zeta}\| + \|\operatorname{id} - \mathcal{T}_{S_\zeta}^{-1}\mathcal{T}_{S_\eta}\mathcal{T}_{S_\xi}\|) \\ &\leq b_0 \max(\operatorname{dist}(x, \bar{x}), \operatorname{dist}(y, \bar{x})) \text{ (by Lemma 6.2.5)},\end{aligned}$$

where b_0 is a positive constant. □

Lemma 6.2.9. *Suppose Assumptions 4.3.1, 4.3.2, 6.2.1 and 6.2.2 hold. Then*

$$\|\bar{y}_k - \bar{s}_k\| \leq \tilde{a}_{17}\epsilon_k^{\min(1,p)}\|\bar{s}_k\| \tag{6.2.15}$$

$$g(\bar{y}_k, \bar{s}_k) \geq (1 - \tilde{a}_{18}\epsilon_k^{\min(1,p)})\|\bar{s}_k\|^2, \tag{6.2.16}$$

where $\tilde{a}_{17}$ *and* $\tilde{a}_{18}$ *are positive constants.*

Proof. We follow the proof of Lemma 6.2.2 modified by replacing (6.2.1) $\leq o_1(\epsilon_k)\|s_k\|$ by (6.2.1) $\leq b_4\epsilon_k^p\|s_k\|$ since Assumption 6.2.2 holds. Therefore, we have

$$\|\bar{y}_k - \bar{s}_k\| \leq c\epsilon_k^{\min(1,p)}\|\bar{s}_k\|, \tag{6.2.17}$$

where b_4 and c are some constants. It follows that

$$\|\bar{y}_k\| - \|\bar{s}_k\| \leq c\epsilon_k^{\min(1,p)}\|\bar{s}_k\| \text{ and } \|\bar{s}_k\| - \|\bar{y}_k\| \leq c\epsilon_k^{\min(1,p)}\|\bar{s}_k\|,$$

which yields

$$(1 - c\epsilon_k^{\min(1,p)})\|\bar{s}_k\| \leq \|\bar{y}_k\| \leq (1 + c\epsilon_k^{\min(1,p)})\|\bar{s}_k\|. \tag{6.2.18}$$

By squaring (6.2.17) and using (6.2.18), we have

$$(1 - c\epsilon_k^{\min(1,p)})^2\|\bar{s}_k\|^2 - 2g(\bar{y}_k, \bar{s}_k) + \|\bar{s}_k\|^2 \leq \|\bar{y}_k\|^2 - 2g(\bar{y}_k, \bar{s}_k) + \|\bar{s}_k\|^2 \leq (c\epsilon_k^{\min(1,p)})^2\|\bar{s}_k\|^2,$$

and therefore

$$g(\bar{y}_k, \bar{s}_k) \geq (1 - c\epsilon_k^{\min(1,p)})\|\bar{s}_k\|^2 \tag{6.2.19}$$

completing the proof. □

Lemma 6.2.10 generalizes [GT82, (45)]. Lemmas 6.2.10 and 6.2.11 show that all Hessian approximations, $\mathcal{B}_k$, given by the RBroyden update (4.2.3) are bounded. Corollaries 6.2.1 and 6.2.2 show the key result that the condition numbers of all of the $\mathcal{B}_k$ are also bounded.

Lemma 6.2.10. *Suppose Assumptions 4.3.1, 4.3.2, 6.2.1 and 6.2.2 hold.* $\phi_k \in [0, 1]$. *Then there exist constants* a_{17} *and* a_{18} *such that*

$$\|\bar{\mathcal{B}}'_{k+1} - \bar{\mathcal{B}}_{k+1}\| \leq (a_{17}\|\mathcal{C}_k\| + a_{18})\epsilon_k^{\min(1,p)},$$

where

$$\bar{\mathcal{B}}'_{k+1} = \mathcal{C}_k - (1-\phi_k)\frac{\mathcal{C}_k\bar{s}_k(\mathcal{C}_k^*\bar{s}_k)^\flat}{(\mathcal{C}_k^*\bar{s}_k)^\flat\bar{s}_k} + (1+\phi_k\frac{(\mathcal{C}_k\bar{s}_k)^\flat\bar{s}_k}{\bar{s}_k^\flat\bar{s}_k})\frac{\bar{s}_k\bar{s}_k^\flat}{\bar{s}_k^\flat\bar{s}_k} - \phi_k(\frac{\bar{s}_k(\mathcal{C}_k\bar{s}_k)^\flat}{\bar{s}_k^\flat\bar{s}_k} + \frac{\mathcal{C}_k\bar{s}_k\bar{s}_k^\flat}{\bar{s}_k^\flat\bar{s}_k}).$$

Proof. From (6.2.9), we have

$$\bar{\mathcal{B}}_{k+1} = \mathcal{C}_k - (1-\phi_k)\frac{\mathcal{C}_k\bar{s}_k(\mathcal{C}_k^*\bar{s}_k)^\flat}{(\mathcal{C}_k^*\bar{s}_k)^\flat\bar{s}_k} + (1+\phi_k\frac{(\mathcal{C}_k\bar{s}_k)^\flat\bar{s}_k}{\bar{y}_k^\flat\bar{s}_k})\frac{\bar{y}_k\bar{y}_k^\flat}{\bar{y}_k^\flat\bar{s}_k} - \phi_k(\frac{\bar{y}_k(\mathcal{C}_k\bar{s}_k)^\flat}{\bar{y}_k^\flat\bar{s}_k} + \frac{\mathcal{C}_k\bar{s}_k\bar{y}_k^\flat}{\bar{y}_k^\flat\bar{s}_k}).$$

$$\begin{aligned}
\|\bar{\mathcal{B}}'_{k+1} - \bar{\mathcal{B}}_{k+1}\| &= \|(1+\phi_k\frac{(\mathcal{C}_k\bar{s}_k)^\flat\bar{s}_k}{\bar{s}_k^\flat\bar{s}_k})\frac{\bar{s}_k\bar{s}_k^\flat}{\bar{s}_k^\flat\bar{s}_k} - \phi_k(\frac{\bar{s}_k(\mathcal{C}_k\bar{s}_k)^\flat}{\bar{s}_k^\flat\bar{s}_k} + \frac{\mathcal{C}_k\bar{s}_k\bar{s}_k^\flat}{\bar{s}_k^\flat\bar{s}_k}) \\
&\quad - ((1+\phi_k\frac{(\mathcal{C}_k\bar{s}_k)^\flat\bar{s}_k}{\bar{y}_k^\flat\bar{s}_k})\frac{\bar{y}_k\bar{y}_k^\flat}{\bar{y}_k^\flat\bar{s}_k} - \phi_k(\frac{\bar{y}_k(\mathcal{C}_k\bar{s}_k)^\flat}{\bar{y}_k^\flat\bar{s}_k} + \frac{\mathcal{C}_k\bar{s}_k\bar{y}_k^\flat}{\bar{y}_k^\flat\bar{s}_k}))\| \\
&\leq \|\frac{\bar{s}_k\bar{s}_k^\flat}{\bar{s}_k^\flat\bar{s}_k} - \frac{\bar{y}_k\bar{y}_k^\flat}{\bar{y}_k^\flat\bar{s}_k}\| && (6.2.20) \\
&\quad + \|\phi_k\frac{(\mathcal{C}_k\bar{s}_k)^\flat\bar{s}_k}{\bar{s}^\flat\bar{s}_k}\frac{\bar{s}_k\bar{s}_k^\flat}{\bar{s}^\flat\bar{s}_k} - \phi_k\frac{(\mathcal{C}_k\bar{s}_k)^\flat\bar{s}_k}{\bar{y}_k^\flat\bar{s}_k}\frac{\bar{y}_k\bar{y}_k^\flat}{\bar{y}_k^\flat\bar{s}_k}\| && (6.2.21) \\
&\quad + \phi_k\|\frac{\bar{y}_k(\mathcal{C}_k\bar{s}_k)^\flat}{\bar{y}_k^\flat\bar{s}_k} - \frac{\bar{s}_k(\mathcal{C}_k\bar{s}_k)^\flat}{\bar{s}_k^\flat\bar{s}_k}\| && (6.2.22) \\
&\quad + \phi_k\|\frac{\mathcal{C}_k\bar{s}_k\bar{y}_k^\flat}{\bar{y}_k^\flat\bar{s}_k} - \frac{\mathcal{C}_k\bar{s}_k\bar{s}_k^\flat}{\bar{s}_k^\flat\bar{s}_k}\|. && (6.2.23)
\end{aligned}$$

Since $\|\cdot\|$ is an induced norm, we have

$$\|uv^\flat\| = \|u\|\|v\|. \quad (6.2.24)$$

It follows that

$$\begin{aligned}
(6.2.20) &\leq \|\frac{\bar{s}_k\bar{s}_k^\flat}{\bar{s}_k^\flat\bar{s}_k} - \frac{\bar{y}_k\bar{y}_k^\flat}{\bar{s}_k^\flat\bar{s}_k}\| + \|\frac{\bar{y}_k\bar{y}_k^\flat}{\bar{s}_k^\flat\bar{s}_k} - \frac{\bar{y}_k\bar{y}_k^\flat}{\bar{y}_k^\flat\bar{s}_k}\| \\
&\leq \|\frac{\bar{s}_k\bar{s}_k^\flat}{\bar{s}_k^\flat\bar{s}_k} - \frac{\bar{y}_k\bar{s}_k^\flat}{\bar{s}_k^\flat\bar{s}_k}\| + \|\frac{\bar{y}_k\bar{s}_k^\flat}{\bar{s}_k^\flat\bar{s}_k} - \frac{\bar{y}_k\bar{y}_k^\flat}{\bar{s}_k^\flat\bar{s}_k}\| + |\frac{\bar{y}_k^\flat\bar{y}_k}{\bar{s}_k^\flat\bar{s}_k} - \frac{\bar{y}_k^\flat\bar{y}_k}{\bar{y}_k^\flat\bar{s}_k}|\|\frac{\bar{y}_k\bar{y}_k^\flat}{\bar{y}_k^\flat\bar{y}_k}\| \\
&= \|\frac{(\bar{s}_k - \bar{y}_k)\bar{s}_k^\flat}{\bar{s}_k^\flat\bar{s}_k}\| + \|\frac{\bar{y}_k(\bar{s}_k^\flat - \bar{y}_k^\flat)}{\bar{s}_k^\flat\bar{s}_k}\| + |\frac{\bar{y}_k^\flat\bar{y}_k}{\bar{s}_k^\flat\bar{s}_k} - \frac{\bar{y}_k^\flat\bar{y}_k}{\bar{y}_k^\flat\bar{s}_k}|\|\frac{\bar{y}_k\bar{y}_k^\flat}{\bar{y}_k^\flat\bar{y}_k}\| \\
&= \frac{\|\bar{s}_k - \bar{y}_k\|\|\bar{s}_k\|}{\|\bar{s}_k\|^2} + \frac{\|\bar{y}_k\|\|\bar{s}_k - \bar{y}_k\|}{\|\bar{s}_k\|^2} + |\frac{\bar{y}_k^\flat\bar{y}_k}{\bar{s}_k^\flat\bar{s}_k} - \frac{\bar{y}_k^\flat\bar{y}_k}{\bar{y}_k^\flat\bar{s}_k}| \text{ (by (6.2.24))} \\
&\leq b_1\epsilon_k^{\min(1,p)} \text{ (by (6.2.15) and (6.2.16))}
\end{aligned}$$

where b_1 is some constant,

$$\begin{aligned}
(6.2.21) &\leq \|\phi_k\frac{(\mathcal{C}_k\bar{s}_k)^\flat\bar{s}_k}{\bar{s}_k^\flat\bar{s}_k}\frac{\bar{s}_k\bar{s}_k^\flat}{\bar{s}_k^\flat\bar{s}_k} - \phi_k\frac{(\mathcal{C}_k\bar{s}_k)^\flat\bar{s}_k}{\bar{s}_k^\flat\bar{s}_k}\frac{\bar{y}_k\bar{y}_k^\flat}{\bar{s}_k^\flat\bar{s}_k}\| + \|\phi_k\frac{(\mathcal{C}_k\bar{s}_k)^\flat\bar{s}_k}{\bar{s}_k^\flat\bar{s}_k}\frac{\bar{y}_k\bar{y}_k^\flat}{\bar{s}_k^\flat\bar{s}_k} - \phi_k\frac{(\mathcal{C}_k\bar{s}_k)^\flat\bar{s}_k}{\bar{y}^\flat\bar{s}_k}\frac{\bar{y}_k\bar{y}_k^\flat}{\bar{y}_k^\flat\bar{s}_k}\| \\
&\leq \phi_k\frac{(\mathcal{C}_k\bar{s}_k)^\flat\bar{s}_k}{\bar{s}_k^\flat\bar{s}_k}(\|\frac{\bar{s}_k\bar{s}_k^\flat}{\bar{s}_k^\flat\bar{s}_k} - \frac{\bar{y}_k\bar{s}_k^\flat}{\bar{s}_k^\flat\bar{s}_k}\| + \|\frac{\bar{y}_k\bar{s}_k^\flat}{\bar{s}_k^\flat\bar{s}_k} - \frac{\bar{y}_k\bar{y}_k^\flat}{\bar{s}_k^\flat\bar{s}_k}\|) + \phi_k\frac{(\mathcal{C}_k\bar{s}_k)^\flat\bar{s}_k}{\bar{s}_k^\flat\bar{s}_k}|\frac{\bar{y}_k^\flat\bar{y}_k}{\bar{s}_k^\flat\bar{s}_k} - \frac{\bar{y}_k^\flat\bar{y}_k\bar{s}_k^\flat\bar{s}_k}{(\bar{y}^\flat\bar{s}_k)^2}|\|\frac{\bar{y}_k\bar{y}_k^\flat}{\bar{y}_k^\flat\bar{y}_k}\| \\
&= \phi_k\frac{(\mathcal{C}_k\bar{s}_k)^\flat\bar{s}_k}{\bar{s}_k^\flat\bar{s}_k}(\|\frac{(\bar{s}_k - \bar{y}_k)\bar{s}_k^\flat}{\bar{s}_k^\flat\bar{s}_k}\| + \|\frac{\bar{y}_k(\bar{s}_k^\flat - \bar{y}_k^\flat)}{\bar{s}_k^\flat\bar{s}_k}\|) + \phi_k\frac{(\mathcal{C}_k\bar{s}_k)^\flat\bar{s}_k}{\bar{s}_k^\flat\bar{s}_k}|\frac{\bar{y}_k^\flat\bar{y}_k}{\bar{s}_k^\flat\bar{s}_k} - \frac{\bar{y}_k^\flat\bar{y}_k\bar{s}_k^\flat\bar{s}_k}{(\bar{y}^\flat\bar{s}_k)^2}| \\
&= \phi_k\frac{(\mathcal{C}_k\bar{s}_k)^\flat\bar{s}_k}{\bar{s}_k^\flat\bar{s}_k}(\frac{\|\bar{s}_k - \bar{y}_k\|\|\bar{s}_k\|}{\|\bar{s}_k\|^2} + \frac{\|\bar{y}_k\|\|\bar{s}_k - \bar{y}_k\|}{\|\bar{s}_k\|^2}) + \phi_k\frac{(\mathcal{C}_k\bar{s}_k)^\flat\bar{s}_k}{\bar{s}_k^\flat\bar{s}_k}|\frac{\bar{y}_k^\flat\bar{y}_k}{\bar{s}_k^\flat\bar{s}_k} - \frac{\bar{y}_k^\flat\bar{y}_k\bar{s}_k^\flat\bar{s}_k}{(\bar{y}^\flat\bar{s}_k)^2}| \\
&\quad \text{(by (6.2.24))} \\
&\leq (b_2\|\mathcal{C}_k\| + b_3)\epsilon_k^{\min(1,p)}, \text{ (by (6.2.15) and (6.2.16))}
\end{aligned}$$

where b_2, b_3 are some constants,

$$\begin{aligned}
(6.2.22) &\leq \phi_k\|\frac{\bar{y}_k(\mathcal{C}_k\bar{s}_k)^\flat}{\bar{y}_k^\flat\bar{s}_k} - \frac{\bar{s}_k(\mathcal{C}_k\bar{s}_k)^\flat}{\bar{y}_k^\flat\bar{s}_k}\| + \phi_k\|\frac{\bar{s}_k(\mathcal{C}_k\bar{s}_k)^\flat}{\bar{y}_k^\flat\bar{s}_k} - \frac{\bar{s}_k(\mathcal{C}_k\bar{s}_k)^\flat}{\bar{s}_k^\flat\bar{s}_k}\| \\
&\leq \phi_k\frac{\bar{s}_k^\flat\bar{s}_k}{\bar{y}_k^\flat\bar{s}_k}\|\frac{(\bar{y}_k - \bar{s}_k)(\mathcal{C}_k\bar{s}_k)^\flat}{\bar{s}_k^\flat\bar{s}_k}\| + \phi_k|\frac{\bar{s}_k^\flat\bar{s}_k}{\bar{y}_k^\flat\bar{s}_k} - 1|\|\frac{\bar{s}_k(\mathcal{C}_k\bar{s}_k)^\flat}{\bar{s}_k^\flat\bar{s}_k}\| \\
&\leq \phi_k(1 + b_4\epsilon_k)\frac{\|\bar{y}_k - \bar{s}_k\|\|\mathcal{C}_k\|\|\bar{s}_k\|}{\|\bar{s}_k\|^2} + \phi_k b_5\epsilon_k\frac{\|\bar{s}_k\|\|\mathcal{C}_k\|\|\bar{s}_k\|}{\|\bar{s}_k\|^2} \\
&\quad \text{(by (6.2.18), (6.2.16), (6.2.24))} \\
&\leq (b_6\|\mathcal{C}_k\| + b_7)\epsilon_k^{\min(1,p)}, \text{ (by (6.2.15) and (6.2.16))}
\end{aligned}$$

where b_4, b_5, b_6 and b_7 are some constants, and

$$\begin{aligned}
(6.2.23) &\leq \phi_k \| \frac{\mathcal{C}_k \bar{s}_k \bar{y}_k^\flat}{\bar{y}_k^\flat \bar{s}_k} - \frac{\mathcal{C}_k \bar{s}_k \bar{s}_k^\flat}{\bar{y}_k^\flat \bar{s}_k} \| + \phi_k \| \frac{\mathcal{C}_k \bar{s}_k \bar{s}_k^\flat}{\bar{y}_k^\flat \bar{s}_k} - \frac{\mathcal{C}_k \bar{s}_k \bar{s}_k^\flat}{\bar{s}_k^\flat \bar{s}_k} \| \\
&\leq \phi_k \frac{\bar{s}_k^\flat \bar{s}_k}{\bar{y}_k^\flat \bar{s}_k} \| \frac{\mathcal{C}_k \bar{s}_k (\bar{y}_k^\flat - \bar{s}_k^\flat)}{\bar{s}_k^\flat \bar{s}_k} \| + \phi_k | \frac{\bar{s}_k^\flat \bar{s}_k}{\bar{y}_k^\flat \bar{s}_k} - 1 | \| \frac{\mathcal{C}_k \bar{s}_k \bar{s}_k^\flat}{\bar{s}_k^\flat \bar{s}_k} \| \\
&\leq \phi_k (1 + b_8 \epsilon_k) \frac{\|\mathcal{C}_k\| \|\bar{s}_k\| \|\bar{y}_k - \bar{s}_k\|}{\|\bar{s}_k\|^2} + \phi_k b_9 \epsilon_k \frac{\|\mathcal{C}_k\| \|\bar{s}_k\| \|\bar{s}_k\|}{\|\bar{s}_k\|^2} \\
&\quad \text{(by (6.2.18), (6.2.16), (6.2.24))} \\
&\leq (b_{10} \|\mathcal{C}_k\| + b_{11}) \epsilon_k^{\min(1,p)}, \text{ (by (6.2.15) and (6.2.16))}
\end{aligned}$$

where b_8, b_9, b_{10} and b_{11} are some constants. Combining the above inequalities, we have

$$\|\bar{\mathcal{B}}'_{k+1} - \bar{\mathcal{B}}_{k+1}\| \leq ((b_2 + b_6 + b_{10}) \|\mathcal{C}_k\| + b_1 + b_3 + b_7 + b_{11}) \epsilon_k^{\min(1,p)}.$$

□

Lemma 6.2.11. *Suppose Assumptions 4.3.1, 4.3.2, 6.2.1 and 6.2.2 hold.* $\phi_k \in [0, 1-\delta]$. *Then the sequence* $\{\bar{\mathcal{B}}_k\} = H_k^{-1/2} \mathcal{B}_k H_k^{-1/2}$ *from the Algorithm 3 is bounded, i.e., there exists a constant* a_{19} *such that*

$$\|\bar{\mathcal{B}}_k\| \leq a_{19}, \tag{6.2.25}$$

for all k.

Proof. For each tangent space $\mathrm{T}_x \mathcal{M}$ we consider a basis that is orthonormal with respect to the Riemannian metric g, with $x = x^*, x_0, x_1, \ldots$, and we let a hat denote expressions in these bases. Consequently, for all k, the matrix expression of the inner product g_{x_k} is the identity; in other words, $g(u, v) = \hat{u}^T \hat{v}$ for all $u, v \in \mathrm{T}_k \mathcal{M}$. Using the notation of Lemma 6.2.10 and by the same calculation as (47) in the paper of Griewank and Toint (1982), we get

$$\begin{aligned}
\|\hat{\bar{\mathcal{B}}}'_{k+1} - I\|_F^2 - \|\hat{\mathcal{C}}_k - I\|_F^2 = &-(1 - \phi_k)((1 - \frac{\hat{\bar{s}}_k^T \hat{\mathcal{C}}_k \hat{\mathcal{C}}_k \hat{\bar{s}}_k}{\hat{\bar{s}}_k^T \hat{\mathcal{C}}_k \hat{\bar{s}}_k})^2 + 2[\frac{\hat{\bar{s}}_k^T \hat{\mathcal{C}}_k \hat{\mathcal{C}}_k \hat{\mathcal{C}}_k \hat{\bar{s}}_k}{\hat{\bar{s}}_k^T \hat{\mathcal{C}}_k \hat{\bar{s}}_k} - (\frac{\hat{\bar{s}}_k^T \hat{\mathcal{C}}_k \hat{\mathcal{C}}_k \hat{\bar{s}}_k}{\hat{\bar{s}}_k^T \hat{\mathcal{C}}_k \hat{\bar{s}}_k})^2]) \\
&- \phi_k((1 - \frac{\hat{\bar{s}}_k^T \hat{\mathcal{C}}_k \hat{\bar{s}}_k}{\hat{\bar{s}}_k^T \hat{\bar{s}}_k})^2 + 2\phi_k[\frac{\hat{\bar{s}}_k^T \hat{\mathcal{C}}_k \hat{\mathcal{C}}_k \hat{\bar{s}}_k}{\hat{\bar{s}}_k^T \hat{\bar{s}}_k} - (\frac{\hat{\bar{s}}_k^T \hat{\mathcal{C}}_k \hat{\bar{s}}_k}{\hat{\bar{s}}_k^T \hat{\bar{s}}_k})^2]) \\
&- \phi_k(1 - \phi_k)[(\frac{\hat{\bar{s}}_k^T \hat{\mathcal{C}}_k \hat{\mathcal{C}}_k \hat{\bar{s}}_k}{\hat{\bar{s}}_k^T \hat{\mathcal{C}}_k \hat{\bar{s}}_k})^2 - (\frac{\hat{\bar{s}}_k^T \hat{\mathcal{C}}_k \hat{\bar{s}}_k}{\hat{\bar{s}}_k^T \hat{\bar{s}}_k})^2].
\end{aligned} \tag{6.2.26}$$

By the Cauchy Schwarz inequality, the terms in brackets are non-negative. We have

$$\|\hat{\bar{\mathcal{B}}}'_{k+1} - I\|_F \leq \|\hat{\mathcal{C}}_k - I\|_F. \tag{6.2.27}$$

From Lemma 6.2.6 and Lemma 6.2.10, we know there exist two constants b_0 and b_1 such that

$$\|\hat{\hat{\mathcal{B}}}'_{k+1} - \hat{\hat{\mathcal{B}}}_{k+1}\|_F \leq (b_0\|\hat{\mathcal{C}}_k - I\|_F + b_1)\epsilon_k^{\min(1,p)}. \tag{6.2.28}$$

Combining (6.2.27) and (6.2.28), we obtain

$$\begin{aligned} \|\hat{\hat{\mathcal{B}}}_{k+1} - I\|_F &\leq \|\hat{\hat{\mathcal{B}}}_{k+1} - \hat{\hat{\mathcal{B}}}'_{k+1}\|_F + \|\hat{\hat{\mathcal{B}}}'_{k+1} - I\|_F \\ &\leq (1 + b_0\epsilon_k^{\min(1,p)})\|\hat{\mathcal{C}}_k - I\|_F + b_1\epsilon_k^{\min(1,p)} \end{aligned} \tag{6.2.29}$$

Since

$$\begin{align*} \|\hat{\mathcal{C}}_k - I\|_F &= \|\hat{\mathcal{T}}_{S_{\zeta_{k+1}}}\hat{H}_*^{-1/2}\hat{\mathcal{T}}^{-1}_{S_{\zeta_{k+1}}}\hat{\mathcal{T}}_{S_{\alpha_k\eta_k}}\hat{\mathcal{B}}_k\hat{\mathcal{T}}^{-1}_{S_{\alpha_k\eta_k}}\hat{\mathcal{T}}_{S_{\zeta_{k+1}}}\hat{H}_*^{-1/2}\hat{\mathcal{T}}^{-1}_{S_{\zeta_{k+1}}} - I\|_F \\ &= \|\hat{H}_*^{-1/2}\hat{\mathcal{T}}^{-1}_{S_{\zeta_{k+1}}}\hat{\mathcal{T}}_{S_{\alpha_k\eta_k}}\hat{\mathcal{B}}_k\hat{\mathcal{T}}^{-1}_{S_{\alpha_k\eta_k}}\hat{\mathcal{T}}_{S_{\zeta_{k+1}}}\hat{H}_*^{-1/2} - I\|_F \tag{6.2.30} \\ &\leq \|\hat{H}_*^{-1/2}\hat{\mathcal{T}}^{-1}_{S_{\zeta_{k+1}}}\hat{\mathcal{T}}_{S_{\alpha_k\eta_k}}\hat{\mathcal{B}}_k\hat{\mathcal{T}}^{-1}_{S_{\alpha_k\eta_k}}\hat{\mathcal{T}}_{S_{\zeta_{k+1}}}\hat{H}_*^{-1/2} - \hat{H}_*^{-1/2}\hat{\mathcal{T}}^{-1}_{S_{\zeta_{k+1}}}\hat{\mathcal{T}}_{S_{\alpha_k\eta_k}}\hat{\mathcal{B}}_k\hat{\mathcal{T}}_{S_{\zeta_k}}\hat{H}_*^{-1/2}\|_F \\ &\quad + \|\hat{H}_*^{-1/2}\hat{\mathcal{T}}^{-1}_{S_{\zeta_{k+1}}}\hat{\mathcal{T}}_{S_{\alpha_k\eta_k}}\hat{\mathcal{B}}_k\hat{\mathcal{T}}_{S_{\zeta_k}}\hat{H}_*^{-1/2} - \hat{H}_*^{-1/2}\hat{\mathcal{T}}^{-1}_{S_{\zeta_k}}\hat{\mathcal{B}}_k\hat{\mathcal{T}}_{S_{\zeta_k}}\hat{H}_*^{-1/2}\|_F \\ &\quad + \|\hat{H}_*^{-1/2}\hat{\mathcal{T}}^{-1}_{S_{\zeta_k}}\hat{\mathcal{B}}_k\hat{\mathcal{T}}_{S_{\zeta_k}}\hat{H}_*^{-1/2} - I\|_F \\ &= \|\hat{H}_*^{-1/2}\hat{\mathcal{T}}^{-1}_{S_{\zeta_{k+1}}}\hat{\mathcal{T}}_{S_{\alpha_k\eta_k}}\hat{\mathcal{B}}_k(\hat{\mathcal{T}}^{-1}_{S_{\alpha_k\eta_k}}\hat{\mathcal{T}}_{S_{\zeta_{k+1}}} - \hat{\mathcal{T}}_{S_{\zeta_k}})\hat{H}_*^{-1/2}\|_F \\ &\quad + \|\hat{H}_*^{-1/2}(\hat{\mathcal{T}}^{-1}_{S_{\zeta_{k+1}}}\hat{\mathcal{T}}_{S_{\alpha_k\eta_k}} - \hat{\mathcal{T}}_{S_{\zeta_k}})\hat{\mathcal{B}}_k\hat{\mathcal{T}}_{S_{\zeta_k}}\hat{H}_*^{-1/2} - I\|_F \\ &\quad + \|\hat{H}_*^{-1/2}\hat{\mathcal{T}}_{S_{\zeta_k}}\hat{\mathcal{B}}_k\hat{\mathcal{T}}_{S_{\zeta_k}}\hat{H}_*^{-1/2} - I\|_F \\ &\leq b_2\|H_*^{-1/2}\mathcal{T}^{-1}_{S_{\zeta_{k+1}}}\mathcal{T}_{S_{\alpha_k\eta_k}}\mathcal{B}_k(\mathcal{T}^{-1}_{S_{\alpha_k\eta_k}}\mathcal{T}_{S_{\zeta_{k+1}}} - \mathcal{T}_{S_{\zeta_k}})H_*^{-1/2}\| \text{ (by Lemma 6.2.6)} \\ &\quad + b_2\|H_*^{-1/2}(\mathcal{T}^{-1}_{S_{\zeta_{k+1}}}\mathcal{T}_{S_{\alpha_k\eta_k}} - \mathcal{T}_{S_{\zeta_k}})\mathcal{B}_k\mathcal{T}_{S_{\zeta_k}}H_*^{-1/2}\| \text{ (by Lemma 6.2.6)} \\ &\quad + \|\hat{\hat{\mathcal{B}}}_k - I\|_F \tag{6.2.31} \\ &\leq b_2\|H_*^{-1/2}\|\|\mathcal{B}_k\|\|\mathcal{T}^{-1}_{S_{\zeta_k}}\mathcal{T}^{-1}_{S_{\alpha_k\eta_k}}\mathcal{T}_{S_{\zeta_{k+1}}} - I\|\|H_*^{-1/2}\| \tag{6.2.32} \\ &\quad + b_2\|H_*^{-1/2}\|\|\mathcal{T}^{-1}_{S_{\zeta_k}}\mathcal{T}^{-1}_{S_{\zeta_{k+1}}}\mathcal{T}_{S_{\alpha_k\eta_k}} - I\|\|\mathcal{B}_k\|\|H_*^{-1/2}\| \tag{6.2.33} \\ &\quad + \|\hat{\hat{\mathcal{B}}}_k - I\|_F \\ &\leq b_3\|\mathcal{B}_k\|\|\mathcal{T}^{-1}_{S_{\zeta_k}}\mathcal{T}^{-1}_{S_{\alpha_k\eta_k}}\mathcal{T}_{S_{\zeta_{k+1}}} - I\| + \|\hat{\hat{\mathcal{B}}}_k - I\|_F \\ &\leq b_4(\|\mathcal{B}_k - I\|_F + \|I\|_F)\epsilon_k + \|\hat{\hat{\mathcal{B}}}_k - I\|_F \text{ (by Lemmas 6.2.6 and 6.2.5)} \\ &= (1 + b_4\epsilon_k)\|\hat{\hat{\mathcal{B}}}_k - I\|_F + b_4\|I\|_F\epsilon_k \tag{6.2.34} \end{align*}$$

we have

$$\|\hat{\hat{\mathcal{B}}}_{k+1} - I\|_F \leq (1 + b_5\epsilon_k^{\min(1,p)})\|\hat{\hat{\mathcal{B}}}_k - I\|_F + b_6\epsilon_k^{\min(1,p)}, \tag{6.2.35}$$

where b_0, b_1, b_2, b_3, b_4, b_5 and b_6 are positive constants. Since $\mathcal{T}_S$ is isometric and $G_{k+1} = G_k = G_* = I$, $\hat{\mathcal{T}}_S$ is orthonormal matrix. Therefore, (6.2.30) and (6.2.31) hold. Since norm is invariant under isometric vector transport, (6.2.32) and (6.2.33) hold. Using inequality (6.2.35) repeatedly, we have

$$\|\hat{\tilde{\mathcal{B}}}_{k+1} - I\|_F \leq \|\hat{\tilde{\mathcal{B}}}_1 - I\|_F \prod_{i=1}^{k}(1 + b_5 \epsilon_i^{\min(1,p)}) + b_6 \sum_{i=1}^{k} \prod_{j=i+1}^{k} (1 + b_5 \epsilon_j^{\min(1,p)}) \epsilon_i^{\min(1,p)}. \qquad (6.2.36)$$

By the relationship between the geometric and arithmetic means and since, by Lemma 6.2.7, $\sum_{i=1}^{\infty} \epsilon_i^{\min(1,p)} \leq \infty$, , we obtain

$$\prod_{i=1}^{k}(1 + b_5 \epsilon_i) \leq (\frac{\sum_{i=1}^{k}(1 + b_5 \epsilon_i^{\min(1,p)})}{k})^k = (1 + b_5 \frac{\sum_{i=1}^{k} \epsilon_i^{\min(1,p)}}{k})^k < b_7$$

for some positive constant b_7. Using this equation for (6.2.36), we know

$$\|\hat{\tilde{\mathcal{B}}}_{k+1} - I\|_F \leq b_7 \|\hat{\tilde{\mathcal{B}}}_1 - I\|_F + b_6 b_7 \sum_{i=1}^{k} \epsilon_i^{\min(1,p)} \leq b_8,$$

where b_8 is a positive constant. Therefore, using the first inequality of Lemma 6.2.6, we have

$$\|\bar{\mathcal{B}}_{k+1}\| \leq \|\hat{\tilde{\mathcal{B}}}_{k+1}\|_F \leq \|\hat{\tilde{\mathcal{B}}}_{k+1} - I\|_F + \|I\|_F < b_9.$$

where b_9 is a positive constant. □

Corollary 6.2.1. *Suppose Assumptions 4.3.1, 4.3.2, 6.2.1 and 6.2.2 hold. $\phi_k \in [0, 1-\delta]$. Then $\mathcal{C}_k = H_{k+1}^{-1/2} \tilde{\mathcal{B}}_k H_{k+1}^{-1/2}$, $\mathcal{B}_k$ are uniformly bounded.*

Proof. The corollary follows immediately from Lemma 6.2.11 and that x^* is nondegenerate. □

Corollary 6.2.2 generalizes a part of [GT82, Proposition 4].

Corollary 6.2.2. *Suppose Assumptions 4.3.1, 4.3.2, 6.2.1 and 6.2.2 hold. $\phi_k \in [0, 1-\delta]$. Then the condition number of $\bar{\mathcal{B}}_k$, $\mathcal{C}_k$ for all k in the sequence are uniformly bounded.*

Proof. The update formula (4.2.3) of $\mathcal{B}_k$ corresponds to an update of its inverse $\mathcal{H}_k = \mathcal{B}_k^{-1}$,

$$\mathcal{H}_{k+1} = \tilde{\mathcal{H}}_k - \frac{\tilde{\mathcal{H}}_k y_k (\tilde{\mathcal{H}}_k^* y_k)^\flat}{(\tilde{\mathcal{H}}_k^* y_k)^\flat y_k} + \frac{s_k s_k^\flat}{s_k^\flat y_k} + \tilde{\phi}_k g(y_k, \tilde{\mathcal{H}}_k y_k) u_k u_k^\flat,$$

where

$$u_k = \frac{s_k}{g(s_k, y_k)} - \frac{\tilde{\mathcal{H}}_k y_k}{g(y_k, \tilde{\mathcal{H}}_k y_k)},$$

and

$$\tilde{\phi}_k = \frac{(1-\phi_k) g^2(y_k, s_k)}{(1-\phi_k) g^2(y_k, s_k) + \phi_k g(y_k, \tilde{\mathcal{H}}_k y_k) g(s_k, \tilde{\mathcal{B}}_k s_k)} \in (0, 1]. \tag{6.2.37}$$

x_k, s_k, y_k generated by $\mathcal{H}_k$'s formula with the Step 3 in Algorithm 3 replaced by $\eta_k = -\mathcal{H}_k \operatorname{grad} f(x_k)$ are the same as those generated by $\mathcal{B}_k$'s formula. Therefore, the statements of Lemma 6.2.9 still hold. The proof of 6.2.10 requires the coefficient of combination $\phi_k \in [0, 1]$ which also holds for $\tilde{\phi}_k$. Therefore, we can use the same idea to obtain a similar result. Note that the reason that Lemma 6.2.11 requires $\phi_k \in [0, 1-\delta]$ is because of the requirement that $\sum_{i=1}^{\infty} \epsilon_i^{\min(1,p)} \leq \infty$. Even though $\tilde{\phi}_k$ is not in $[0, 1-\delta]$, $\sum_{i=1}^{\infty} \epsilon_i^{\min(1,p)} \leq \infty$ still holds since the sequence x_k generated by $\mathcal{H}_k$'s formula is the same as those generated by $\mathcal{B}_k$'s formula. Therefore, using the same idea as Lemma 6.2.11, we obtain $\bar{\mathcal{B}}_k^{-1}$ is bounded. Since x^* is nondegenerate, we obtain $\mathcal{C}_k^{-1}$ is also uniformly bounded. Thus, their condition numbers are uniformly bounded. □

The main convergence result, Theorem 6.2.2, that generalizes a part of [GT82, Proposition 4] can now be proven.

Theorem 6.2.2 (Superlinear Convergence)**.** *Suppose Assumptions 4.3.1, 4.3.2, 6.2.1 and 6.2.2 hold. $\phi_k \in [0, 1-\delta]$ and $\alpha_k = 1$ whenever it satisfies Wolfe conditions (4.2.1) and (4.2.2). Then x_k converges to x^* superlinearly.*

Proof. For each tangent space $\mathrm{T}_x \mathcal{M}$ we consider a basis that is orthonormal with respect to the Riemannian metric g, with $x = x^*, x_0, x_1, \ldots$, and we let a hat denote expressions in these bases. Consequently, for all k, the matrix expression of the inner product g_{x_k} is the identity; in other words, $g(u, v) = \hat{u}^T \hat{v}$ for all $u, v \in \mathrm{T}_k \mathcal{M}$. We have

$$\begin{aligned}
&\|\hat{\mathcal{C}}_k - I\|_F^2 - \|\hat{\bar{\mathcal{B}}}'_{k+1} - I\|_F^2 \\
\leq\ &\|\hat{\mathcal{C}}_k - I\|_F^2 + \|\hat{\bar{\mathcal{B}}}_{k+1} - \hat{\bar{\mathcal{B}}}'_{k+1}\|_F^2 - \|\hat{\bar{\mathcal{B}}}_{k+1} - I\|_F^2 + 2\|\hat{\bar{\mathcal{B}}}_{k+1} - \hat{\bar{\mathcal{B}}}'_{k+1}\|_F \|\hat{\bar{\mathcal{B}}}'_{k+1} - I\|_F \\
\leq\ &\|\hat{\mathcal{C}}_k - I\|_F^2 - \|\hat{\bar{\mathcal{B}}}_{k+1} - I\|_F^2 + b_1 \epsilon_k^{\min(1,p)} \qquad (6.2.38) \\
&\text{(by Lemmas 6.2.10, 6.2.11 and Corollary 6.2.1)} \\
\leq\ &\|\hat{\mathcal{C}}_k - I\|_F^2 - \|\hat{\bar{\mathcal{B}}}_k - I\|_F^2 + \|\hat{\bar{\mathcal{B}}}_k - I\|_F^2 - \|\hat{\bar{\mathcal{B}}}_{k+1} - I\|_F^2 + b_1 \epsilon_k^{\min(1,p)} \\
\leq\ &\|\hat{\bar{\mathcal{B}}}_k - I\|_F^2 - \|\hat{\bar{\mathcal{B}}}_{k+1} - I\|_F^2 + b_2 \epsilon_k^{\min(1,p)} \text{ (by (6.2.34), Lemmas 6.2.11, 6.2.6)} \qquad (6.2.39)
\end{aligned}$$

where b_1, b_2 are positive constants. It follows that

$$\sum_{k=1}^{\infty}(\|\hat{\mathcal{C}}_k - I\|_F^2 - \|\hat{\bar{\mathcal{B}}}'_{k+1} - I\|_F^2) \leq \sum_{k=1}^{\infty}(\|\hat{\bar{\mathcal{B}}}_k - I\|_F^2 - \|\hat{\bar{\mathcal{B}}}_{k+1} - I\|_F^2) + b_2 \sum_{k=1}^{\infty} \epsilon_k^{\min(1,p)}$$

$$= \|\hat{\bar{\mathcal{B}}}_1 - I\|_F^2 + b_2 \sum_{k=1}^{\infty} \epsilon_k^{\min(1,p)} \leq \infty. \text{ (by Lemma 6.2.7)}$$

Hence, we have

$$\lim_{k\to\infty} \|\hat{\mathcal{C}}_k - I\|_F^2 - \|\hat{\bar{\mathcal{B}}}'_{k+1} - I\|_F^2 = 0.$$

Noting (6.2.26), if $\phi \neq 1$, we have

$$\lim_{k\to\infty} \frac{\hat{\bar{s}}^T \hat{\mathcal{C}}_k \hat{\mathcal{C}}_k \hat{\mathcal{C}}_k \hat{\bar{s}}}{\hat{\bar{s}}^T \hat{\mathcal{C}}_k \hat{\bar{s}}} = 0, \lim_{k\to\infty} \frac{\hat{\bar{s}}^T \hat{\mathcal{C}}_k \hat{\mathcal{C}}_k \hat{\bar{s}}}{\hat{\bar{s}}^T \hat{\mathcal{C}}_k \hat{\bar{s}}} = 0. \tag{6.2.40}$$

Using (6.2.40), we obtain

$$\lim_{k\to\infty} \frac{\|(\hat{\mathcal{C}}_k)^{1/2}(\hat{\mathcal{C}}_k - I)\hat{\bar{s}}\|_F}{\|(\hat{\mathcal{C}}_k)^{1/2}\hat{\bar{s}}\|_F} = 0.$$

Since Lemma 6.2.6 holds, we have

$$\lim_{k\to\infty} \frac{\|(\hat{\mathcal{C}}_k)^{1/2}(\hat{\mathcal{C}}_k - I)\hat{\bar{s}}\|}{\|(\hat{\mathcal{C}}_k)^{1/2}\hat{\bar{s}}\|} = 0.$$

Since condition number of $\mathcal{C}_k$ are uniformly bounded (Corollary 6.2.2), we get

$$\lim_{k\to\infty} \frac{\|(\hat{\mathcal{C}}_k - I)\hat{\bar{s}}\|}{\|\hat{\bar{s}}\|} = 0.$$

Since

$$\|H_*^{1/2}\|^2 \frac{\|(\mathcal{C}_k - I)\bar{s}_k\|}{\|\bar{s}_k\|} = \|H_{k+1}^{1/2}\|^2 \frac{\|(H_{k+1}^{-1/2}\tilde{B}_k H_{k+1}^{-1/2} - I)H_{k+1}^{1/2}s_k\|}{\|H_{k+1}^{1/2}s_k\|} \geq \frac{\|(\tilde{B}_k - H_{k+1})s_k\|}{\|s_k\|},$$

we have

$$\lim_{k\to\infty} \frac{\|(\tilde{B}_k - H_{k+1})s_k\|}{\|s_k\|} = 0. \tag{6.2.41}$$

Let $\tilde{s}_k = \alpha_k \eta_k = \mathcal{T}_{S_{\alpha_k\eta_k}}^{-1} s_k$. It follows that

$$\begin{aligned}
\frac{\|(\mathcal{B}_k - H_k)\eta_k\|}{\|\eta_k\|} &= \frac{\|(\mathcal{B}_k - H_k)\tilde{s}_k\|}{\|\tilde{s}_k\|} = \frac{\|(\mathcal{B}_k \mathcal{T}_{S_{\alpha_k\eta_k}}^{-1} - H_k \mathcal{T}_{S_{\alpha_k\eta_k}}^{-1})s_k\|}{\|s_k\|} \\
&= \frac{\|(\mathcal{T}_{S_{\alpha_k\eta_k}} \mathcal{B}_k \mathcal{T}_{S_{\alpha_k\eta_k}}^{-1} - \mathcal{T}_{S_{\alpha_k\eta_k}} H_k \mathcal{T}_{S_{\alpha_k\eta_k}}^{-1})s_k\|}{\|s_k\|} \\
&= \frac{\|(\tilde{\mathcal{B}}_k - H_{k+1} + H_{k+1} - \mathcal{T}_{S_{\alpha_k\eta_k}} H_k \mathcal{T}_{S_{\alpha_k\eta_k}}^{-1})s_k\|}{\|s_k\|} \\
&\leq \frac{\|(\tilde{\mathcal{B}}_k - H_{k+1})s_k\|}{\|s_k\|} + \frac{\|(H_{k+1} - \mathcal{T}_{S_{\alpha_k\eta_k}} H_k \mathcal{T}_{S_{\alpha_k\eta_k}}^{-1})s_k\|}{\|s_k\|} \\
&\to 0. \text{ (by (6.2.41) and Lemma 6.2.8)}
\end{aligned}$$

What is more, from the Assumption 6.2.2 and $\mathcal{B}_k \eta_k = -\operatorname{grad} f(x_k)$, we know

$$\lim_{k\to\infty} \frac{\|\operatorname{grad} f(x_k) + \operatorname{Hess} f(x_k)\eta_k\|}{\|\eta_k\|} = 0.$$

Using the Riemannian Dennis-Moré Condition in Theorem 5.2.4 completes the proof.

□

CHAPTER 7

OPTIMIZING PARTLY SMOOTH FUNCTIONS ON A RIEMANNIAN MANIFOLD

7.1 Introduction

There are applications of interest for which the cost function f is continuously differentiable on much of the domain of the problem but is not differentiable at one or more of the minimizers. For example, in the important application area of computational geometry, the bounding box problem [BA10], and in nonlinear dimension reduction for data analysis and representation, the secant-based projection approach uses such a cost function [BK05]. Clarke generalized the gradient for a class of such functions, specifically for Lipschitz continuous functions on an open set, i.e., no boundary, in [Cla90]. The generalized directional derivative of a Lipschitz continuous function $f : \mathbb{R}^n \to \mathbb{R}$ evaluated at x in the direction v is given by

$$f^o(x; v) = \limsup_{y \to x, \lambda \downarrow 0} \frac{f(y + \lambda v) - f(y)}{\lambda}$$

and the generalized gradient of f at x is the set

$$\partial f(x) = \{\eta \in \mathbb{R}^n : f^o(x; v) \geq \langle v, \eta \rangle \text{ for all } v \text{ in } \mathbb{R}^n\}.$$

This generalized gradient reduces to a single vector, the standard gradient, when f is differentiable at x and to the subdifferential if f is convex but not differentiable at x (since a convex function on an open convex set is locally Lipschitz continuous [RV74]).

For partly smooth functions, the norm of the gradient cannot be used to specify a stationary point. Clarke defines a generalization of a stationary point to be x^* where $f^o(x^*; v) \geq 0$ for all $v \in \mathbb{R}^n$, i.e., there is no descent direction in which to move with a positive step size. We will refer to x^* as a Clarke stationary point. This definition is equivalent to $0 \in \partial f(x^*)$.

Bundle algorithms for nonsmooth locally Lipschitz problems are discussed in Kiwiel's book [Kiw85]. They are reasonably efficient and effective for convex nonsmooth problems but become significantly more complicated for nonconvex nonsmooth problems (see [BLO05] for a discussion).

As an alternative to bundle algorithms for nonconvex nonsmooth problems, Burke, Lewis and Overton [BLO05] develop the gradient sampling algorithm (GS), given in Algorithm 5 using the notation of Table 7.1, which make uses of the gradients of points in the neighborhood of current iterate and avoids computing any element of the generalized gradient. A convergence analysis is also given when the cost function is locally Lipschitz and has bounded level sets, but not necessarily convex. Their numerical results also show that GS algorithm works well even for functions that are not Lipschitz.

Quasi-Newton methods have been proposed to optimize a nonconvex nonsmooth functions but the research is still limited. Lewis and Overton [LO13] give a good overview and demonstrate empirically that BFGS works very well for functions that are locally Lipschitz continuous with bounded level sets. They do not give a convergence analysis. So far, most of the quasi-Newton work has considered the Broyden family of methods, especially BFGS. The trust region with symmetric rank-1 update has not been considered for nonsmooth optimization. Limited-memory BFGS works well in practice in some cases [ZZA^{+}00, Ska10]. However, negative comments on its behavior can be found in [Haa04, YVGS08].

Table 7.1: Glossary of Notation

k: Iteration counter.	μ: Sampling radius reduction factor.
x_k: Current iterate.	θ: Optimality tolerance reduction factor.
γ: Backtracking reduction factor.	m: Sample size.
$\mathcal{L}$: $\{x \vert f(x) \leq f(\tilde{x})\}$.	D: Points of differentiability.
u_{kj}: Unit ball samples.	x_{kj}: Sampling points.
c_1: Armijo parameter.	g_k: Shortest approximate subgradient.
ϵ_k: Sampling radius.	d_k: Search direction.
ν_k: Optimality tolerance.	t_k: Step length.

Optimization algorithms for problems with a partly smooth function in the Euclidean setting are the subject of current research while problems on Riemannian manifolds have received little consideration. RTR-SR1 does not work well for nonsmooth functions and it can stagnate in a neighborhood where the cost function nonsmooth. The limited-memory BFGS has difficulty in the Euclidean nonsmooth case, so we do not expect it will work well for Riemannian problems without more careful consideration of its Euclidean behavior. In this chapter we consider RBFGS and Riemannian GS (RGS) since their Euclidean versions of them not only work well for nonsmooth

Algorithm 5 The gradient sampling algorithm on $\mathbb{R}^n$

```
Input: $x_0 \in \mathcal{L} \bigcap D$, $\gamma \in (0,1)$, $c_1 \in (0,1)$, $\epsilon_0 > 0$, $\nu_0 \geq 0$, $\mu \in (0,1]$, $\theta \in (0,1]$, $k = 0$, and $m \in \{n+1, n+2, \ldots\}$.
Output: Sequence of iterates $x_k$.
1: Let $u_{k1}, \ldots, u_{km}$ be sampled independently and uniformly from $\mathbf{B}$, where $\mathbf{B} = \{x | \|x\| \leq 1\}$ is the closed unit ball and $\| \cdot \|$ is the 2-norm. And set
       $x_{k0} = x_k \text{ and } x_{kj} = x_k + \epsilon_k u_{kj}, j = 1, \ldots, m.$
   If for some $j = 1, \ldots, m$, the point $x_{kj} \notin D$, then STOP; otherwise, set
       $G_k = \text{conv}\{\text{grad} f(x_{k0}), \text{grad} f(x_{k1}), \ldots, \text{grad} f(x_{km})\},$
   and go to Step 2.
2: Let $g_k \in G_k$ solve the quadratic program $\min_{g \in G_k} \|g\|^2$, i.e.,
       $\|g_k\| = \text{dist}(0|G_k) \text{ and } g_k \in G_k.$
   If $\nu_k = \|g_k\| = 0$, STOP. If $\|g_k\| \leq \nu_k$, set $t_k = 0$, $\nu_{k+1} = \theta \nu_k$, and $\epsilon_{k+1} = \mu \epsilon_k$ and go to Step 4; otherwise, set $\nu_{k+1} = \nu_k$, $\epsilon_{k+1} = \epsilon_k$, and $d_k = -g_k / \|g_k\|$, and go to Step 3.
3: Set
       $t_k = \max \gamma^s \text{ subject to } s \in \{0, 1, \ldots\} \text{ and } f(x_k + \gamma^s d_k) < f(x_k) - c_1 \gamma^s \|g_k\|$
   and go to Step 4.
4: If $x_k + t_k d_k \in D$, set $x_{k+1} = x_k + t_k d_k$, $k = k + 1$, and go to Step 1. If $x_k + t_k d_k \notin D$, let $\hat{x}_k$ be any point in $x_k + \epsilon_k \mathbb{B}$ satisfying $\hat{x}_k + t_k d_k \in D$ and
       $f(\hat{x}_k + t_k d_k) < f(x_k) - c_1 t_k \|g_k\|,$
   (such an $\hat{x}_k$ exists due to the continuity of $f$). Then set $x_{k+1} = x_k + t_k d_k$, $k = k + 1$ and go to Step 1.
```

functions but also do not require convexity.

From the per iteration complexity point of view, RBFGS has an advantage over RGS. An expensive quadratic programming computation, is required in each iteration of RGS while RBFGS requires it only when close to convergence (see Section 7.3.2). Additionally, RGS, in general, requires many more gradient evaluations and vector transports in each iteration than RBFGS. We therefore do not expect RGS to be faster than RBFGS in most cases. However, there are two main reasons that we consider RGS. First, GS works well even for functions that are not Lipschitz continuous (see Section 11.4.12 for RGS). Second, it has complete Euclidean convergence analysis while BFGS does not.

In this chapter we define the Riemannian algorithms RGS and RBFGS for nonsmooth functions on a manifold. We investigate empirically their behavior in Chapter 11. A convergence theory is not developed here but is considered in other current research and it appears very likely that a complete theory is possible for RGS while the likelihood for RBFGS is less clear at present.

7.2 Gradient Sampling Algorithm on a Riemannian Manifold

The proposed generalization of the gradient sampling algorithm to Riemannian manifolds is given in Algorithm 6. There are two obvious differences from the Euclidean version. First, the Riemannian version uses retraction and, second, a vector transport is required in Step 1.

The convergence analysis of GS on a Euclidean space has been given by Burke, Lewis and Overton [BLO05]. They proved that when the sampling radius $\epsilon_k = \epsilon$ is fixed and the optimality tolerance ν_k is 0, iterates eventually stay in a neighborhood of a local minimum and the size of the neighborhood is dependent on ϵ. In addition, if ϵ_k and ν_k are not fixed, in other words, $\epsilon_k > 0$, $\nu_k > 0$, $\mu \in (0, 1)$ and $\theta \in (0, 1)$, and iterates $\{x_k\}$ converge to some point $\bar{x}$, then with probability 1, $\bar{x}$ is a Clarke stationary point for f.

7.3 Modifications for RBFGS Algorithms

Significant modifications are required to adapt RBFGS for partly smooth functions. The two most important are a modification to the line search algorithm to determine a step size and a modification to the stopping criterion.

Algorithm 6 The gradient sampling algorithm on d-dimensional Riemannian manifold

```
Input: $x_0 \in \mathcal{L} \bigcap D$, $\gamma \in (0,1)$, $c_1 \in (0,1)$, $\epsilon_0 > 0$, $\nu_0 \geq 0$, $\tau_\nu \in (0,1)$, $\mu \in (0,1]$, $\theta \in (0,1]$, $k = 0$, and $m \in \{d+1, d+2, \ldots\}$.
Output: Sequence of iterates $x_k$.
1: Let $u_{k1}, \ldots, u_{km}$ be sampled independently and uniformly from $\mathbb{B}_k$, where $\mathbb{B}_k = \{v | v \in \mathrm{T}_{x_k}\mathcal{M}, \|v\| \leq 1\}$ is the closed unit ball of tangent space of $x_k$ and $\| \cdot \|$ is the induced norm. And set
       $$x_{k0} = x_k \text{ and } x_{kj} = R_{x_k}(\epsilon_k u_{kj}), j = 1, \ldots, m.$$
   If for some $j = 1, \ldots, m$, the point $x_{kj} \notin D$, then STOP; otherwise, compute
       $$\mathrm{grad}(f(x)) \text{ for } x = x_{kj}, j = 0, \ldots, m.$$
   and use a vector transport to transport the gradients to $\mathrm{T}_{x_k}\mathcal{M}$. Denote the transported gradients as
       $$\eta_{kj} \in T_{x^k}(\mathcal{M}), j = 0, \ldots, m.$$
   set
       $$G_k = \mathrm{conv}\{\eta_{kj}, j = 0, \ldots, m\},$$
   and go to Step 2.
2: Let $g_k \in G_k$ solve the quadratic programming problem $\min_{g \in G_k} \|g\|^2$, i.e.,
       $$\|g_k\| = \mathrm{dist}(0|G_k) \text{ and } g_k \in G_k.$$
   If $\|g_k\| = 0$ and $\nu_k < \tau_\nu$, STOP. If $\|g_k\| \leq \nu_k$, set $t_k = 0$, $\nu_{k+1} = \theta\nu_k$, and $\epsilon_{k+1} = \mu\epsilon_k$ and go to Step 4; otherwise, set $\nu_{k+1} = \nu_k$, $\epsilon_{k+1} = \epsilon_k$, and $d_k = -g_k/\|g_k\|$, and go to Step 3.
3: Set
       $$t_k = \max \gamma^s \text{ subject to } s \in \{0, 1, \ldots\} \text{ and } f(R_{x_k}(\gamma^s d_k)) < f(x_k) - c_1\gamma^s\|g_k\|$$
   and go to Step 4.
4: If $R_{x_k}(t_k d_k) \in D$, set $x_{k+1} = R_{x_k}(t_k d_k)$, $k = k+1$, and go to Step 1. If $R_{x_k}(t_k d_k) \notin D$, perturb $t_k$ and get $\hat{t}_k$ satisfying $R_{x_k}(\hat{t}_k d_k) \in D$ and
       $$f(R_{x_k}(\hat{t}_k d_k)) < f(x_k) - c_1\hat{t}_k\|g_k\|,$$
   (such an $\hat{t}_k$ exists due to the continuity of $f$ and $R$). Set $x_{k+1} = R_{x_k}(\hat{t}_k d_k)$, $k = k+1$ and go to Step 1.
```

7.3.1 Line Search Algorithm for Partly Smooth Function

Lewis and Overton [LO13] provide an inexact line search algorithm for nonsmooth function optimization in a Euclidean space. They prove that the algorithm always gives a step size that satisfies the Wolfe conditions under some reasonable assumptions. Since the line search algorithm on a Riemannian manifold also deals with a function on $\mathbb{R}$, the conclusion of the Euclidean setting guaranteeing the existence of a step size satisfying the conditions is extended to Riemannian manifolds naturally.

The line search objective function, $h(t) = f(R_x(t\eta)) - f(x)$, is a partly smooth function defined on $\mathbb{R}$. In order to obtain the desired result, we make the following assumption which is the same as [LO13, Assumption 4.1].

Assumption 7.3.1. *The function $h : \mathbb{R}_+ \to \mathbb{R}$ is absolutely continuous on every bounded interval, and bounded below. Furthermore, it satisfies*

$$h(0) = 0 \text{ and } s = \limsup_{t \downarrow 0} \frac{h(t)}{t} < 0.$$

If f is differentiable at x, Assumption 7.3.1 is not required and s is $g(\operatorname{grad} f(x), \eta)$.

The Wolfe conditions for a partly smooth function on $\mathbb{R}$ are

$$\begin{aligned} A(t) :& h(t) < c_1 st, \\ W(t) :& h \text{ is differentiable at } t \text{ with } h'(t) > c_2 s, \end{aligned}$$

where $0 < c_1 < c_2 < 1$. Notice that since the second condition requires that the selected step size t is such that $h(t)$ is differentiable, f is differentiable at each iterate $R_x(t\eta)$. The Wolfe conditions for the line search here can be shown to be the same as the Wolfe conditions (4.2.1) and (4.2.2) derived earlier for Riemannian quasi-Newton line search methods such the Riemannian Broyden family. The Algorithm 7 is the inexact line search algorithm that determine the step size under the assumption

$$\lim_{t \uparrow \bar{t}} h'(t) \text{ exists in} [-\infty, +\infty] \text{ for all } \bar{t} > 0.$$

Algorithm 7 Inexact line search for partly smooth function

```
$\alpha \leftarrow 0$
$\beta \leftarrow +\infty$
$t \leftarrow 1$
loop
  if $A(t)$ fails then
    $\beta \leftarrow t$
  else if $W(t)$ fails then
    $\alpha \leftarrow t$
  else
    break
  end if
  if $\beta < +\infty$ then
    $t \leftarrow (\alpha + \beta)/2$
  else
    $t \leftarrow 2\alpha$
  end if
end loop
```

7.3.2 Stopping Criterion of RBroyden Family Algorithms for a Partly Smooth Function

Since the function is partly smooth, we cannot expect the norms of gradients go to zero. We need a method to check whether a subsequence of the iterates defines a suitably small region containing a Clarke stationarity point. There is a Euclidean space method in [LO13, Section 6.3]. Let J be a positive integer which is greater than the dimension of the Euclidean space and let τ_x and τ_d be two small positive user-specified tolerances. Define $j_0 = 1$ and $G_0 = \{\operatorname{grad} f_0\}$ and, for $k = 1, 2, \dots$ define

$$j_k = 1, G_k = \{\operatorname{grad} f_k\} \text{ if } \|x_k - x_{k-1}\| > \tau_x,$$

$$j_k = j_{k-1} + 1, G_k = \{\operatorname{grad} f_{k-j_k+1}, \dots, \operatorname{grad} f_k\} \text{ if } \|x_k - x_{k-1}\| \leq \tau_x, \text{ and } j_{k-1} < J$$

$$j_k = J, G_k = \{\operatorname{grad} f_{k-J+1}, \dots, \operatorname{grad} f_k\} \text{ if } \|x_k - x_{k-1}\| \leq \tau_x, \text{ and } j_{k-1} < J.$$

By construction, G_k is a set of $j_k \leq J$ gradients evaluated at points near x_k. The smallest vector in the convex hull of the set,

$$d_k = \arg\min\{\|d\| : d \in \operatorname{conv} G_k\},$$

is obtained by solving a convex quadratic program in j_k variables. If $d_k = 0$ or $\|d_k\|$ is sufficiently small then a Clarke stationary point is in a region defined by the iterates with gradients in G_k. If both τ_x and τ_d are small then this region is also small and x_k is near the Clarke stationary point.

We can generalize this idea to a Riemannian manifold. The only difference, the definition of G_k, is described here. Define $j_0 = 1$ and $G_0 = \{\operatorname{grad} f_0\}$ and, for $k = 1, 2, \ldots$ and define

$$\begin{aligned} j_k = 1, G_k = \{\operatorname{grad} f_k\} &\text{ if } \operatorname{dist}(x_k, x_{k-1}) > \tau_x, \\ j_k = j_{k-1} + 1, G_k = \{\operatorname{grad} f_{k-j_k+1}^{(k)}, \ldots, \operatorname{grad} f_{k-1}^{(k)}, \operatorname{grad} f_k^{(k)}\} &\text{ if } \operatorname{dist}(x_k, x_{k-1}) \le \tau_x, \text{ and } j_{k-1} < J \\ j_k = J, G_k = \{\operatorname{grad} f_{k-J+1}^{(k)}, \ldots, \operatorname{grad} f_{k-1}^{(k)}, \operatorname{grad} f_k^{(k)}\} &\text{ if } \operatorname{dist}(x_k, x_{k-1}) \le \tau_x, \text{ and } j_{k-1} < J. \end{aligned}$$

where $\operatorname{grad} f_i^{(j)} = \mathcal{T}_{R_{x_i}^{-1}(x_j)} \operatorname{grad} f(x_i)$. Note that this definition, as given, requires the repeated transport of the gradients in the set G_k for each new x_{k+1}. The complexity implications of this depend upon considerations of the possibility of an intrinsic approach for vector transport and representation of tangent vectors in a set of related tangent spaces (see Chapter 9).

CHAPTER 8

RIEMANNIAN OPTIMIZATION AND CONSTRAINED OPTIMIZATIONS ON EUCLIDEAN SPACE

8.1 Introduction

Constrained optimization considers a problem of the type

$$\min f(x), \text{ subject to } x \in \mathcal{X} \subseteq \mathbb{R}^n, \tag{8.1.1}$$

while Riemannian optimization considers

$$\min f(x), \text{ subject to } x \in \mathcal{M}, \tag{8.1.2}$$

where $\mathcal{X}$ is compact, $\mathcal{M}$ is a Riemannian manifold and $\mathbb{R}^n$ denotes an n-dimensional vector space with the metric unspecified. $\mathbb{E}^n$ denotes n-dimensional Euclidean space, i.e., n dimension vector space with the standard Euclidean metric. The domains of (8.1.1) and (8.1.2) are different and, very importantly, neither domain subsumes the other. Riemannian optimization is more general in the sense that constrained optimization requires the domain to be a subset of $\mathbb{R}^n$, and in some cases, such as standard nonlinear programming, the domain is characterized by specific algebraic equality and inequality constraints, while the domain of Riemannian optimization may not be a subset of $\mathbb{R}^n$. It is allowed to be a more abstract structure, e.g., a quotient or infinite dimensional space, and even when it is a subset of $\mathbb{R}^n$ the choice of representation of local tangent spaces can be chosen to algorithmic advantage. Conversely, constrained optimization is more general in the sense that the subset $\mathcal{X}$ may be not a Riemannian manifold.

In this chapter, we compare the concepts and objects involved in constrained optimization on $\mathbb{R}^n$ with the closest form of Riemannian optimization, i.e., when $\mathcal{M}$ is a submanifold of $\mathbb{R}^n$. The chapter is organized as follows. Section 8.2 briefly summarizes the ideas of constrained optimization on $\mathbb{E}^n$. Section 8.3 shows possibilities of converting a constrained optimization problem into a Riemannian optimization problem. Finally, Section 8.4 compares variants of the gradient projection method with Riemannian optimization methods.

8.2 Constrained Optimization

Constrained optimization has rich history and the discussion can be found in [Kel99, Ber03, NW06] and the references therein. There are many methods, i.e., feasible direction methods, gradient projection methods, penalty methods, barrier methods and dual approaches. To compare with Riemannian optimization, we briefly introduce the ideas of feasible direction methods, penalty methods and barrier methods in this section. Gradient projection methods have some features that are close in spirit to some Riemannian optimization algorithms and are discussed in Section 8.4. Dual approaches involve identifying a dual problem for the primal problem and solving the dual or both in an effective manner. These are not directly related to the Riemannian algorithms of this dissertation and we therefore defer discussions of potential relationships to later work.

8.2.1 Feasible Direction Methods

A vector $x \in \mathbb{R}^n$ is called feasible if it is in the set $\mathcal{X}$. A direction d of x is called feasible if $\{x + \alpha d, \alpha \in [0, t]\} \subseteq \mathcal{X}$ for some $t > 0$. Note that the feasible direction may not exist for arbitrary constraints, e.g., $\mathcal{X} = \{x \in \mathbb{R}^n | x^T x = 1\}$. Therefore, feasible direction methods are not applicable to all problems. The analysis in [Ber03] requires the constrained set $\mathcal{X}$ to be convex to guarantee that a feasible direction exists. The discussion in this section follows the requirement of convexity of $\mathcal{X}$.

A feasible direction method starts with forming a feasible iterate x_0 and generates a sequence of iterates $\{x_k\}$ by

$$x_{k+1} = x_k + \alpha_k d_k,$$

where d_k is a feasible descent direction of x_k and α_k is a step size. The descent direction can be characterized by

$$\operatorname{grad} f(x_k) d_k < 0.$$

where $\operatorname{grad} f(x_k)$ is the gradient with respect to the Euclidean metric.

A sequence of directions $\{d_k\}$ is called gradient related, if for any subsequence $\{x_k\}|_{k \in \mathcal{K}}$ that converges to a nonstationary point, the corresponding subsequence $\{d_k\}|_{k \in \mathcal{K}}$ is bounded and satisfies

$$\lim_{k \to \infty} \sup_{k \in \mathcal{K}} \operatorname{grad} f(x_k)^T d_k < 0.$$

Given a direction d_k the step size α_k is determined by the line search algorithm associated with the particular method. Bertsekas [Ber03, page 217] uses the limited minimization rule or the Armijo rule as the line search algorithm.

Definition 8.2.1 (Limited minimization rule)**.** *The step size α_k is chosen so that*

$$f(x_k + \alpha_k d_k) = \min_{\alpha \in [0,1]} f(x_k + \alpha d_k).$$

Definition 8.2.2 (Armijo rule)**.** *Fixed scalars $\beta \in (0, 1)$ and $\sigma \in (0, 1)$ are chosen, and we set $\alpha_k = \beta^{m_k}$, where m_k is the first nonnegative integer m for which*

$$f(x_k) - f(x_k + \beta^m d_k) \geq -\sigma \beta^m \operatorname{grad} f(x_k)^T d_k.$$

Proposition 2.2.1 [Ber03] shows that if the sequence of directions $\{d_k\}$ is gradient related and α_k is chosen by the limited minimization rule or the Armijo rule, then every limit point of $\{x_k\}$ generated by a feasible direction method is a stationary point.

8.2.2 Barrier Methods

A point x of a set $\mathcal{S} \subset \mathbb{R}^n$ is called an interior point of $\mathcal{S}$ is there exists an open set $\mathcal{U}$ of $\mathbb{R}^n$ such that $x \in \mathcal{U}$ and $\mathcal{U} \subset \mathcal{S}$. Suppose the feasible set $\mathcal{X}$ is $\mathcal{X}_1 \cap \mathcal{X}_2$. Let $\mathcal{X}_2^o$ denote the set of interior points of $\mathcal{X}_2$. Given any $x \in \mathcal{X}_2$ and any $\delta > 0$, if there exists a $\tilde{x} \in \mathcal{X}_2^o$ such that $\|\tilde{x} - x\|_2 < \delta$, then barrier methods can be used. In barrier methods, a function called "barrier function" is defined to force each iterate in the interior point set of $\mathcal{X}_2$. A barrier function $B(x)$ is defined over $\mathcal{X}^o$ such that it is continuous and approaches infinity when x goes to $\mathcal{X}_2 \setminus \mathcal{X}_2^o$. Barrier methods generate a sequence $\{x_k\}$ by

$$x_k = \arg \min_{x \in \mathcal{X}_1} f(x) + \epsilon_k B(x), \tag{8.2.1}$$

where ϵ_k is a given parameter and satisfies $\epsilon_{k+1} < \epsilon_k$, $\epsilon \to 0$. We can see by adding the barrier function, the constraints $x \in \mathcal{X}_2$ is removed. If $\mathcal{X}_2$ is the feasible set $\mathcal{X}$, the problem (8.1.1) becomes a sequence of unconstrained optimization problems. In the more typical case, the use of the barrier function allows the removal of the constraints from explicit consideration as iterates remain in $\mathcal{X}_2$. The algorithm concentrates on enforcing explicitly the constraints enforcing membership of the iterates in $\mathcal{X}_1$. Notice that since each iterate x_k is an interior point of $\mathcal{X}_2$, barrier methods are also called interior point methods.

Proposition 4.1.1 [Ber03] proves that every limit point of a sequence $\{x_k\}$ generated by a barrier method is a global minimum of the constrained optimization problem (8.1.1).

If $\mathcal{X}_2$ is given by inequalities,

$$\mathcal{X}_2 = \{x | g_j(x) \leq 0, j = 1, 2, \ldots, r\},$$

then there are two standard barrier functions, i.e., a logarithmic function

$$B(x) = -\sum_{j=1}^{r} \ln(-g_j(x)),$$

and an inverse function

$$B(x) = -\sum_{j=1}^{r} \frac{1}{g_j(x)}.$$

The logarithmic barrier function is commonly used in many algorithms (see the descriptions in [Kar84, NW06, Ber03]).

Finding each iterate (8.2.1) requires the solution of an optimization problem. In practice, it is not necessary to solve it exactly. In [Ber03], a strategy of solving (8.2.1) approximately is discussed for the linear programming problem

$$\min c^T x, \text{ subject to } Ax = b, x \geq 0,$$

where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$. Instead of solving (8.2.1) exactly, some method is applied for a few iterations, e.g., a Newton method with one iteration.

8.2.3 Penalty Methods

Contrary to barrier methods which force each iterate into an interior point set, penalty methods allow iterates to be outside the feasible set $\mathcal{X}$. Penalty methods consider a sequence of problems and find $\{x_k\}$ by

$$x_k = \arg\min_{x \in \mathbb{R}^n} f(x) + c_k P(x),$$

where c_k is a given positive parameter and satisfies $c_{k+1} > c_k$, $c_k \to \infty$, $P(x)$ is a penalty function that is continuous, and satisfies the condition that $P(x) \geq 0$ and $P(x) = 0$ if and only if $x \in \mathcal{X}$. (Note that x_k is a global minimum of the problem at step k.) Similar to barrier methods, penalty methods turn a constrained optimization problem to a sequence of unconstrained optimization problems.

There is no convergence proof for general penalty methods. However, the convergence analysis can be shown under some conditions. Suppose $\mathcal{X}$ is given by equalities, the functions in the equalities are differentiable and $f(x)$ is C^1, then the problem (8.1.1) becomes

$$\min f(x), \text{ subject to } h(x) = 0, \tag{8.2.2}$$

where $h(x) : \mathbb{R}^n \to \mathbb{R}^m$ and $h(x) \in C^1$. Theorem 17.1 in [NW06] proves that for the problem (8.2.2) if the penalty function $P(x)$ is chosen to be $\|h(x)\|_2^2$, then every limit point of the sequence $\{x_k\}$ is a global minimum of the problem (8.1.1).

If the penalty function $P(x)$ is chosen such that it is possible to solve (8.1.1) with one single unconstrained optimization problem, then $P(x)$ is called an exact penalty function. For the equality constrained problem (8.2.2), Bertsekas [Ber03] discusses two exact penalty functions, i.e., a nondifferentiable penalty function

$$P(x) = \max_{i=1,\dots,m} |h_i(x)|,$$

and a differentiable penalty function

$$P(x, \lambda) = \lambda^T h(x) + \frac{1}{2}\|W(x) \operatorname{grad}_x L(x, \lambda)\|_2^2 + \frac{c}{2}\|h(x)\|_2^2,$$

where $L(x, \lambda) = f(x) + \lambda^T h(x)$, $\lambda \in \mathbb{R}^m$, c is a positive parameter and $W(x)$ is any continuously differentiable $m \times n$ matrix such that the $m \times m$ matrix $W(x) \operatorname{grad} h(x)$ is nonsingular for all x.

Even though the choices of penalty functions above are based on equality constraints, they are applicable also to a constraints that contains inequalities. Consider the problem

$$\min f(x), \text{ subject to } h(x) = 0, \text{ and } g(x) \leq 0,$$

where $g(x) : \mathbb{R}^n \to \mathbb{R}^r$. It can be converted to an equality constrained problem by adding new variables,

$$\min f(x), \text{ subject to } h(x) = 0, \text{ and } g_i(x) + z_i^2 = 0, i = 1, \dots, r.$$

A popular variation of a penalty method, that can also be viewed as a dual algorithm, is the augmented Lagrangian method where for problem (8.2.2), the iterates $\{x_k\}$ are generated by

$$x_k = \arg\min_{x \in \mathcal{X}} f(x) + \lambda_k^T h(x) + \frac{c_k}{2}\|h(x)\|_2^2,$$

where $\lambda_k \in \mathbb{R}^m$. Proposition 4.2.1 in [Ber03] proves that for the augmented Lagrangian method if λ_k are bounded, then every limit point of the sequence $\{x_k\}$ is a global minimum of the problem (8.1.1).

8.3 Riemannian Optimization

Riemannian optimization problems of the type considered in this chapter and Euclidean optimization problems have the similarity that in both cases the cost function is independent of the feasible set and the metric of the space. They have the obvious main difference that in the former case the feasible set is a submanifold of $\mathbb{R}^n$ while in the latter it is merely a subset of $\mathbb{R}^n$. However, there is a key difference that can be exploited to algorithm and computational advantage in the Riemannian case. The Euclidean metric is global in that it does not depend on the particular elements of the space at which it is evaluated while the Riemannian metric varies with the element of the manifold that determines the tangent space in which the metric is to be evaluated. The flexibility of the metric is often a key computational and analytical aspect of efficiency.

Clearly, not all feasible sets in $\mathbb{R}^n$ are manifolds. Nevertheless, it is possible to change non-Riemannian manifold constraints into a Riemannian manifold constraint by substitution. Example 8.3.1 is a method to check whether an equality constrained set is a Riemannian manifold. It is a direct consequence of Proposition 3.3.4 in [AMS08]. Example 8.3.2 shows an idea to turn a non-Riemannian set into a Riemannian manifold. Example 8.3.3 illustrates a possible approach to turn an inequality constrained set to be a Riemannian unconstrained set.

Example 8.3.1. *A nonempty feasible set defined by equality constraints*

$$\mathcal{X} = \{x | h(x) = 0, h(x) : \mathbb{R}^n \to \mathbb{R}^m, h(x) \in C^\infty\}$$

is a Riemannian manifold with the endowed metric from the Euclidean space if the Jacobian matrix of $h(x)$ has a constant rank for all $x \in h^{-1}(0)$.

Example 8.3.2. *A feasible set that is a simplex is specified by*

$$\mathcal{X} = \{x = (x_1, x_2 \ldots, x_n)^T | x \geq 0, \sum_{i=1}^{n} x_i = r\},$$

where r is a positive constant. However, it is not a differentiable manifold. By substitution $y_i^2 = x_i, i = 1, \ldots, n$, we have

$$\mathcal{Y} = \{y = (y_1, y_2, \ldots, y_n)^T | \|y\|_2^2 = r\},$$

which is a sphere.

Example 8.3.3. *A feasible set defined by the upper half plane is*

$$\mathcal{X} = \{(x, y) \in \mathbb{R}^2 | y \geq 0\}.$$

It is a compact set with an inequality constraint. Using the same idea as barrier methods, we consider the interior points of $\mathcal{X}$,

$$\mathcal{X}^o = \{(x, y) \in \mathbb{R}^2 | y > 0\}.$$

It can be shown that by imposing a metric

$$\langle (u_1, v_1), (u_2, v_2) \rangle_{(x,y)} = \frac{u_1 u_2 + v_1 v_2}{y^2},$$

$\mathcal{X}^o$ is a hyperbolic manifold, where $(x, y) \in \mathcal{X}^o$ and $(u_1, v_1), (u_2, v_2) \in \mathrm{T}_{(x,y)} \mathcal{X}^o$.

It should also be noted that once the manifold is specified the flexibility of the metric that varies with the point on the manifold is accompanied by the freedom to specify the algebraic characterization of the local tangent space, e.g., the basis used, and the characterization of the manifold elements themselves. The approach is not restricted to the particular algebraic constraints given by the problem only the associated geometry.

8.4 Comparison of Riemannian Optimization and Gradient Projection Methods

Gradient projection methods project a proposed iterate $\tilde{x}_{k+1} = x_k + \alpha_k d_k$ onto the feasible set $\mathcal{X}$, which does not require a feasible direction or modify the cost function or change the domain. Therefore, the features of gradient projection methods are close to those of Riemannian optimization and we compare them in this section. Feasible direction methods require that search directions are feasible while Riemannian optimization does not. Both barrier methods and penalty methods add extra terms to the objective function and change the feasible set while Riemannian optimization does not change the cost function and the domain. Therefore, we do not explicitly compare these three kinds of methods with the Riemannian optimizations since they are significantly different. However, some ideas from feasible direction, barrier and penalty methods appear occasionally and naturally in the discussion of gradient projection methods and these are mentioned in this section.

Three frameworks of gradient projection methods are considered. The first, discussed in [LY08, Chapter 12], is for problems with domains given by nonlinear inequalities and equalities constraints, i.e., $\mathcal{X} = \{x \in \mathbb{R}^n | h(x) = 0, g(x) \leq 0\}$ where $h(x) : \mathbb{R}^n \to \mathbb{R}^m$, $g(x) : \mathbb{R}^n \to \mathbb{R}^r$ and $g(x), h(x) \in C^3$. The second is for problems with convex domains and is discussed in [Ber03, Chapter 2]. The third is discussed in [Kel99, Chapter 5] and is for bounds constrained problems with feasible set $\mathcal{X} = \{x \in \mathbb{R}^n | L_i \leq (x)_i \leq U_i\}$.

8.4.1 Nonlinear Inequalities and Equalities Constraints

The framework of the gradient projection method generalizes the steepest descent method of unconstrained optimization problems to constrained optimization problems. Luenberger and Ye [LY08] contains a recent presentation of Luenberger's 1972 approach to nonlinear inequalities and equalities constraints. The basic idea is given in the following. Given a feasible iterate x_k, the active constraints can be found and we use $\tilde{h}(x) = 0 \in \mathbb{R}^{\tilde{m}}$ to denote the constraints, where the active constraints are the constraints given by equations and $m \leq \tilde{m} \leq m + r$. A search direction d_k is obtained by projecting the negative gradient onto the subspace of $\mathbb{R}^n$ tangent to the surface $\tilde{h}(x) = 0$ at x_k. The search direction is not necessarily a feasible direction and a method of pulling the proposed iterate $\tilde{x}_{k+1} = x_k + \alpha_k d_k$ onto the surface is required.

The basic idea of the gradient projection method for nonlinear inequalities and equalities constraints is similar to the Riemannian steepest descent method for a submanifold of $\mathbb{E}^n$. If $\tilde{h}(x) = 0$ defines a Riemannian manifold $\mathcal{M}_k$ in $\mathbb{R}^n$ and the metric of the manifold is endowed from $\mathbb{E}^n$, then the subspace of $\mathbb{R}^n$ tangent to the surface $\tilde{h}(x) = 0$ at x_k is $\mathrm{T}_{x_k} \mathcal{M}_k$ and the Riemannian gradient of the cost function f for the manifold $\mathcal{M}_k$ is $P_k \operatorname{grad} f(x_k)$, where P_k is the projection onto $\mathrm{T}_{x_k} \mathcal{M}_k$. Therefore, a search direction of a Riemannian steepest descent, a negative Riemannian gradient $-P_k \operatorname{grad} f(x_k)$, is identical to the search direction d_k used in the gradient projection method. A method of pulling a proposed iterate onto the surface $\tilde{h}(x) = 0$ plays a same role as a retraction.

Luenberger and Ye [LY08] point out two difficulties of the gradient projection method for nonlinear inequalities and equalities constraints. The first problem is related to the variations of active constraints when a proposed iterate is returned to the feasible set. Suppose a gradient projection method has a proposed iterate $\tilde{x}_{k+1} = x_k - \alpha_k P_k \operatorname{grad} f(x_k)$ that does not necessarily satisfy the active constraints $\{x \in \mathbb{R}^n | \tilde{h}(x) = 0\}$. Some method is required to pull the $\tilde{x}_{k+1}$ back to satisfying active constraints. However, even though the new point satisfies $\tilde{h}(x) = 0$, it may not be in the feasible set $\mathcal{X} = \{x \in \mathbb{R}^n | h(x) = 0, g(x) \leq 0\}$ since the active constraints may have changed and more care must be taken to handle the problem.

The second problem discussed by Luenberger and Ye concerns returning to the feasible set from points outside the set. They give a method of returning to the set that finds a feasible point x_{k+1} such that $\tilde{x}_{k+1} - x_{x+1}$ is perpendicular to the subspace tangent to the surface $\tilde{h}(x) = 0$ at x_k. The x_{k+1} may not always exist but it does when the step size α_k is sufficiently small. Therefore, it follows that

$$x_{k+1} = \tilde{x}_{k+1} + J_{\tilde{h}}(x_k)v,$$

for some vector v in $\mathbb{R}^{\tilde{m}}$, which yields

$$0 = h(x_{k+1}) = h(\tilde{x}_{k+1} + J_{\tilde{h}}(x_k)v).$$

By linearizing the equation at x_k, we have

$$0 = h(\tilde{x}_{k+1} + J_{\tilde{h}}(x_k)v) \simeq h(\tilde{x}_{k+1}) + J_{\tilde{h}}(x_k)J_{\tilde{h}}(x_k)^T v.$$

The approximation is accurate when $\|v\|_2$ and $\|\tilde{x}_{k+1} - x_k\|$ are small. Now, we obtain the first order approximation

$$\begin{aligned} v &= -(J_{\tilde{h}}(x)J_{\tilde{h}}(x)^T)^{-1} h(\tilde{x}_{k+1}), \\ x_{k+1} &= \tilde{x}_{k+1} - J_{\tilde{h}}(x_k)(J_{\tilde{h}}(x)J_{\tilde{h}}(x)^T)^{-1} h(\tilde{x}_{k+1}). \end{aligned}$$

In the Riemannian setting, the first problem does not exist since there is no variation of active constraints. The second problem also does not exist for Riemannian optimization since the function, called retraction, of pulling a proposed iterate back to the feasible set is defined in a different way. A retraction R for a submanifold $\mathcal{M}$ of $\mathbb{E}^n$ is defined to satisfy

(i) $R_x(0) = x$ for all $x \in \mathcal{M}$,

(ii) $\frac{dR_x(t\eta)}{dt} = \eta$, for all $x \in \mathcal{M}$ and all $\eta \in \mathrm{T}_x \mathcal{M}$.

The properties are sufficient to guarantee many convergence results. For a given Riemannian manifold, there are more than one retraction (Figure 8.1) and the method given by Luenberger and Ye is a particular retraction. This flexibility of retraction allows us to choose an efficient and effective one to improve the performance.

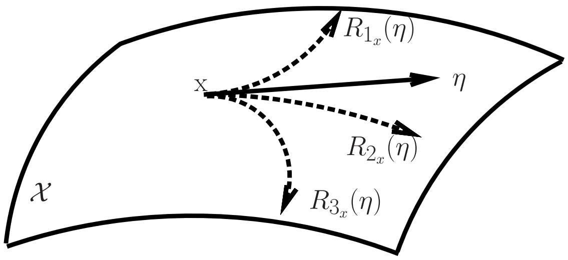


Figure 8.1: Since a retraction only requires local information, $R_x(0) = x$ and $\frac{d}{dt}R_x(t\eta)|_{t=0} = \eta$, there are many retractions for a given manifold. Above figure shows 3 retractions R_1, R_2 and R_3 as examples.

When Luenberger and Ye analyze the convergence rate of the gradient projection method, they consider a problem that has only equalities constraints

$$\mathcal{X} = \{x \in \mathbb{R}^n | h(x) = 0, \text{ Jacobian of } h(x) \text{ has a constant rank.}\}$$

which is a manifold. When defining a geodesic on $\mathcal{X}$, they use the metric from the embedding space, the Euclidean space $\mathbb{E}^n$. The "line" in the line search algorithm is the geodesic, which means the method of returning to the feasible set is the exponential mapping. We can conclude that the algorithm Luenberger and Ye analyze in their framework of the gradient projection method is the Riemannian steepest descent algorithm with a retraction chosen to be the exponential mapping. In fact, this approach is the work that motivated the notion of making Riemannian optimization efficient by finding a replacement for the generically costly exponential mapping.

8.4.2 Convex Set Constraints

The gradient projection method based on Bertsekas [Ber03] generates a sequence of iterates $\{x_k\}$ by

$$x_{k+1} = x_k + \alpha_k(\bar{x}_k - x_k), \tag{8.4.1}$$

where

$$\bar{x}_k = [x_k - s_k \operatorname{grad} f(x_k)]^+,$$

$\alpha_k \in (0, 1]$, s_k is a positive number and $[x]^+$ denotes $\arg\min_{z \in \mathcal{X}} \|z - x\|_2$. Convexity guarantees the uniqueness of $[x]^+$ for all $x \in \mathbb{R}^n$. In practice, finding $\arg\min_{z \in \mathcal{X}} \|z - x\|_2$ may be expensive and therefore $\mathcal{X}$ usually has a relatively simple structure such that finding $[x]^+$ is cheap.

Either α_k or s_k can be viewed as a step size in the update (8.4.1). If s_k is some fixed number and α_k is set to be a step size, then the gradient projection method is a feasible direction method with feasible direction $d_k = \bar{x}_k - x_k$. Note that if $\mathcal{X}$ is not convex, then $\bar{x}_k - x_k$ may be not a feasible direction. If α_k is 1 and s_k is taken to be a step size, then the update of the gradient projection method is

$$x_{k+1} = [x_k - s_k \operatorname{grad} f(x_k)]^+. \tag{8.4.2}$$

This update is working on the projection arc and is similar to Riemannian optimization since a feasible direction is not required.

The convergence analysis of the gradient projection method is given in [Ber03]. We consider three line search algorithms used by Bertsekas [Ber03, page 230].

Definition 8.4.1 (Limited minimization rule)**.** *s_k is chosen to be a constant s and α_k is chosen so that*

$$f(x_k + \alpha_k(\bar{x}_k - x_k)) = \min_{\alpha_k \in [0,1]} f(x_k + \alpha(\bar{x}_k - x_k)).$$

Definition 8.4.2 (Armijo rule along the feasible direction)**.** *s_k is chosen to be a constant s. Fixed scalars $\beta \in (0,1)$ and $\sigma \in (0,1)$ are chosen, and then $\alpha_k = \beta^{m_k}$, where m_k is the first nonnegative integer m for which*

$$f(x_k) - f(x_k + \beta^m(\bar{x}_k - x_k)) \geq -\sigma\beta^m \operatorname{grad} f(x_k)^T(\bar{x}_k - x_k).$$

Definition 8.4.3 (Armijo rule along the projection arc)**.** *α_k is fixed to be a unity, $\alpha_k \equiv 1$. Fixed scalars $\bar{s} > 0$, $\beta \in (0,1)$ and $\sigma \in (0,1)$ are chosen, and we set $s_k = \beta^{m_k}\bar{s}$, where m_k is the first nonnegative integer m for which*

$$f(x_k) - f([x_k - \beta^m \bar{s} \operatorname{grad} f(x_k)]^+) \geq \sigma \operatorname{grad} f(x_k)^T(x_k - [x_k - \beta^m \bar{s} \operatorname{grad} f(x_k)]^+).$$

Proposition 2.3.1 in [Ber03] shows that if the step sizes α_k are chosen by the limited minimization rule or by the Armijo rule along the feasible direction, then every limit point of $\{x_k\}$ is stationary. Proposition 2.3.2 in [Ber03] proves that if the gradient of f is Lipschitz continuous, i.e., $\|\operatorname{grad} f(x) - \operatorname{grad} f(y)\| \leq L\|x-y\|, \forall x, y \in \mathcal{X}$, α_k is 1, s_k is a constant s and $0 < s < 2/L$, then every limit point of $\{x_k\}$ is stationary. Proposition 2.3.3 in [Ber03] proves that if s_k is taken to be a constant, then the Armijo rule along the projection arc line search algorithm stops in finite steps. Furthermore, if the Armijo rule along the projection arc is applied to choose s_k, then every limit point of $\{x_k\}$ is stationary.

The convergence rates of gradient projection methods are essentially the same as those of unconstrained steepest descent methods. The local convergence rate depends on the eigenvalues of the Hessian at the minimum.

The two most important algorithmic aspects that influence the performance difference between gradient projection methods and Riemannian methods such as Riemannian steepest descent are the geometric relationship of the direction vector used for the line search with the boundary of the feasible set and the method of pulling back candidate iterates to the feasible set. If a subspace of $\mathbb{R}^n$ tangent to a point x on the boundary of the convex feasible set $\mathcal{X}$ exists then the update in the Riemannian steepest descent algorithm with line search is defined to be

$$x_{k+1} = R_{x_k}(x_k - s_k P_{x_k} \operatorname{grad} f(x_k)), \tag{8.4.3}$$

where s_k is a step size and $P_{x_k}(v)$ is the projection that projects v on to the subspace of $\mathbb{R}^n$ tangent to x_k. Also, $P_{x_k} \operatorname{grad} f(x_k)$ is the Riemannian gradient when the feasible set $\mathcal{X}$ is a submanifold of $\mathbb{E}^n$. The gradient projection method of Bertsekas projects the point $x_k - s_k \operatorname{grad} f(x_k)$ (once the step size of either α_k or s_k is set) to the feasible set.

While the gradient projection method does not involve a space during the line search the candidate step sizes are applied to a particular direction vector $-\operatorname{grad} f(x_k)$. In general this direction vector is not tangential to the boundary of $\mathcal{X}$. The update formula (8.4.3) is different from (8.4.2) since the direction vectors $-\operatorname{grad} f(x_k)$ and $-P_{x_k} \operatorname{grad} f(x_k)$ are generically different.

The way in which the candidate iterates are pulled back to $\mathcal{X}$ is generically different. The gradient projection method uses projection $[\cdot]^+$ which projects to the nearest boundary point while Riemannian optimization uses one of many possible retractions from the tangent space. The projection $[\cdot]^+$ is a unique operation which is not always well-defined for non-convex sets. Its main characteristic is that it finds the nearest point and the residual vector $x_{k+1} - (x_k - s_k \operatorname{grad} f(x_k))$ is perpendicular to an hyperplane that is tangent to the boundary of $\mathcal{X}$ at x_{k+1}. The direction vector generically is neither perpendicular to this hyperplane nor tangent to the boundary at x_k. Additionally, when the projection is well-defined, it may still be expensive. In practice, the gradient projection method is usually applied to simple feasible sets, e.g., bounds constraints. As discussed in Section 8.4.1, for Riemannian embedded manifolds it is usually possible to choose an efficient and effective retraction. In practical terms, the lack of tangency in the gradient projection method's search direction yields a inappropriate scaling of the step size that often causes line search procedures that require more time than the corresponding Riemannian line search procedure and result in smaller step sizes which in turn slow convergence.

Selvan et. al. [SAGQ12] discuss the consequences of two updates (8.4.2) and (8.4.3) for the oblique manifold when performing independent component analysis. The effects described above were clearly observed. The search direction $-\operatorname{grad} f(x_k)$ in the gradient projection method suffers from bad performance in the sense that the movement of x_{k+1} may be arbitrary small even though a step size is chosen to be infinite while the search direction $-P_{x_k} \operatorname{grad} f(x_k)$ in Riemannian optimization does not suffer from this problem since it is the scale appropriate for the constraints.

To improve the performance based on the idea of the gradient projection method, the second order term is considered and this gives the scaled gradient projection method [Ber03]. To derive the scaled gradient projection method, let H_k be a positive definite matrix and y is defined as $(H_k)^{-1/2}x$. The problem (8.1.1) becomes

$$\min h_k(y) = f((H^k)^{-1/2}y), \text{ subject to } y \in \mathcal{Y}_k = \{y|(H_k)^{-1/2}y \in \mathcal{X}\}.$$

Using the same update (8.4.1) for y_k, we obtain

$$y_{k+1} = y_k + \alpha_k(\bar{y}_k - y_k), \tag{8.4.4}$$

where

$$\bar{y}_k = [y_k - s_k \operatorname{grad} f(y_k)]^+, \tag{8.4.5}$$

$[y]^+$ denotes $\arg\min_{z\in\mathcal{Y}_k} \|z - y\|_2$. Given the connections between x and y

$$\begin{aligned} x = (H_k)^{-1/2}y, x_k = (H_k)^{-1/2}y_k,\\ \bar{x}_k = (H_k)^{-1/2}\bar{y}_k, \operatorname{grad} h_k(y_k) = (H_k)^{-1/2} \operatorname{grad} f(x_k) \end{aligned}$$

(8.4.4) can be written as

$$x_{k+1} = x_k + \alpha_k(\bar{x}_k - x_k), \tag{8.4.6}$$

where $\bar{x}_k$ can be derived from (8.4.5) as

$$\bar{x}_k = \arg\min_{x\in\mathcal{X}} \left(\operatorname{grad} f(x_k)^T(x - x_k) + \frac{1}{2s_k}(x - x_k)H_k(x - x_k)\right). \tag{8.4.7}$$

The update (8.4.6) defines the scaled gradient projection method. H_k is a second order approximation. When H_k is chosen to be $\operatorname{Hess} f(x_k)$, (8.4.6) is a version of constrained Newton method [Ber03]. However, if $\operatorname{Hess} f(x_k)$ is not positive definite, the solution of (8.4.7) may not exist.

Proposition 2.3.4 [Ber03] shows that for the scaled gradient projection method, if α_k is chosen by the limited minimization rule or by the Armijo rule along the feasible direction and there exist positive c_1 and c_2 such that $c_1\|z\|^2 \leq z^T H_k z \leq c_2\|z\|^2$, then every limit point of $\{x_k\}$ is stationary. Proposition 2.3.5 [Ber03] proves that for the scaled gradient projection method, if f is C^2, H_k is chosen to be $\operatorname{Hess} f(x_k)$, $\operatorname{Hess} f(x)$ is positive definite for all $x \in \mathcal{X}$, let x_* be a local minimum of f over $\mathcal{X}$, there exists a $\delta > 0$ such that if $\|x_0 - x_*\| < \delta$, then $\{x_k\}$ generated by (8.4.6) with $\alpha_k = s_k = 1$ for all k satisfies $\|x_k - x_*\| < \delta$ for all k and $x_k \to x_*$. Furthermore, $\|x_k - x_*\|$ converges to zero superlinearly.

The update (8.4.6) shows the scaled gradient projection method is a feasible direction method. Therefore, it is not suitable for Riemannian optimization generally since a manifold is not necessarily a convex set and a feasible direction may not exist. If α_k is chosen to be 1, then the line search step is skipped and the method turns a nonlinear constrained optimization problem to be a sequence of constrained quadratic problems (8.4.7). This idea sounds like a trust region method but a trust region method builds a local quadratic model while the constrained quadratic problem (8.4.7) is not necessarily local. More work has to be done to solve the constrained quadratic problem. When a Riemannian optimization method makes use of a second order information, the Hessian is defined on the tangent space. However, the scaled gradient projection method considers the Hessian on the embedding space $\mathbb{E}^n$.

8.4.3 Bounds Constraints

Kelley [Kel99] discusses a gradient projection method when the feasible set is determined by bounds constraints, i.e., $\mathcal{X} = \{x \in \mathbb{R}^n | L_i \leq (x)_i \leq U_i\}$. This feasible set is convex and the line search algorithms of Bertsekas and their convergence analyses are applicable. However, Kelley defines a different line search algorithm.

Definition 8.4.4 (Line search algorithm [Kel99, p. 91])**.** *Fixed scalars $\beta \in (0,1)$ and $\sigma \in (0,1)$ are chosen, and $\alpha_k = \beta^{m_k}$, where m_k is least integer m for which*

$$f([x_k - \beta^m \operatorname{grad} f(x_k)]^+) - f(x_k) \leq \frac{-\sigma}{\beta^m}\|x - [x_k - \beta^m \bar{s} \operatorname{grad} f(x_k)]^+\|_2^2.$$

Kelley proves the existence of a step size of the line search algorithm [Kel99, Theorem 5.4.5] and also proves that if $\operatorname{grad} f(x)$ is Lipschitz continuous with Lipschitz constant L, then every limit point of the sequence $\{x_k\}$ generated by the gradient projection method is a stationary point [Kel99, Theorem 5.4.6].

For bounds constraints, Kelley discusses a variation of gradient projection methods using or approximating second order information. The variation of the gradient projection method is called two-metric projection method [Ber03] [1]. The update is

$$x_{k+1} = [x_k - s_k D_k \operatorname{grad} f(x_k)]^+, \tag{8.4.8}$$

where D_k is a symmetric positive definite not necessarily diagonal matrix.

An arbitrary positive definite symmetric D_k does not guarantee descent. Nevertheless, there is a class of matrices for D_k such that the update (8.4.8) is descent. Let ϵ-active set $A_\epsilon(x)$ at x be

$$\{i | L_i \leq x(i) \leq L_i + \epsilon \text{ or } U_i - \epsilon \leq x(i) \leq U_i\}.$$

A symmetric matrix $D \in \mathbb{R}^{n \times n}$ with element d_{ij} is diagonal with respect to a subset of indices $I \subset \{1, \dots, n\}$, if

$$d_{ij} = 0, \forall i \in I, j = 1, \dots, n, j \neq i.$$

It can be shown if D_k is symmetric positive definite and diagonal with respect to the indices $A_\epsilon(x_k)$, then the descent direction exists unless x_k is a stationary point [Kel99, Lemma 5.5.1]. The flexibility of D_k allows it to be chosen to contain the second order information and increase the rate of convergence as discussed below.

The projected Newton method for bounds constraints [Ber82] [Kel99, §5.5.2] uses the Hessian. Let ϵ-inactive set $I_\epsilon(x)$ be the complement of $A_\epsilon(x)$. If S is a set of indices, define

$$(P_S(x))_i = \begin{cases} x(i), & i \in S; \\ 0, & i \notin S. \end{cases}$$

The projected Newton method chooses

$$\begin{aligned} D_k &= P_{A_\epsilon(x_k)} + P_{I_\epsilon(x_k)} \operatorname{Hess} f(x_k) P_{I_\epsilon(x_k)} \\ &= \begin{cases} \delta_{ij}, & \text{if } i \in A_\epsilon(x_k) \text{ or } j \in A_\epsilon(x_k); \\ (\operatorname{Hess} f(x_k))_{ij}, & \text{otherwise.} \end{cases} \end{aligned} \tag{8.4.9}$$

[1] Kelley calls this method the scaled gradient projection algorithm. However, it is not consistent with the name used in Bertsekas's book [Ber03]. To be consistent, we use the names in [Ber03].

and obtains quadratic convergence [Kel99, Theorem 5.5.3].

The projected BFGS-Armijo algorithm [Kel99, §5.5.3] uses the BFGS update to approximate the Hessian. The D_k is given by adapting the inverse Hessian update,

$$D_{k+1} = P_{A_\epsilon(x_k)} + \left[\left(I - \frac{s_k y_k^T}{y_k^T s_k}\right) P_{I_\epsilon(x_k)} D_k P_{I_\epsilon(x_k)} \left(I - \frac{y_k s_k^T}{y_k^T s_k}\right) + \frac{s_k s_k^T}{y_k^T s_k}\right], \tag{8.4.10}$$

where $y_k = P_{I_\epsilon(x_k)}(\operatorname{grad} f(x_{k+1}) - \operatorname{grad} f(x_k))$ and $s_k = P_{I_\epsilon(x_k)}(x_{k+1} - x_k)$. The method is superlinearly convergent [Kel99, Theorem 5.5.4].

The boundaries of bounds constraints feasible set are hyperplanes and the ϵ-active set $A_\epsilon(x_k)$ indicates the hyperplane to which x_k is close. The hyperplane is

$$\{x \in \mathbb{R}^n | x(i) = L_i \text{ if } L_i \leq x_k(i) \leq L_i + \epsilon \text{ and } x(j) = U_j \text{ if } U_i - \epsilon \leq x_k(j) \leq U_i\}. \tag{8.4.11}$$

A hyperplane in $\mathbb{R}^n$ can be viewed as a Riemannian manifold with metric endowed from $\mathbb{E}^n$. The projection onto the hyperplane is $[y]^+ = P_{I_\epsilon(x_k)}(y) + P_{A_\epsilon(x_k)}(x_k)$ and the projection onto the tangent space is $P_{x_k}(v) = P_{I_\epsilon(x_k)}(v)$. Therefore, the Riemannian gradient at x_k is $P_{I_\epsilon(x_k)}(\operatorname{grad} f(x_k))$ and the Riemannian Hessian at x_k is

$$J_{P_{I_\epsilon(x)}(\operatorname{grad} f(x))}(x_k) = P_{I_\epsilon(x_k)} \operatorname{Hess} f(x_k) P_{I_\epsilon(x_k)} \tag{8.4.12}$$

where $J_{t(x)}$ denotes the Jacobian of function $t(x)$. We can see that the Riemannian Hessian is the same as the second component proposed in (8.4.9). In the Riemannian setting, iterates are not allowed to move outside the manifold, but iterates may move away from the hyperplane (8.4.11) and still stay feasible for bound constraints. Therefore, the first component in (8.4.9) is kept which does not use any second order information. Using the same idea, the Riemannian BFGS update applied for the hyperplane is identical to the second component of (8.4.10).

In [Kel99, Theorem 5.5.3], Kelley shows that eventually the ϵ-active set does not change and proves, from a Euclidean point of view, the convergence rate under this constant ϵ-active set assumption. In fact, if the ϵ-active set does not change in the two-metric gradient projection algorithm, the Riemannian convergence analysis directly applies. From this observation, it is clear that these methods using first or second order information are Euclidean/Riemannian hybrids that use the fixed global endowed Euclidean metric. The flexibility of the Riemannian approach can subsume and improve these methods.

CHAPTER 9

GENERAL IMPLEMENTATION TECHNIQUES

9.1 Introduction

A d-dimensional manifold $\mathcal{M}$ often has elements that can be represented by a vector in $\mathbb{R}^n$. There are some common situations where this is encountered in practice:

1. $\mathcal{M}$ is embedded in $\mathbb{R}^n$ and inherits its metric from a metric on $\mathbb{R}^n$.
2. $\mathcal{M}$ is a subset of $\mathbb{R}^n$ with a metric g_x on $\mathrm{T}_x\,\mathcal{M}$ that is not necessarily a restriction of a metric on $\mathbb{R}^n$ nor can it necessarily be extended to be a metric on all of $\mathbb{R}^n$.
3. $\mathcal{M}$ is a quotient of a manifold $\bar{\mathcal{M}}$, (which can be of either of the first two types).
4. $\mathcal{M}$ is a product of two or more manifolds (each of which can be any of the first three types).

In these cases, $\mathrm{T}_x\,\mathcal{M}$, the tangent space of x, can be identified with a subspace of $\mathbb{R}^n$ with dimension d. As a result, one key implementation choice is the use of an n-dimensional or a d-dimensional vector to represent a tangent vector. The main difference is the need for a basis of $\mathrm{T}_x\,\mathcal{M}$. More specifically, if $\xi \in \mathrm{T}_x\,\mathcal{M}$ is a n-dimensional vector a basis is not required. The associated computations on the tangent space typically exploit some characterization of vectors in $\mathbb{R}^n$ that are tangent vectors at a point $x \in \mathcal{M}$. If d is not significantly smaller than n the complexity is often not that much more than working in d dimensions explicitly.

A d-dimensional representation, u, requires that u satisfies $\xi = B_x u$ where the columns of $B_x \in \mathbb{R}^{n\times d}$ are a basis of $\mathrm{T}_x\,\mathcal{M}$. Given B_x, working in d dimensions directly tends to reduce the complexity of working with linear transformations, e.g., updating a quasi-Newton Hessian approximation is inexpensive. Unfortunately, retraction requires lifting the d-dimensional representation to an n-dimensional representation which requires extra work.

The key consideration in assessing the computational viability of using a d-dimensional representation on a particular manifold is the cost of producing and updating B_x as the optimization iteration proceeds. For example, if a function that builds basis is smooth (at least locally), i.e.,

$B : x \to B_x$ is smooth with respect to x, then the d-dimensional representation often leads to the most efficient implementations.

This chapter is organized as follows. Sections 9.2, 9.3 and 9.4 present the implementations based on an n-dimensional representation for the situations 1 – 4. Section 9.5 concentrates on the d-dimensional intrinsic representation for situation 2 since adapting to the other situations is straightforward.

9.2 A Manifold in $\mathbb{R}^n$

In this section, we consider a manifold $\mathcal{M} \subset \mathbb{R}^n$ where the metric g of $\mathcal{M}$ is not necessarily an inherited Euclidean metric, i.e., situation 2. Implementations for situation 1 can be derived from the results here and are presented for specific manifolds in Chapter 10.

9.2.1 Basic Properties of the Metric as a Matrix

Since $\mathcal{M}$ is a subset of $\mathbb{R}^n$, there exists a function $\tilde{B} : \mathcal{M} \to \mathbb{R}^{n \times d} : x \mapsto \tilde{B}_x$ such that the $\tilde{B}_x$ is a basis of $\mathrm{T}_x \mathcal{M}$. From the discussion in [AMS08, page 37], we can choose $\tilde{B}$ to be smooth at least locally. Suppose $\tilde{B}$ is smooth in a neighborhood $\mathcal{U}$. Using the QR decomposition for $\tilde{B}_x$ and noting that QR decomposition is smooth on the set of full-rank matrices, we can obtain a smooth orthonormal basis $B_x \in \mathbb{R}^{n \times d}$ for all $x \in \mathcal{U}$, [DE99].

If $\eta, \xi \in \mathrm{T}_x \mathcal{M}$, then there exist $u, v \in \mathbb{R}^d$ such that $\eta = \tilde{B}_x u$ and $\xi = \tilde{B}_x v$. Therefore, we can define an intrinsic inner product,

$$\hat{g}(u, v) = g(\tilde{B}_x u, \tilde{B}_x v) = g(\eta, \xi). \tag{9.2.1}$$

If the matrix $\hat{G}_x \in \mathbb{R}^{d \times d}$ is defined such that

$$(\hat{G}_x)_{ij} = e_i^T \hat{G}_x e_j = \hat{g}(e_i, e_j),$$

where e_i and e_j are the ith and jth canonical basis of $\mathbb{R}^d$, then $\hat{G}_x$ is symmetric positive definite and satisfies

$$\hat{g}(u, v) = u^T \hat{G}_x v. \tag{9.2.2}$$

$\hat{G}_x$ is the unique $\mathbb{R}^{d \times d}$ matrix expression of the metric with respect to the basis $\tilde{B}_x$. If the s-mooth basis $\tilde{B}_x$ is replaced by a smooth orthonormal basis B_x, then the intrinsic dimension metric expression $\hat{G}_x$ is the identity, i.e., $\hat{G}_x = I_d$.

We seek a matrix $G_x \in \mathbb{R}^{n \times n}$ such that the metric can be written

$$g(\eta, \xi) = \eta^T G_x \xi.$$

This can be combined with (9.2.1) and (9.2.2), to obtain

$$u^T \hat{G}_x v = u^T \tilde{B}_x^T G_x \tilde{B}_x v.$$

Noting u, v are arbitrary, we have

$$\hat{G}_x = \tilde{B}_x^T G_x \tilde{B}_x. \tag{9.2.3}$$

(9.2.3) is a necessary and sufficient condition for a matrix G_x to be a matrix expression of the metric on $\mathrm{T}_x\, \mathcal{M}$. Rewriting (9.2.3) as a Kronecker product, we obtain

$$\operatorname{vec}(\hat{G}_x) = (\tilde{B}_x^T \otimes \tilde{B}_x^T) \operatorname{vec}(G_x), \tag{9.2.4}$$

where $\otimes$ denotes the Kronecker product. Using a property of the Kronecker product, $rank(\tilde{B}_x^T \otimes \tilde{B}_x^T) = rank(\tilde{B}_x) rank(\tilde{B}_x)$, and noting $\tilde{B}_x$ is full column rank, we have $\tilde{B}_x^T \otimes \tilde{B}_x^T$ has full row rank. In addition, $\tilde{B}_x^T \otimes \tilde{B}_x^T$ is a short fat matrix. Hence, (9.2.4) is an underdetermined system and there are multiple solutions G_x that satisfy (9.2.3). One can choose a G_x satisfying (9.2.3) and make G_x be a smooth function with respect to $x \in \mathcal{U}$ since $\hat{G}_x$ and $\tilde{B}_x$ are smooth. Moreover, there is a unique smooth G_x such that G_x is symmetric and the columns space of G_x is $\mathrm{T}_x\, \mathcal{M}$. To this end, let G_x be written as

$$G_x = \tilde{B}_x M \tilde{B}_x^T, \tag{9.2.5}$$

where $M \in \mathbb{R}^{d \times d}$. When plugging (9.2.5) into (9.2.3), we have

$$\hat{G}_x = \tilde{B}_x^T \tilde{B}_x M \tilde{B}_x^T \tilde{B}_x. \tag{9.2.6}$$

Since $\tilde{B}_x^T \tilde{B}_x$ is full rank, (9.2.6) has a unique solution, $M = (\tilde{B}_x^T \tilde{B}_x)^{-1} \hat{G}_x (\tilde{B}_x^T \tilde{B}_x)^{-1}$. The solution M is symmetric and nonsingular and therefore G_x is symmetric and its column space is $\mathrm{T}_x\, \mathcal{M}$ as desired.

We must also consider the inverse of G_x in the appropriate context. If we have a basis B_x then the rank-d form in (9.2.5) leads naturally to a simple form of the usual generalized or Moore-Penrose inverse $G_x^\dagger$ which for (9.2.5) is given by $G_x^\dagger = \tilde{C}_x \tilde{M}^{-1} \tilde{C}_x^T$, where $\tilde{B}_x = \tilde{C}_x \tilde{R}_x$ is a QR decomposition and $\tilde{M} = \tilde{R}M\tilde{R}^T \in \mathbb{R}^{d \times d}$. This acts like the inverse when the vectors are restricted to $\mathrm{T}_x \mathcal{M} \subseteq \mathbb{R}^n$ by removing all effects of the null space.

Form (9.2.5) is very special in the sense that it assumes that G_x is a rank-d matrix. However, we are interested in using an $n \times n$ matrix G_x that satisfies the requirements of the metric g on $\mathrm{T}_x \mathcal{M}$ but may have rank greater than d. We must therefore broaden the definition of the inverse beyond $G^\dagger$ since we must remove or ignore the effects of more than just the null space of G_x.

Let $G_x^{\mathbf{-1}}$ denote the matrix that satisfies

$$G_x G_x^{\mathbf{-1}} \tilde{B}_x = G_x^{\mathbf{-1}} G_x \tilde{B}_x = \tilde{B}_x.$$

and therefore acts like an inverse of G_x when restricted to acting on vectors in $\mathrm{T}_x \mathcal{M}$.

Similarly, let $G_x^{\mathbf{1/2}}$ denote a matrix such that

$$\tilde{B}_x^T G_x^{\mathbf{1/2}} G_x^{\mathbf{1/2}} \tilde{B}_x = \tilde{B}_x^T G_x \tilde{B}_x \text{ and } \tilde{B}_x^T G_x^{\mathbf{1/2}} \tilde{B}_x = \tilde{B}_x^T (G_x^{\mathbf{1/2}})^T \tilde{B}_x, \tag{9.2.7}$$

e.g., $G_x^{\mathbf{1/2}} = \tilde{C}_x \tilde{M}^{\mathbf{1/2}} \tilde{C}_x^T$. Let $G_x^{\mathbf{-1/2}}$ denote a matrix such that

$$G_x^{\mathbf{-1/2}} G_x^{\mathbf{1/2}} \tilde{B}_x = G_x^{\mathbf{1/2}} G_x^{\mathbf{-1/2}} \tilde{B}_x = \tilde{B}_x, \tag{9.2.8}$$

e.g., $G_x^{\mathbf{-1/2}} = \tilde{C}_x \tilde{M}^{\mathbf{-1/2}} \tilde{C}_x^T$.

Finally, we let $\overset{\mathrm{T}_x \mathcal{M}}{=\!=\!=}$ denote two operators are equivalent when restricted the act on $\mathrm{T}_x \mathcal{M}$, e.g., we have

$$G_x G_x^{\mathbf{-1}} \overset{\mathrm{T}_x \mathcal{M}}{=\!=\!=} G_x^{\mathbf{-1}} G_x \overset{\mathrm{T}_x \mathcal{M}}{=\!=\!=} I_n,$$
$$G_x^{\mathbf{-1/2}} G_x^{\mathbf{1/2}} \overset{\mathrm{T}_x \mathcal{M}}{=\!=\!=} G_x^{\mathbf{1/2}} G_x^{\mathbf{-1/2}} \overset{\mathrm{T}_x \mathcal{M}}{=\!=\!=} I_n.$$

9.2.2 Operations Using n Dimensional Representation

Based on the discussion above, we have the metric defined in terms of an n-dimensional representation

$$g(\eta_x, \xi_x) = \eta_x^T G_x \xi_x, \tag{9.2.9}$$

where $\eta_x, \xi_x \in \mathrm{T}_x \mathcal{M}$ are column vectors in $\mathbb{R}^n$ and $G_x \in \mathbb{R}^{n \times n}$ is a matrix expression of the metric. $\eta_x^\flat$ is $\eta_x^T G_x$ such that $\eta_x^\flat \xi_x$ is $\eta_x^T G_x \xi_x$.

The adjoint $\mathcal{A}_x^*$ of a linear operator $\mathcal{A}_x$ satisfies, abstractly, i.e., independently of representation,

$$g(\eta_x, \mathcal{A}_x \xi_x) = g(\mathcal{A}_x^* \eta_x, \xi_x),$$

which is, when using the n-dimensional representation of the tangent vectors and operators,

$$\eta_x^T G_x \mathcal{A}_x \xi_x = \eta_x^T (\mathcal{A}_x^*)^T G_x \xi_x, \tag{9.2.10}$$

for all $\eta_x, \xi_x \in \mathrm{T}_x \mathcal{M}$. If $\mathcal{A}_x^* = \mathcal{A}_x$, then $\mathcal{A}_x$ is called self-adjoint. Lemma 9.2.1 gives necessary and sufficient conditions for $\mathcal{A}_x$ to be self-adjoint.

Lemma 9.2.1. *A d-dimensional manifold $\mathcal{M}$ is a subset of $\mathbb{R}^n$. Then a linear operator $\mathcal{A}_x$ on $\mathrm{T}_x \mathcal{M}$ is self-adjoint if and only if $B_x^\flat \mathcal{A}_x B_x = B_x^T G_x \mathcal{A}_x B_x$ is symmetric, where $G_x \in \mathbb{R}^{n \times n}$ denotes a matrix expression of the metric in $\mathrm{T}_x \mathcal{M}$, $B_x \in \mathbb{R}^{n \times d}$ denotes an orthonormal basis of $\mathrm{T}_x \mathcal{M}$ and $B_x^\flat \in \mathbb{R}^{d \times n}$ is the matrix $B_x^T G_x$.*

Proof. First, suppose $\mathcal{A}_x$ is self-adjoint. For any $\eta_x, \xi_x \in \mathrm{T}_x \mathcal{M}$, there exist d-dimensional vectors u, v such that $\eta_x = B_x u$ and $\xi_x = B_x v$. Therefore, (9.2.10) implies

$$u^T B_x^T G_x \mathcal{A}_x B_x v = u^T B_x^T \mathcal{A}_x^T G_x B_x v. \tag{9.2.11}$$

Since (9.2.10) holds for all η_x and ξ_x, (9.2.11) holds for arbitrary u and v. Therefore, we obtain

$$B_x^T G_x \mathcal{A}_x B_x = B_x^T \mathcal{A}_x^T G_x B_x,$$

which means $B_x^T G_x \mathcal{A}_x B_x$ is symmetric.

Conversely, if $B_x^T G_x \mathcal{A}_x B_x = B_x^T \mathcal{A}_x^T G_x B_x$, then by reversing the steps $\mathcal{A}_x$ is shown to be self-adjoint. □

A vector transport, $\mathcal{T}$, can be represented as an $n \times n$ matrix and we are particularly interested in the representation of isometric vector transports. Lemma 9.2.2 gives the necessary and sufficient conditions.

Lemma 9.2.2. *A d-dimensional manifold $\mathcal{M}$ is a subset of $\mathbb{R}^n$. Then a vector transport $\mathcal{T} \in \mathbb{R}^{n \times n} : \mathrm{T}_x \mathcal{M} \to \mathrm{T}_y \mathcal{M}$ is isometric if and only if $B_y^\flat \mathcal{T}_{\eta_x} B_x = B_y^T G_y \mathcal{T}_{\eta_x} B_x$ is an orthonormal matrix, where $G_x, G_y \in \mathbb{R}^{n \times n}$ denote a matrix expression of the metric in $\mathrm{T}_x \mathcal{M}, \mathrm{T}_y \mathcal{M}$ respectively, $B_x, B_y \in \mathbb{R}^{n \times d}$ denote an orthonormal basis of $\mathrm{T}_x \mathcal{M}, \mathrm{T}_y \mathcal{M}$ respectively.*

Proof. If $\mathcal{T}$ is isometric, then from its definition (1.2.1), we have

$$\xi_x^T \mathcal{T}_{\eta_x}^T G_y \mathcal{T}_{\eta_x} \zeta_x = \xi_x^T G_x \zeta_x,$$

for all $\xi_x, \zeta_x \in \mathrm{T}_x \mathcal{M}$. Noting that there exist $u, v \in \mathbb{R}^d$ such that $\xi_x = B_x u$ and $\zeta_x = B_x v$, we have

$$u^T B_x^T \mathcal{T}_{\eta_x}^T G_y \mathcal{T}_{\eta_x} B_x v = u^T B_x^T G_x B_x v,$$

for all $u, v \in \mathbb{R}^d$. Therefore, it follows that

$$B_x^T \mathcal{T}_{\eta_x}^T G_y \mathcal{T}_{\eta_x} B_x = B_x^T G_x B_x.$$

Since B_x is an orthonormal basis with respect to the metric G_x, we have $B_x^T G_x B_x = B_x^\flat B_x = I_d$. Hence, we know

$$B_x^T \mathcal{T}_{\eta_x}^T G_y \mathcal{T}_{\eta_x} B_x = I_d, \tag{9.2.12}$$

which shows that $\mathcal{T}_{\eta_x} B_x$ is also an orthonormal basis of $\mathrm{T}_y \mathcal{M}$. Therefore, there exists an orthonormal matrix $O \in \mathbb{R}^{d \times d}$ such that $\mathcal{T}_{\eta_x} B_x = B_y O$. Substituting this into (9.2.12), we have

$$O^T B_y^T G_y \mathcal{T}_{\eta_x} B_x = I_d.$$

Therefore, $B_y^T G_y \mathcal{T}_{\eta_x} B_x = O$ is an orthonormal matrix.

Conversely, suppose $B_y^T G_y \mathcal{T}_{\eta_x} B_x$ is an orthonormal matrix. Note that $\mathcal{T}_{\eta_x} B_x \in \mathrm{T}_y \mathcal{M}$. There exists a matrix $O \in \mathbb{R}^{d \times d}$ such that $\mathcal{T}_{\eta_x} B_x = B_y O$. Therefore, it follows that

$$B_y^T G_y \mathcal{T}_{\eta_x} B_x = B_y^T G_y B_y O. \tag{9.2.13}$$

Since B_y is an orthonormal basis with respect to the metric G_y, we have $B_y^\flat B_y = B_y^T G_y B_y = I_d$. Therefore, by (9.2.13) and assumption, we obtain O is orthonormal. Therefore, we know

$$B_x^T \mathcal{T}_{\eta_x}^T G_y \mathcal{T}_{\eta_x} B_x = O^T O = I_d,$$

which is the same as (9.2.12). By reversing the previous steps, we obtain $\mathcal{T}$ is isometric. □

9.2.3 Construction of Isometric Vector Transports

A vector transport is a map from one tangent space to another tangent space. In our framework, it can be represented by an n by n matrix. Qi [Qi11] discussed what kinds of functions give vector transports. She gives the following definition.

Definition 9.2.1. *A* subspace matching function *is a smooth (partial) function*

$$\ell : \mathrm{Gr}(d, n) \times \mathrm{Gr}(d, n) \to L(\mathbb{R}^n, \mathbb{R}^n),$$

where $\mathrm{Gr}(d, n)$ *is Grassmann manifold, i.e., all d-dimensional subspaces of* $\mathbb{R}^n$*,* $L(\mathbb{R}^n, \mathbb{R}^n)$ *denotes the set of all linear maps from* $\mathbb{R}^n$ *into itself, with the following conditions:*

1. *The domain of ℓ, denoted by* $\mathrm{dom}(\ell)$*, contains a neighborhood of the diagonal* $\Delta_{\mathrm{Gr}(d,n)} = \{(\mathcal{X}, \mathcal{X}) : \mathcal{X} \in \mathrm{Gr}(d,n)\}$.
2. $\ell(\mathcal{X}, \mathcal{Y})\mathcal{X} \subseteq \mathcal{Y}$
3. $\ell(\mathcal{X}, \mathcal{Y})\mathcal{X}_\perp = \{0\}$
4. *(Consistency)* $\ell(\mathcal{X}, \mathcal{X})|_{\mathcal{X}} = \mathrm{id}_{\mathcal{X}}$, *for all* $\mathcal{X} \in \mathrm{Gr}(d,n)$.

If moreover $\ell(\mathcal{X}, \mathcal{Y})|_{\mathcal{X}}$ *is an isometry for all* $(\mathcal{X}, \mathcal{Y}) \in \mathrm{dom}(\ell)$*, where the metric is the one induced from the canonical metric in* $\mathbb{R}^n$*, then we say that ℓ is* isometric*. We say that ℓ is* isotropic *if*

$$\ell(U\mathcal{X}, U\mathcal{Y}) = U\ell(\mathcal{X}, \mathcal{Y})U^T$$

for all $U \in \mathrm{O}_n$*; in this case, ℓ is fully determined by specifying* $\ell(\mathrm{col}(I_{n,d}), \mathcal{Y})$ *for all* $\mathcal{Y} \in \mathrm{Gr}(d,n)$.

Qi proves that $\mathcal{T}$ defined by $\mathcal{T}_{\eta_x}\xi_x = \ell(\mathrm{T}_x \mathcal{M}, \mathrm{T}_{R(\eta_x)} \mathcal{M})\xi_x$ is a vector transport. Both the RBroyden family and RTR-SR1 require the vector transport to be isometric. In this section, methods of constructing isometric vector transports when $\mathcal{M}$ is a subset of $\mathbb{R}^n$ are discussed.

Since given two points x and y in $\mathcal{M}$, two tangent spaces $\mathrm{T}_x \mathcal{M}$ and $\mathrm{T}_y \mathcal{M}$ are d-dimensional subspaces of $\mathbb{R}^n$, the first choice for the isometric vector transport $\mathcal{T}$ from x to y is the direct rotation [DK70, Def. 3.1] from $\mathrm{T}_x \mathcal{M}$ to $\mathrm{T}_y \mathcal{M}$, restricted to act on $\mathrm{T}_x \mathcal{M}$. The direct rotation exists and is unique in the acute case, i.e., when all canonical angles between the subspaces are acute, which is assumed throughout. The acute case is guaranteed if x is sufficiently close to y.

We consider the implementation of the direct rotation vector transport for case where the matrix G_z of $g_z(\cdot, \cdot)$ is identity for $z \in \mathcal{M}$. Let B_x and B_y be orthonormal bases of $\mathrm{T}_x \mathcal{M}$ and $\mathrm{T}_y \mathcal{M}$ respectively. Hence B_x and B_y can be viewed as n by d matrices and $B_x^T B_x = B_y^T B_y = I_d$. The direct rotation vector transport from x to y is given by

$$\mathcal{T} = B_y U_b^T B_x^T, \tag{9.2.14}$$

where $B_x^T B_y = U_b P_b$ is the unique polar decomposition; this can be deduced, e.g., from the considerations in [QZL05, §2].

Since the matrix expression, G_z, of the metric $g_z(\cdot, \cdot)$ is the identity when restricted to action on $\mathrm{T}_z \mathcal{M}$ for $z \in \mathcal{M}$, we can extend the G_z and define it to be I_n. We then can extent the definition of a normal space for an embedded manifold (situation 1) to situation 2 considered in this section, i.e., $\mathrm{N}_z \mathcal{M}$ at $z \in \mathcal{M}$ is defined to be $\{v \in \mathbb{R}^n | g_z(v, \eta_z) = v^T \eta_z = 0, \text{for all } \eta_z \in \mathrm{T}_z \mathcal{M}\}$. If the codimension, $n - d$, is sufficiently smaller than the dimension, d, and if, moreover, an orthonormal basis N_z of the normal space $\mathrm{N}_z\mathcal{M}$, $z \in \mathcal{M}$, is readily available, then the following reformulation of (9.2.14) becomes computationally advantageous. Observe that any $\eta \in \mathrm{T}_x \mathcal{M}$ can be decomposed into a component η_1 in $\mathrm{T}_x \mathcal{M} \cap \mathrm{T}_y \mathcal{M}$ and η_2 in $\mathrm{T}_x \mathcal{M} \ominus (\mathrm{T}_x \mathcal{M} \cap \mathrm{T}_y \mathcal{M}) := \mathrm{T}_x \mathcal{M} \cap (\mathrm{T}_x \mathcal{M} \cap \mathrm{T}_y \mathcal{M})_\perp$. It can be deduced from [QZL05, §2] that the direct rotation from $\mathrm{T}_x \mathcal{M}$ to $\mathrm{T}_y \mathcal{M}$ applied to $\eta \in \mathrm{T}_x \mathcal{M}$ amounts to keeping η_1 invariant and applying to η_2 the direct rotation from $\mathrm{T}_x \mathcal{M} \ominus (\mathrm{T}_x \mathcal{M} \cap \mathrm{T}_y \mathcal{M})$ to $\mathrm{T}_y \mathcal{M} \ominus (\mathrm{T}_x \mathcal{M} \cap \mathrm{T}_y \mathcal{M})$. Since $(\mathrm{T}_x \mathcal{M} \cap \mathrm{T}_y \mathcal{M})_\perp = \mathrm{N}_x\mathcal{M} \oplus \mathrm{N}_y\mathcal{M}$ and $\mathrm{T}_x \mathcal{M} = (\mathrm{N}_x\mathcal{M})_\perp$, one finds that a spanning set of their intersection $\mathrm{T}_x \mathcal{M} \ominus (\mathrm{T}_x \mathcal{M} \cap \mathrm{T}_y \mathcal{M})$ is given by $(I - N_x N_x^T) \begin{bmatrix} N_x & N_y \end{bmatrix}$, i.e., by $(I - N_x N_x^T) N_y$. Hence, an orthonormal basis Q_x of $\mathrm{T}_x \mathcal{M} \ominus (\mathrm{T}_x \mathcal{M} \cap \mathrm{T}_y \mathcal{M})$ is obtained by orthonormalizing $(I - N_x N_x^T) N_y$, and likewise with x and y interchanged. Consequently, the direct rotation vector transport $\mathcal{T}$ (9.2.14) is equivalent to

$$\mathcal{T} = (I_n - Q_x Q_x^T) + Q_y U_q^T Q_x^T, \tag{9.2.15}$$

where $Q_x^T Q_y = U_q P_q$ is the unique polar decomposition. Of course, the implementation of the $(I_n - Q_x Q_x^T)$ must be considered carefully from the numerical point of view, i.e., sum of products vs product of sums, depending on how orthogonal the elements of the basis are guaranteed to be.

Lemma 9.2.3. *The vector transports* (9.2.14) *and* (9.2.15) *are equivalent when $B_x^T B_y$ is invertible.*

Proof. Suppose the dimension of $\mathrm{T}_x \mathcal{M} \cap \mathrm{T}_y \mathcal{M}$ is d_c. Since $\mathrm{T}_x \mathcal{M}$ can be decomposed into two perpendicular spaces $\mathrm{T}_x \mathcal{M} \cap \mathrm{T}_y \mathcal{M}$ and $\mathrm{T}_x \mathcal{M} \ominus (\mathrm{T}_x \mathcal{M} \cap \mathrm{T}_y \mathcal{M})$, similarly for $\mathrm{T}_y \mathcal{M}$, we know there exist orthonormal matrices O_x and O_y such that the first d_c columns of $\tilde{B}_x = B_x O_x$ and $\tilde{B}_y = B_y O_y$ are an orthonormal basis $\tilde{B}_{d_c}$ of $\mathrm{T}_x \mathcal{M} \cap \mathrm{T}_y \mathcal{M}$ and the last $d - d_c$ columns of $\tilde{B}_x$ and $\tilde{B}_y$ are Q_x and Q_y respectively.

Note the structures of $\tilde{B}_x$ and $\tilde{B}_y$, we have

$$\tilde{B}_x^T \tilde{B}_y = \begin{pmatrix} I_{d_c} & 0 \\ 0 & Q_x^T Q_y \end{pmatrix}.$$

Therefore, the polar decomposition gives

$$\tilde{B}_x^T \tilde{B}_y = \begin{pmatrix} I_{d_c} & 0 \\ 0 & U_q \end{pmatrix} \begin{pmatrix} I_{d_c} & 0 \\ 0 & P_q \end{pmatrix}$$

In addition, we also have an another expression of the polar decomposition,

$$\tilde{B}_x^T \tilde{B}_y = O_x^T U_b P_b O_y = O_x^T U_b O_y O_y^T P_b O_y.$$

$\tilde{B}_x^T \tilde{B}_y$ is invertible since $B_x^T B_y$ is invertible. Therefore, the polar decomposition is unique. We obtain

$$O_x^T U_b O_y = \begin{pmatrix} I_{d_c} & 0 \\ 0 & U_q \end{pmatrix}$$

The vector transport (9.2.14) can be formulated into

$$\begin{aligned} \mathcal{T} &= B_y U_b B_x^T = \tilde{B}_y O_y^T U_b O_x \tilde{B}_x^T = \tilde{B}_y \begin{pmatrix} I_{d_c} & 0 \\ 0 & U_q \end{pmatrix} \tilde{B}_x^T \\ &= \tilde{B}_{d_c} \tilde{B}_{d_c}^T + Q_y U_q^T Q_x^T, \end{aligned}$$

which is equivalent to $(I_n - Q_x Q_x^T) + Q_y U_q^T Q_x^T$. □

G_z restricted to $\mathrm{T}_z \mathcal{M}$ is not always identity. If not, we can choose a smooth function $G : x \mapsto G_x$ such that G_x satisfies (9.2.3). $G^{\mathbf{1/2}} : x \mapsto G_x^{\mathbf{1/2}}$ denotes a smooth function such that $G^{\mathbf{1/2}}$ satisfies (9.2.7). $G^{-\mathbf{1/2}} : x \mapsto G_x^{-\mathbf{1/2}}$ denotes a smooth function such that $G^{-\mathbf{1/2}}$ satisfies (9.2.8). Let B_x and B_y, be orthonormal bases of $\mathrm{T}_x \mathcal{M}$ and $\mathrm{T}_y \mathcal{M}$ respectively. Hence B_x and B_y satisfy $B_x^\flat B_x = B_x^T G_x B_x = I_d$ and $B_y^\flat B_y = B_y^T G_y B_y = I_d$. We can make substitutions $\tilde{B}_x = G_x^{\mathbf{1/2}} B_x$ and $\tilde{B}_y = G_y^{\mathbf{1/2}} B_y$. The direct rotation vector transport (9.2.14) is

$$\mathcal{T} = \tilde{B}_y U_b^T \tilde{B}_x^T,$$

where $\tilde{B}_x^T \tilde{B}_y = B_x^T G_x^{\mathbf{1/2}} G_y^{\mathbf{1/2}} B_y = U_b P_b$ is the unique polar decomposition. This is a vector transport under $\tilde{B}_x$ and $\tilde{B}_y$. In order to make it consistent with the original basis B_x and B_y, we obtain a vector transport

$$\begin{aligned} \mathcal{T} &= G_y^{-\mathbf{1/2}} \tilde{B}_y U_b^T \tilde{B}_x^T G_x^{\mathbf{1/2}}, \\ &\xlongequal{\mathrm{T}_x \mathcal{M}} B_y U_b^T B_x^\flat \end{aligned} \tag{9.2.16}$$

where $B_x^T G_x^{\mathbf{1/2}} G_y^{\mathbf{1/2}} B_y = U_b P_b$ is the unique polar decomposition. The operator (9.2.16) is called a vector transport by direct rotation based on the tangent space.

As when G_z was the identity, if G_z is a symmetric positive definite matrix, then we can broaden the definition of a normal space, i.e., $\mathrm{N}_z \mathcal{M}$ at $z \in \mathcal{M}$ is defined to be $\{v \in \mathbb{R}^n | g_z(v, \eta_z) = v^T G_z \eta_z =$

$0,$ for all $\eta_z \in \mathrm{T}_z\,\mathcal{M}\}$. N_z is an orthonormal basis of $N_z\mathcal{M}$ with respect to the metric defined by G_z. Let $\tilde{N}_z$ denote $G_z^{1/2}N_z$. Note $G_z^{1/2}$ in $\tilde{N}_z$is the usual Euclidean operator definition of the matrix square root decomposition for G, that satisfies $G^{1/2}G^{1/2} = G$ and $(G^{1/2})^T = G^{1/2}$. $\tilde{N}_z$ is an orthonormal basis with respect to the metric defined by I_n. By using an idea similar to (9.2.14), we obtain a formulation of (9.2.16)

$$\mathcal{T} = G_y^{\mathbf{-1/2}}(I_n - Q_xQ_x^T + Q_yU_q^TQ_x^T)G_x^{\mathbf{1/2}}, \tag{9.2.17}$$

where $Q_x^TQ_y = U_qP_q$ is the unique polar decomposition, $Q_x = \mathrm{orth}((I_n - \tilde{N}_x\tilde{N}_x^T)\tilde{N}_y)$, likewise Q_y with x and y interchanged and $\mathrm{orth}(A)$ denotes orthonormalizing A. The operator (9.2.17) is called a vector transport by direct rotation based on normal space.

The vector transport (9.2.16) needs $G_x^{\mathbf{1/2}}$ and $G_y^{\mathbf{1/2}}$ that might be expensive. We can use an alternative form related to (9.2.16)

$$\mathcal{T} = B_yU_b^TB_x^{\flat}, \tag{9.2.18}$$

where U_b is from a polar decomposition of $B_x^{\flat}B_y = U_bP_b$ or $(B_y^{\flat}B_x)^T = P_bU_b$. We also called (9.2.18) a vector transport by direct rotation based on tangent space. The alternative (9.2.18) has computation advantages. In some cases, (9.2.18) is equivalent to (9.2.16), e.g., $G_z = I_n$, for all $z \in \mathcal{M}$. However, the vector transport (9.2.17) does not have an alternative form in general. The idea of (9.2.17) is to keep the component in $\mathrm{T}_x\,\mathcal{M} \cap \mathrm{T}_y\,\mathcal{M}$ and transport the component in $\mathrm{T}_x\,\mathcal{M}$ to $\mathrm{T}_y\,\mathcal{M}$ isometrically. However, since G_x and G_y are different in general, the component in $\mathrm{T}_x\,\mathcal{M} \cap \mathrm{T}_y\,\mathcal{M}$ with the metric G_x does not have same length as the that with the metric G_y.

The vector transports defined by (9.2.17), (9.2.18) do not require the basis functions B_x and N_x, respectively, to be smooth with respect to x. If smoothness is imposed, i.e., $B : x \to B_x$ and $N : x \to N_x$ are smooth functions to build a basis of $\mathrm{T}_x\,\mathcal{M}$ and $\mathrm{N}_x\,\mathcal{M}$, then we have simpler forms of isometric vector transports,

$$\mathcal{T} = B_yB_x^{\flat}, \tag{9.2.19}$$

$$\mathcal{T} = G_y^{\mathbf{-1/2}}(I_n - Q_xQ_x^T - Q_yQ_x^T)G_x^{\mathbf{1/2}}, \tag{9.2.20}$$

where Q_x and Q_y are the same as defined in (9.2.17). As with (9.2.17), (9.2.20) requires G_z to be a symmetric positive definite matrix. The operator (9.2.19) is called a vector transport by parallelization and (9.2.20) is called a vector transport by rigging. It is not difficult to verify

(9.2.19) is isometric and satisfies the conditions of definition 9.2.1. It is also easy to verify (9.2.20) except smoothness. When x approaches to y, N_x approaches to N_y, but the matrix Q_x **does not** approach matrix Q_y. This can be seen by considering the expressions

$$Q_x = \mathrm{orth}((I_n - \tilde{N}_x \tilde{N}_x^T)\tilde{N}_y) \text{ and } Q_y = \mathrm{orth}((I_n - \tilde{N}_y \tilde{N}_y^T)\tilde{N}_x).$$

Since $\tilde{N}_x^T \tilde{N}_y$ and $\tilde{N}_y^T \tilde{N}_x$ both approach the identity, we have that Q_y approaches $-Q_x$. This is the reason that $-Q_y Q_x^T$ appears in (9.2.20) rather than $Q_y Q_x^T$. Even though the continuity of (9.2.20) is shown, whether it is smooth is still an open question.

If $d \ll n - d$, then (9.2.18) and (9.2.19) are preferred computationally. Likewise, if $d \gg n - d$, then (9.2.17) and (9.2.20) are preferred. If d is not too different from $n - d$, then (9.2.17) and (9.2.20) are more expensive due to computation of "orth". Forms (9.2.17) and (9.2.20) potentially suffer more from numerical errors than (9.2.18) and (9.2.19). Forms (9.2.17) and (9.2.18) do not need the function that builds the bases to be smooth but (9.2.19) and (9.2.20) do. As a result, (9.2.17) and (9.2.18) are more expensive.

We have considered situation 2 in detail. Note that situation 1 is a simple special case and all of the forms above are easily adapted.

9.3 Quotient Manifold of a Manifold in $\mathbb{R}^n$

9.3.1 General Discussion

A detailed discussion of quotient manifold concepts is in [AMS08]. An element x in a quotient manifold, $\mathcal{M}$, represents an equivalent class $[\bar{x}]$ in the total space $\bar{\mathcal{M}}$, i.e.,

$$[\bar{x}] = \{\bar{y} \in \bar{\mathcal{M}} | \bar{y} \sim \bar{x}\},$$

where $\sim$ denotes an equivalence relation, i.e., a relation that is

1. reflexive: $x \sim x$ for all $x \in \bar{\mathcal{M}}$,
2. symmetric: $x \sim y$ if and only if $y \sim x$ for all $x, y \in \bar{\mathcal{M}}$,
3. transitive: if $x \sim y$ and $y \sim z$ then $x \sim z$ for all $x, y, z \in \bar{\mathcal{M}}$.

An alternative way to interpret an equivalent class $[\bar{x}]$ is to use a group $\mathfrak{G}$ and its action on $\bar{\mathcal{M}}$. This interpretation is used in this section.

Definition 9.3.1 (Group). *A group $\mathfrak{G}$ is a set having an associative binary operation, denoted by $\cdot$, such that:*

1. *there is an element e in $\mathfrak{G}$ such that $e \cdot h = h \cdot e = h$ for all $h \in \mathfrak{G}$,*
2. *for every $h \in \mathfrak{G}$, there exists a unique k such that $h \cdot k = k \cdot h = e$.*

e is called the identity element of $\mathfrak{G}$ and k is called the inverse of h, denoted by h^{-1}.

Definition 9.3.2 (Group action). *$\mathfrak{G}$ is a group and $\mathcal{X}$ is a set. A group action of $\mathfrak{G}$ on $\mathcal{X}$ is a function*

$$\mathfrak{G} \times \mathcal{X} \to \mathcal{X}, (h, x) \mapsto h \bullet x$$

that satisfies

1. *Associativity: $(h \cdot k) \bullet x = h \bullet (k \bullet x)$ and*
2. *Identity: $e \bullet x = x$ for all $x \in \mathcal{X}$.*

The equivalent class $[\bar{x}]$ therefore can be written as

$$[\bar{x}] = \{h \bullet \bar{x} | h \in \mathfrak{G}\}$$

and represents the notion of invariance under the group action, i.e., two elements that are related by an application of a group member are equivalent. The quotient space is

$$\mathcal{M} = \bar{\mathcal{M}} / \sim = \bar{\mathcal{M}} / \mathfrak{G} = \{[\bar{x}] : \bar{x} \in \bar{\mathcal{M}}\}.$$

The necessary and sufficient conditions for the quotient space to be a quotient manifold, not necessarily a Riemannian manifold, are given by [AMS08, Proposition 3.4.2]. If $\bar{\mathcal{M}}$ is a Riemannian manifold and $\mathcal{M}$ is a manifold, then a necessary and sufficient condition for $\mathcal{M}$ to have a metric endowed from $\bar{\mathcal{M}}$ is that the action of $\mathfrak{G}$ is isometry, i.e.,

$$\mathrm{dist}(\bar{x}, \bar{y}) = \mathrm{dist}(h \bullet \bar{x}, h \bullet \bar{y}),$$

where $h \in \mathfrak{G}$.

The tangent space of x in a quotient manifold is an abstract concept and is difficult to work on directly. To overcome this difficulty, first of all, the horizontal distribution, $\mathcal{H}$, is defined (see [AMS08, Section 3.5.8]). The mapping $\mathcal{H}$ assigns a subspace $\mathcal{H}_{\bar{x}}$ of $\mathrm{T}_{\bar{x}} \bar{\mathcal{M}}$ to each element

$\bar{x} \in [\bar{x}] = x$. The subspace $\mathcal{H}_{\bar{x}}$ is called the horizontal space at $\bar{x}$ and satisfies $\mathcal{H}_{\bar{x}} \oplus \mathcal{V}_{\bar{x}} = \mathrm{T}_{\bar{x}} \bar{\mathcal{M}}$. The subspace $\mathcal{V}_{\bar{x}} = \mathrm{T}_{\bar{x}}[\bar{x}]$ is called the vertical space at $\bar{x}$. Let $\pi : \bar{\mathcal{M}} \to \mathcal{M} : \bar{x} \mapsto x = [\bar{x}]$ denote the canonical projection. There exists a unique vector $\eta_{\uparrow_{\bar{x}}} \in \mathcal{H}_{\bar{x}}$ such that $\mathrm{D}\,\pi(\bar{x})[\eta_{\uparrow_{\bar{x}}}] = \eta_x$ and $\eta_{\uparrow_{\bar{x}}}$ is called a horizontal lift of η_x at $\bar{x}$. Horizontal spaces are used instead of the tangent space when implementing the algorithms discussed in this dissertation.

If $\bar{x}_1, \bar{x}_2$ are representations of a single element $x \in \mathcal{M}$, $\eta_{\uparrow_{\bar{x}_1}}$ and $\eta_{\uparrow_{\bar{x}_2}}$ are horizontal lifts of $\eta_x \in \mathrm{T}_x \mathcal{M}$ at $\bar{x}_1$ and $\bar{x}_2$, then the relationship between $\eta_{\uparrow_{\bar{x}_1}}$ and $\eta_{\uparrow_{\bar{x}_2}}$ is

$$\mathrm{D}\,\pi(\bar{x}_1)[\eta_{\uparrow_{\bar{x}_1}}] = \mathrm{D}\,\pi(\bar{x}_2)[\eta_{\uparrow_{\bar{x}_2}}]. \tag{9.3.1}$$

However (9.3.1) is general but abstract and difficult to use in practice. Theorem 9.3.1 gives a specific way to compute the relationship.

Theorem 9.3.1. *$\mathfrak{G}$ is a group and has a group action $\mathfrak{G} \times \mathcal{N} \to \mathcal{N}$ where $\mathcal{N} \subset \mathbb{R}^n$. For any $h \in \mathfrak{G}$, $h : \mathcal{N} \to \mathcal{N} : \bar{x} \mapsto h \bullet \bar{x}$ is a differentiable function.*

(i) Suppose $\mathcal{N}$ is $\mathbb{R}^n$. If $r(t)$ is a smooth curve on $\mathcal{N}$ and $r(0) = \bar{x}$, $\frac{d}{dt} r(t)|_{t=0} = \xi_{\bar{x}}$, $\xi_{h\bullet\bar{x}} = \frac{d}{dt}(h \bullet r(t))|_{t=0}$, where $h \in \mathfrak{G}$, then

$$\xi_{h\bullet\bar{x}} = J_h(\bar{x})\xi_{\bar{x}},$$

where J_h is the Jacobian of h.

(ii) Suppose $\mathcal{N}$ is a manifold, denoted by $\bar{\mathcal{M}}$, $r(t)$ is a smooth curve on $\bar{\mathcal{M}}$, $r(0) = \bar{x}$, and $\frac{d}{dt} r(t)|_{t=0} = \xi_{\bar{x}} \in \mathrm{T}_{\bar{x}} \bar{\mathcal{M}}$. Let $\xi_{h\bullet\bar{x}}$ denote the $\frac{d}{dt}(h \bullet r(t))|_{t=0}$, where $h \in \mathfrak{G}$. Then

$$\xi_{h\bullet\bar{x}} \in \mathrm{T}_{h\bullet\bar{x}} \bar{\mathcal{M}} \tag{9.3.2}$$

$$\xi_{h\bullet\bar{x}} = J_h(\bar{x})\xi_{\bar{x}}, \tag{9.3.3}$$

(iii) Suppose $\mathcal{N}$ is a manifold, denoted by $\bar{\mathcal{M}}$, and $\bar{\mathcal{M}}$ is a Riemannian manifold with a metric g that is not necessarily a Euclidean metric. If the action of $\mathfrak{G}$ is isometric with respect to the metric, then

$$\xi_{\bar{x}} G_{\bar{x}} \eta_{\bar{x}} = \xi_{\bar{x}} J_h(\bar{x})^T G_{h\bullet\bar{x}} J_h(\bar{x}) \eta_{\bar{x}}.$$

for all $\xi_{\bar{x}}, \eta_{\bar{x}} \in \mathrm{T}_{\bar{x}} \bar{\mathcal{M}}$, where $G_{\bar{x}}$ is a matrix expression of the metric at $\bar{x}$. Moreover, if $G_{\bar{x}}$ is symmetric positive definite in a neighborhood $\mathcal{U}$ of $\bar{x}$, where $\mathcal{U}$ is an open set in $\mathbb{R}^n$, and the action of $\mathfrak{G}$ is isometric in $\mathcal{U}$, then

$$G_{\bar{x}} = J_h(\bar{x})^T G_{h\bullet\bar{x}} J_h(\bar{x}), \tag{9.3.4}$$

where $h \bullet \bar{x} \in \mathcal{U}$.

(iv) Suppose $\mathcal{N}$ is a manifold, denoted by $\bar{\mathcal{M}}$, $\bar{\mathcal{M}}$ is a Riemannian manifold with a metric g which is not necessarily a Euclidean metric, and the action of $\mathfrak{G}$ is isometric with respect to the metric. If $\mathcal{M} = \bar{\mathcal{M}}/\mathfrak{G}$ is a Riemannian quotient manifold with metric endowed from $\bar{\mathcal{M}}$, then the relationship between two horizontal lifts $\eta_{\uparrow_{\bar{x}_1}}$ and $\eta_{\uparrow_{\bar{x}_2}}$ of η_x at $\bar{x}_1$ and $\bar{x}_2$ is

$$\eta_{\uparrow_{\bar{x}_2}} = J_h(\bar{x}_1)\eta_{\uparrow_{\bar{x}_1}}$$

where h satisfies $\bar{x}_2 = h \bullet \bar{x}_1$.

Proof. (i): We have

$$\xi_{h\bullet\bar{x}} = \frac{d}{dt}(h \bullet r(t))|_{t=0} = J_h(r(0))\frac{d}{dt}r(t)|_{t=0} = J_h(\bar{x})\xi_{\bar{x}}.$$

(ii): (9.3.3) is a consequence of (i). Since h is defined from $\bar{\mathcal{M}}$ to $\bar{\mathcal{M}}$, $h \bullet r(t)$ is also a curve on $\bar{\mathcal{M}}$. Therefore, $\frac{d}{dt}(h \bullet r(t))|_{t=0}$ is a tangent vector on $\mathrm{T}_{h\bullet r(0)}\,\bar{\mathcal{M}}$, which is $\xi_{h\bullet\bar{x}} \in \mathrm{T}_{h\bullet\bar{x}}\,\bar{\mathcal{M}}$.

(iii): Since h is isometric, we know it preserves the distance, i.e,

$$\int_0^s \|\frac{d}{dt}r(t)\|dt = \int_0^s \|\frac{d}{dt}(h \bullet r(t))\|dt,$$

where $r(t)$ is a smooth curve on $\bar{\mathcal{M}}$. Taking the derivative with respect to s for both sides yields

$$\|\frac{d}{dt}r(s)\| = \|\frac{d}{dt}(h \bullet r(s))\|.$$

Setting $s = 0$, we have

$$\|\frac{d}{dt}r(0)\| = \|\frac{d}{dt}(h \bullet r(0))\|.$$

which is

$$\|\zeta_{\bar{x}}\| = \|\zeta_{h\bullet\bar{x}}\|,$$

where $\zeta_{\bar{x}} = \frac{d}{dt}r(0)$ and $\zeta_{h\bullet\bar{x}} = \frac{d}{dt}(h \bullet r(t))$. Using (ii) and noting $\zeta_{\bar{x}}$ can be an arbitrary tangent vector in $\mathrm{T}_{\bar{x}}\,\bar{\mathcal{M}}$, we have

$$\|\zeta_{\bar{x}}\| = \|J_h(\bar{x})\zeta_{\bar{x}}\|,$$

for all $\zeta_{\bar{x}} \in \mathrm{T}_{\bar{x}}\,\bar{\mathcal{M}}$. Therefore, for any $\xi_{\bar{x}}, \eta_{\bar{x}} \in \mathrm{T}_{\bar{x}}\,\bar{\mathcal{M}}$, we have

$$\|\xi_{\bar{x}} + \eta_{\bar{x}}\| = \|J_h(\bar{x})(\xi_{\bar{x}} + \eta_{\bar{x}})\|,$$

which yields

$$g(\xi_{\bar{x}}, \eta_{\bar{x}}) = g(J_h(\bar{x})\xi_{\bar{x}}, J_h(\bar{x})\eta_{\bar{x}}). \tag{9.3.5}$$

Using a matrix expression for the metric, we obtain

$$\xi_{\bar{x}}^T G_{\bar{x}} \eta_{\bar{x}} = \xi_{\bar{x}}^T J_h(\bar{x})^T G_{h \bullet \bar{x}} J_h(\bar{x}) \eta_{\bar{x}}.$$

Moreover, if h is isometric in an open set of $\mathbb{R}^n$ and $\bar{x}$ is in the open set, then we can choose $r(t)$ such that $r(0) = \bar{x}$ and $\frac{d}{dt} r(t)|_{t=0}$ is an arbitrary vector in $\mathbb{R}^n$. Therefore, $\xi_{\bar{x}}$ and $\eta_{\bar{x}}$ are arbitrary vectors in $\mathbb{R}^n$. We obtain

$$G_{\bar{x}} = J_h(\bar{x})^T G_{h \bullet \bar{x}} J_h(\bar{x}).$$

(iv): Since $\eta_{\uparrow_{\bar{x}_1}}$ is in $\mathcal{H}_{\bar{x}_1}$, there exists a smooth path $r(t)$ such that $r(0) = \bar{x}_1$, $\dot{r}(0) = \eta_{\uparrow_{\bar{x}_1}}$ and $g(\eta_{\uparrow_{\bar{x}_1}}, \xi_{\bar{x}_1}) = 0$ for all $\xi_{\bar{x}_1} \in \mathrm{T}_{\bar{x}_1}[\bar{x}_1]$. Let $\zeta_{\bar{x}_2}$ denote $\frac{d}{dt}(h \bullet r(t))|_{t=0}$. By (ii), we have

$$\zeta_{\bar{x}_2} = J_h(\bar{x}_1) \eta_{\uparrow_{\bar{x}_1}}.$$

We next show $\zeta_{\bar{x}_2} \in \mathcal{H}_{\bar{x}_2}$ and $\zeta_{\bar{x}_2}$ is the horizontal lift of η_x at $\bar{x}_2$. Therefore, we can obtain $\zeta_{\bar{x}_2} = \eta_{\uparrow_{\bar{x}_2}}$.

First, for any $\xi_{\bar{x}_1} \in \mathcal{V}_{\bar{x}_1}$, there exists a path $r_v(t) \subset [\bar{x}_1]$ such that $r_v(0) = \bar{x}_1$ and $\dot{r}_v(0) = \xi_{\bar{x}_1}$. Due to the definition of $[\bar{x}_1]$, we know $h \bullet r_v(t)$ is in $\mathcal{V}$. Therefore, $\xi_{\bar{x}_2}$, the derivative of $h \bullet r_v(t)$ at 0, is in $\mathcal{V}_{\bar{x}_2}$. Since $\xi_{\bar{x}_2}$ can be arbitrary, we have

$$\begin{aligned} g(\xi_{\bar{x}_2}, \zeta_{\bar{x}_2}) &= g(\xi_{h \bullet \bar{x}_1}, \zeta_{h \bullet \bar{x}_1}) \\ &= g(\xi_{\bar{x}_1}, \zeta_{\bar{x}_1}) \text{ (since the inner product is preserved by (9.3.5))} \\ &= 0, \text{ (since } \xi_{\bar{x}_1} \in \mathcal{V}_{\bar{x}_1} \text{ and } \zeta_{\bar{x}_1} \in \mathcal{H}_{\bar{x}_1} \text{)} \end{aligned}$$

for all $\xi_{\bar{x}_2} \in \mathrm{T}_{\bar{x}_2} \mathcal{V}$. Therefore, we obtain $\zeta_{\bar{x}_2} \in \mathcal{H}_{\bar{x}_2}$.

Second, let f be a smooth function defined on a neighborhood of $[\bar{x}_1]$. To show $\zeta_{\bar{x}_2}$ is a horizontal lift of η_x at $\bar{x}_2$, we need to show (9.3.1) which is equivalent to

$$\mathrm{D}\,\pi(\bar{x}_1)[\eta_{\uparrow_{\bar{x}_1}}] f = \mathrm{D}\,\pi(\bar{x}_2)[\zeta_{\bar{x}_2}] f.$$

Let $\bar{f}$ denote $f \circ \pi$. We need to show

$$\frac{d}{dt} \bar{f}(r(t))|_{t=0} = \frac{d}{dt} \bar{f}(h \bullet r(t))|_{t=0}.$$

This holds since $\bar{f}(r(t)) = \bar{f}(h \bullet r(t))$. □

Note that the statement for the Grassmann manifold in [AMS08, Proposition 3.6.1] is a consequence of (iv) of Theorem 9.3.1.

9.3.2 Operations Using n Dimensional Representation

In this section we consider a total space that is also a manifold $\bar{\mathcal{M}} \subset \mathbb{R}^n$. The metric of $\bar{\mathcal{M}}$ is not necessarily a Euclidean metric. Let $G_{\bar{x}}$ denote a matrix expression of the metric at $\bar{x} \in \bar{\mathcal{M}}$. Since the assumptions of $\bar{\mathcal{M}}$ are identical to those in Section 9.2, the results of Section 9.2 can be applied.

The metric is

$$g(\eta_x, \xi_x) = g(\eta_{\uparrow_{\bar{x}}}, \xi_{\uparrow_{\bar{x}}}) = \eta_{\uparrow_{\bar{x}}}^T G_{\bar{x}} \xi_{\uparrow_{\bar{x}}},$$

where $\eta_x, \xi_x \in \mathrm{T}_x \mathcal{M}$, $\eta_{\uparrow_{\bar{x}}}, \xi_{\uparrow_{\bar{x}}} \in \mathcal{H}_{\bar{x}}$ and $G_{\bar{x}} \in \mathbb{R}^{n \times n}$ is a matrix expression of the metric at $\bar{x}$. The relationship between matrix expressions at different representations is given by (iii) of Theorem 9.3.1. $\eta_x^\flat$ at $\bar{x}$ is represented by $\eta_{\uparrow_{\bar{x}}}^\flat = \eta_{\uparrow_{\bar{x}}}^T G_{\bar{x}}$ such that $\eta_x^\flat \xi_x$ at $\bar{x}$ is $\eta_{\uparrow_{\bar{x}}}^\flat \xi_{\uparrow_{\bar{x}}} = \eta_{\uparrow_{\bar{x}}}^T G_{\bar{x}} \xi_{\uparrow_{\bar{x}}}$. $\mathcal{A}_x$, a linear operator on $\mathrm{T}_x \mathcal{M}$, at $\bar{x}$ is denoted by $\mathcal{A}_{\uparrow_{\bar{x}}}$. The relationship between $\mathcal{A}_{\uparrow_{\bar{x}}}$ and $\mathcal{A}_{\uparrow_{h \bullet \bar{x}}}$ is

$$\mathcal{A}_{\uparrow_{h \bullet \bar{x}}} J_h(\bar{x}) \overset{\mathcal{H}_{\bar{x}}}{=\!=} J_h(\bar{x}) \mathcal{A}_{\uparrow_{\bar{x}}}. \tag{9.3.6}$$

By definition, the adjoint $\mathcal{A}_x^*$ of linear operator $\mathcal{A}_x$ satisfies

$$g(\eta_x, \mathcal{A}_x \xi_x) = g(\mathcal{A}_x^* \eta_x, \xi_x).$$

Considering the representation of x at $\bar{x}$, we obtain that the adjoint linear operator $\mathcal{A}_x^*$ at $\bar{x}$ satisfies

$$\eta_{\uparrow_{\bar{x}}}^T G_{\bar{x}} \mathcal{A}_{\uparrow_{\bar{x}}} \xi_{\uparrow_{\bar{x}}} = \eta_{\uparrow_{\bar{x}}}^T (\mathcal{A}_{\uparrow_{\bar{x}}}^*)^T G_{\bar{x}} \xi_{\uparrow_{\bar{x}}},$$

for all $\eta_{\uparrow_{\bar{x}}}, \xi_{\uparrow_{\bar{x}}} \in \mathcal{H}_{\bar{x}}$. Similar to the proof of Lemma 9.2.1, $\mathcal{A}$ is self-adjoint if and only if $B_{\uparrow_{\bar{x}}}^T G_{\bar{x}} \mathcal{A}_{\uparrow_{\bar{x}}} B_{\uparrow_{\bar{x}}}$, which is $B_{\uparrow_{\bar{x}}}^\flat \mathcal{A}_{\uparrow_{\bar{x}}} B_{\uparrow_{\bar{x}}}$, is a symmetric matrix, where the columns of $B_{\uparrow_{\bar{x}}} \in \mathbb{R}^{n \times d}$ are horizontal lifts of columns of an orthonormal basis B_x at $\bar{x}$.

Let $\bar{x}, \bar{y} \in \bar{\mathcal{M}}$ be representations of $x, y \in \mathcal{M}$ respectively. A vector transport $\mathcal{T}$ from x to y can be represented by an n by n matrix $\mathcal{T}_{\eta_{\uparrow_{\bar{x}}}}^{(\bar{x},\bar{y})}$ such that $\mathcal{T}_{\eta_{\uparrow_{\bar{x}}}}^{(\bar{x},\bar{y})} : \mathcal{H}_{\bar{x}} \to \mathcal{H}_{\bar{y}}$, where $\eta_{\uparrow_{\bar{x}}}$ is the horizontal lift of η_x at $\bar{x}$ and η_x satisfies $R_x(\eta_x) = y$. Using the idea of the proof of Lemma 9.2.2, we have that a vector transport is isometric if and only if $B_{\uparrow_{\bar{y}}}^T G_{\bar{y}} \mathcal{T}_{\eta_{\uparrow_{\bar{x}}}}^{(\bar{x},\bar{y})} B_{\uparrow_{\bar{x}}}$, which is $B_{\uparrow_{\bar{y}}}^\flat \mathcal{T}_{\eta_{\uparrow_{\bar{x}}}}^{(\bar{x},\bar{y})} B_{\uparrow_{\bar{x}}}$, is an orthonormal matrix, where the columns of $B_{\uparrow_{\bar{x}}}$ and $B_{\uparrow_{\bar{y}}}$ are horizontal lifts of columns of orthonormal basis B_x and B_y at $\bar{x}$ and $\bar{y}$ respectively.

9.3.3 Construction of Isometric Vector Transports

A vector transport of $\bar{\mathcal{M}}$ can be used to define a vector transport of $\mathcal{M}$. $\bar{\mathcal{T}}$ is a vector transport of $\bar{\mathcal{M}}$ with an associated retraction $\bar{R}$. If $\bar{\mathcal{T}}$ is to induce a vector transport of $\mathcal{M}$ then it must satisfy some conditions. First, the associated retraction $\bar{R}$ must satisfy

$$\tilde{h} \bullet \bar{R}_{\bar{x}}(\eta_{\uparrow_{\bar{x}}}) = \bar{R}_{h\bullet\bar{x}}(\eta_{\uparrow_{h\bullet\bar{x}}}), \tag{9.3.7}$$

for all $h \in \mathfrak{G}$, all $\bar{x} \in \bar{\mathcal{M}}$ and all $\eta_{\uparrow_{\bar{x}}} \in \mathcal{H}_{\bar{x}}$ where $\tilde{h} \in \mathfrak{G}$. Equation (9.3.7) means that when different representations and the corresponding horizontal lifts are chosen, the retracted elements must be able to represent a single element in $\mathcal{M}$. Equation (9.3.7) is a necessary and sufficient condition for a retraction of $\bar{\mathcal{M}}$ to define a retraction of $\mathcal{M}$ by

$$R_x(\eta_x) = \left[\bar{R}_{\bar{x}}(\eta_{\uparrow_{\bar{x}}})\right], \tag{9.3.8}$$

where $\bar{x}$ is a representation of x, $\eta_{\uparrow_{\bar{x}}}$ is a horizontal lift of η_x at $\bar{x}$.

Second, the vector transport $\bar{\mathcal{T}}$ must satisfy also

$$\bar{\mathcal{T}}_{\eta_{\uparrow_{\bar{x}}}}\xi_{\uparrow_{\bar{x}}} \in \mathcal{H}_{\bar{y}} \tag{9.3.9}$$

$$\bar{\mathcal{T}}_{\eta_{\uparrow_{h\bullet\bar{x}}}}\xi_{\uparrow_{h\bullet\bar{x}}} \in \mathcal{H}_{\tilde{h}\bullet\bar{y}} \tag{9.3.10}$$

$$\bar{\mathcal{T}}_{\eta_{\uparrow_{h\bullet\bar{x}}}}\xi_{\uparrow_{h\bullet\bar{x}}} = J_{\tilde{h}}(\bar{y})(\bar{\mathcal{T}}_{\eta_{\uparrow_{\bar{x}}}}\xi_{\uparrow_{\bar{x}}}), \tag{9.3.11}$$

for all $\eta_{\uparrow_{\bar{x}}}, \xi_{\uparrow_{\bar{x}}} \in \mathcal{H}_{\bar{x}}$, all $h \in \mathfrak{G}$, where $\bar{y} = \bar{R}_{\bar{x}}(\eta_{\uparrow_{\bar{x}}})$. Equation (9.3.11) means that when different horizontal lifts are chosen, the transported tangent vectors of $\bar{\mathcal{M}}$ must be horizontal lifts of a single tangent vector of $\mathcal{M}$. Equations (9.3.7), (9.3.9), (9.3.10) and (9.3.11) are necessary and sufficient conditions for a vector transport of $\bar{\mathcal{M}}$ to define a vector transport of $\mathcal{M}$ by

$$\mathcal{T}_{\eta_x}\xi_x = \zeta_y, \tag{9.3.12}$$

where $y = R_x(\eta_x)$, the horizontal lift of $\zeta_y \in \mathrm{T}_y\,\mathcal{M}$ at $\bar{R}_{\bar{x}}(\eta_{\uparrow_{\bar{x}}})$ is $\bar{\mathcal{T}}_{\eta_{\uparrow_{\bar{x}}}}\xi_{\uparrow_{\bar{x}}}$, $\bar{x}$ is a representation of x, $\eta_{\uparrow_{\bar{x}}}$ and $\xi_{\uparrow_{\bar{x}}}$ are horizontal lifts of η_x and ξ_x at $\bar{x}$.

Next, we discuss the modification of vector transports of $\bar{\mathcal{M}}$ based on the idea in Section 9.2.3 to make them define vector transports of $\mathcal{M}$. Since all the vector transports in Section 9.2.3 can be associated with any retraction, the choice of retraction has nothing with designing vector transport. Therefore, we assume (9.3.7) holds.

Let $B : \bar{x} \mapsto B_{\bar{x}}$ denote a function that builds an orthonormal basis of $\mathcal{H}_{\bar{x}}$ and let $C : \bar{x} \mapsto C_{\bar{x}}$ denote a smooth function that builds an orthonormal basis of $\mathcal{V}_{\bar{x}}$. We have the following four isometric vector transports of $\bar{\mathcal{M}}$ from using the idea in Section 9.2.3 but restricted to the Horizontal spaces,

$$\bar{\mathcal{T}} = B_{\bar{y}} U_b^T B_{\bar{x}}^{\flat} + C_{\bar{y}} C_{\bar{x}}^{\flat}, \tag{9.3.13}$$

$$\bar{\mathcal{T}} = G_{\bar{y}}^{-\mathbf{1/2}} (I_n - Q_{\bar{x}} Q_{\bar{x}}^T + Q_{\bar{y}} U_q^T Q_{\bar{x}}^T) G_{\bar{x}}^{\mathbf{1/2}} + C_{\bar{y}} C_{\bar{x}}^{\flat}, \tag{9.3.14}$$

$$\bar{\mathcal{T}} = B_{\bar{y}} B_{\bar{x}}^{\flat} + C_{\bar{y}} C_{\bar{x}}^{\flat}, \tag{9.3.15}$$

$$\bar{\mathcal{T}} = G_{\bar{y}}^{-\mathbf{1/2}} (I_n - Q_{\bar{x}} Q_{\bar{x}}^T - Q_{\bar{y}} Q_{\bar{x}}^T) G_{\bar{x}}^{\mathbf{1/2}} + C_{\bar{y}} C_{\bar{x}}^{\flat}, \tag{9.3.16}$$

where $G : \bar{z} \mapsto G_{\bar{z}}$ is a smooth function and in addition, G_z is symmetric positive definite for (9.3.14) and (9.3.16), U_b is from a polar decomposition of $B_{\bar{x}}^{\flat} B_{\bar{y}} = U_b P_b$ or $(B_{\bar{y}}^{\flat} B_{\bar{x}})^T = P_b U_b$, $Q_{\bar{x}}^T Q_{\bar{y}} = U_q P_q$ is the unique polar decomposition, $Q_{\bar{x}} = \mathrm{orth}((I - \tilde{N}_{\bar{x}} \tilde{N}_{\bar{x}}^T) \tilde{N}_{\bar{y}})$, $\tilde{N}_{\bar{x}} = G_{\bar{x}}^{1/2} N_{\bar{x}}$ (as in (9.2.17), $G_{\bar{x}}^{1/2}$ is the usual Euclidean operator definition of the matrix square root decomposition for $G_{\bar{x}}$) , $N : \bar{x} \mapsto N_{\bar{x}}$ is a function that builds an orthonormal basis of $(\mathcal{H}_{\bar{x}})_{\perp} = \{v \in \mathbb{R}^n | v^T G_{\bar{x}} \eta_{\uparrow_{\bar{x}}} = 0 \text{ for all } \eta_{\uparrow_{\bar{x}}} \in \mathcal{H}_{\bar{x}}\}$, likewise $Q_{\bar{y}}$ with x and y interchanged and $\mathrm{orth}(A)$ denotes orthonormalizing A. Functions, $N : \bar{x} \mapsto N_{\bar{x}}$ and $B : \bar{x} \mapsto B_{\uparrow_{\bar{x}}}$, are not necessarily smooth for (9.3.13) and (9.3.14), but are required to be smooth for (9.3.15) and (9.3.16).

When the actions of (9.3.13), (9.3.14), (9.3.15) and (9.3.16) are restricted on $\mathcal{H}_{\bar{x}}$, all of them satisfy (9.3.9) and (9.3.10). In addition, the last term $C_{\bar{y}} C_{\bar{x}}^{\flat}$ is zero. Therefore, they reduce to the following four mappings

$$\bar{\mathcal{T}} = B_{\bar{y}} U_b^T B_{\bar{x}}^{\flat}, \tag{9.3.17}$$

$$\bar{\mathcal{T}} = G_{\bar{y}}^{-\mathbf{1/2}} (I_n - Q_{\bar{x}} Q_{\bar{x}}^T + Q_{\bar{y}} U_q^T Q_{\bar{x}}^T) G_{\bar{x}}^{\mathbf{1/2}}, \tag{9.3.18}$$

$$\bar{\mathcal{T}} = B_{\bar{y}} B_{\bar{x}}^{\flat}, \tag{9.3.19}$$

$$\bar{\mathcal{T}} = G_{\bar{y}}^{-\mathbf{1/2}} (I_n - Q_{\bar{x}} Q_{\bar{x}}^T - Q_{\bar{y}} Q_{\bar{x}}^T) G_{\bar{x}}^{\mathbf{1/2}}. \tag{9.3.20}$$

Unfortunately, the four mappings do not satisfy (9.3.11) in general. Lemma 9.3.1 provides sufficient conditions for them to define vector transports of $\mathcal{M}$.

Lemma 9.3.1. *Suppose the retractions associated with* (9.3.17)*,* (9.3.18)*,* (9.3.19) *and* (9.3.20) *satisfy* (9.3.7) *with* $\tilde{h} = h$*.*

(i) If (9.3.4) *holds for all $h \in \mathfrak{G}$ and $J_h(\bar{x})$ is independent of $\bar{x}$ for all $\bar{x} \in \bar{\mathcal{M}}$ and all $h \in \mathfrak{G}$, i.e., $J_h(\bar{x}) = J_h(\bar{y})$ for all $\bar{x}, \bar{y} \in \bar{\mathcal{M}}$ and all $h \in \mathfrak{G}$, then* (9.3.17) *defines an vector transport of $\mathcal{M}$ by* (9.3.12)*.*

(ii) If (9.3.4) *holds for all $h \in \mathfrak{G}$ and $G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) G_{\bar{x}}^{-1/2}$ is independent of $\bar{x}$ for all $\bar{x} \in \bar{\mathcal{M}}$ and all $h \in \mathfrak{G}$, i.e., $G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) G_{\bar{x}}^{-1/2} = G_{h\bullet\bar{y}}^{1/2} J_h(\bar{y}) G_{\bar{y}}^{-1/2}$ for all $\bar{x}, \bar{y} \in \bar{\mathcal{M}}$ and all $h \in \mathfrak{G}$, then* (9.3.18) *defines an vector transport of $\mathcal{M}$ by* (9.3.12)*.*

(iii) If the function $B : \bar{x} \mapsto B_{\uparrow_{\bar{x}}}$ satisfies $J_h(\bar{x}) B_{\uparrow_{\bar{x}}} = B_{\uparrow_{h\bullet\bar{x}}}$ for all $\bar{x} \in \bar{\mathcal{M}}$ and all $h \in \mathfrak{G}$, then (9.3.19) *defines an vector transport of $\mathcal{M}$ by* (9.3.12)*.*

(iv) If (9.3.4) *holds for all $h \in \mathfrak{G}$, $G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) G_{\bar{x}}^{-1/2}$ is independent of $\bar{x}$ for all $\bar{x} \in \bar{\mathcal{M}}$ and all $h \in \mathfrak{G}$, i.e., $G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) G_{\bar{x}}^{-1/2} = G_{h\bullet\bar{y}}^{1/2} J_h(\bar{y}) G_{\bar{y}}^{-1/2}$ for all $\bar{x}, \bar{y} \in \bar{\mathcal{M}}$ and all $h \in \mathfrak{G}$ and the function $N : \bar{x} \mapsto N_{\bar{x}}$ satisfies $J_h(\bar{x}) N_{\bar{x}} = N_{h\bullet\bar{x}}$ for all $\bar{x} \in \bar{\mathcal{M}}$ and all $h \in \mathfrak{G}$, then* (9.3.20) *defines an vector transport of $\mathcal{M}$ by* (9.3.12)*.*

Proof. (i): Let $B : x \mapsto B_x$ be a smooth function that builds a basis of $\mathrm{T}_x \mathcal{M}$. We choose $B_{\uparrow_{\bar{x}}}$ such that the columns of $B_{\uparrow_{\bar{x}}}$ are the horizontal lifts of B_x at $\bar{x}$. We consider the first choice of U_b, i.e., from the polar decomposition of $B_{\uparrow_{\bar{x}}}^{\flat} B_{\uparrow_{\bar{y}}} = U_b P_b$. The derivation of second choice of U_b is similar and we do not include it. We have

$$\begin{aligned}
B_{\uparrow_{h\bullet\bar{x}}}^{\flat} B_{\uparrow_{h\bullet\bar{y}}} &= B_{\uparrow_{h\bullet\bar{x}}}^{T} G_{h\bullet\bar{x}} B_{\uparrow_{h\bullet\bar{y}}} \\
&= B_{\uparrow_{\bar{x}}}^{T} J_h(\bar{x})^T G_{h\bullet\bar{x}} J_h(\bar{y}) B_{\uparrow_{\bar{y}}} \text{ (by (iv) of Theorem 9.3.1)} \\
&= B_{\uparrow_{\bar{x}}}^{T} J_h(\bar{x})^T G_{h\bullet\bar{x}} J_h(\bar{x}) B_{\uparrow_{\bar{y}}} \text{ (by assumption)} \\
&= B_{\uparrow_{\bar{x}}}^{T} G_{\bar{x}} B_{\uparrow_{\bar{y}}} \text{ (by assumption)} \\
&= B_{\uparrow_{\bar{x}}}^{\flat} B_{\uparrow_{\bar{y}}},
\end{aligned}$$

which implies that U_b from polar decomposition of $B_{\uparrow_{h\bullet\bar{x}}}^{\flat} B_{\uparrow_{h\bullet\bar{y}}}$ is the same as U_b from polar decomposition of $B_{\uparrow_{\bar{x}}}^{\flat} B_{\uparrow_{\bar{y}}}$. Therefore, it follows that

$$\begin{aligned}
\bar{\mathcal{T}}_{\eta_{\uparrow_{h\bullet\bar{x}}}} \xi_{\uparrow_{h\bullet\bar{x}}} &= B_{\uparrow_{h\bullet\bar{y}}} U_b B_{\uparrow_{h\bullet\bar{x}}}^{\flat} \xi_{\uparrow_{h\bullet\bar{x}}} \\
&= J_h(\bar{y}) B_{\uparrow_{\bar{y}}} U_b B_{\uparrow_{\bar{x}}}^{T} J_h(\bar{x})^T G_{\uparrow_{h\bullet\bar{x}}} J_h(\bar{x}) \xi_{\uparrow_{\bar{x}}} \text{ (by (iv) of Theorem 9.3.1)} \\
&= J_h(\bar{y}) B_{\uparrow_{\bar{y}}} U_b B_{\uparrow_{\bar{x}}}^{T} G_{\bar{x}} \xi_{\uparrow_{\bar{x}}} \text{ (by (iii) of Theorem 9.3.1)} \\
&= J_h(\bar{y}) (\bar{\mathcal{T}}_{\eta_{\uparrow_{\bar{x}}}} \xi_{\uparrow_{\bar{x}}}),
\end{aligned}$$

which is the desired result. Moreover, noting that (9.3.17) is independent of the choice of the bases of $\mathcal{H}_{\bar{x}}$ and $\mathcal{H}_{\bar{y}}$, $B_{\bar{x}}$ need not be the horizontal lifts of B_x. The only requirement for $B_{\bar{x}}$ is that it is an orthonormal basis of $\mathcal{H}_{\bar{x}}$.

(ii): Let $B : x \mapsto B_x$ be a smooth function that builds a basis of $\mathrm{T}_x\,\mathcal{M}$. We choose $B_{\uparrow_{\bar{x}}}$ such that the columns of $B_{\uparrow_{\bar{x}}}$ are the horizontal lifts of B_x at $\bar{x}$. Noting that (9.3.18) is equivalent to

$$\bar{\mathcal{T}} = B_{\uparrow_{\bar{y}}} U_b^T B_{\uparrow_{\bar{x}}}^{\flat},$$

where U_b is from the polar decomposition, $B_{\uparrow_{\bar{x}}}^T G_{\bar{x}}^{1/2} G_{\bar{y}}^{1/2} B_{\uparrow_{\bar{y}}} = U_b P_b$. We have

$$\begin{aligned}
B_{\uparrow_{h\bullet\bar{x}}}^T G_{h\bullet\bar{x}}^{1/2} G_{h\bullet\bar{y}}^{1/2} B_{\uparrow_{h\bullet\bar{y}}} &= B_{\uparrow_{\bar{x}}}^T J_h(\bar{x})^T G_{h\bullet\bar{x}}^{1/2} G_{h\bullet\bar{y}}^{1/2} J_h(\bar{y}) B_{\uparrow_{\bar{y}}} \text{ (by (iv) of Theorem 9.3.1)} \\
&= B_{\uparrow_{\bar{x}}}^T G_{\bar{x}}^{1/2} G_{\bar{x}}^{-1/2} J_h(\bar{x})^T G_{h\bullet\bar{x}}^{1/2} G_{h\bullet\bar{y}}^{1/2} J_h(\bar{y}) G_{\bar{y}}^{-1/2} G_{\bar{y}}^{1/2} B_{\uparrow_{\bar{y}}} \\
&= B_{\uparrow_{\bar{x}}}^T G_{\bar{x}}^{1/2} G_{\bar{x}}^{-1/2} J_h(\bar{x})^T G_{h\bullet\bar{x}}^{1/2} G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) G_{\bar{x}}^{-1/2} G_{\bar{y}}^{1/2} B_{\uparrow_{\bar{y}}} \text{ (by assumption)} \\
&= B_{\uparrow_{\bar{x}}}^T G_{\bar{x}}^{1/2} G_{\bar{y}}^{1/2} B_{\uparrow_{\bar{y}}}, \text{ (by (9.3.4))}
\end{aligned}$$

which implies that U_b from polar decomposition of $B_{\uparrow_{h\bullet\bar{x}}}^T G_{h\bullet\bar{x}}^{1/2} G_{h\bullet\bar{y}}^{1/2} B_{\uparrow_{h\bullet\bar{y}}}$ is the same as U_b from polar decomposition of $B_{\uparrow_{\bar{x}}}^T G_{\bar{x}}^{1/2} G_{\bar{y}}^{1/2} B_{\uparrow_{\bar{y}}}$. Result (ii) follows by using the idea in the proof of (i).

(iii): We have

$$\begin{aligned}
\bar{\mathcal{T}}_{\eta_{\uparrow_{h\bullet\bar{x}}}} \xi_{\uparrow_{h\bullet\bar{x}}} &= B_{\uparrow_{h\bullet\bar{y}}} B_{\uparrow_{h\bullet\bar{x}}}^{\flat} \xi_{\uparrow_{h\bullet\bar{x}}} \\
&= J_h(\bar{y}) B_{\uparrow_{\bar{y}}} B_{\bar{x}}^T J_h(\bar{x})^T G_{\uparrow_{h\bullet\bar{x}}} J_h(\bar{x}) \xi_{\bar{x}} \text{ (by (iv) of Theorem 9.3.1)} \\
&= J_h(\bar{y}) B_{\uparrow_{\bar{y}}} B_{\bar{x}}^T G_{\bar{x}} \xi_{\bar{x}} \text{ (by (iii) of Theorem 9.3.1)} \\
&= J_h(\bar{y}) (\bar{\mathcal{T}}_{\eta_{\uparrow_{\bar{x}}}} \xi_{\uparrow_{\bar{x}}}),
\end{aligned}$$

which gives the desired result.

(iv): By (9.3.4), we have

$$G_{\bar{x}} = J_h(\bar{x})^T G_{h\bullet\bar{x}} J_h(\bar{x}).$$

It follows that

$$\begin{aligned}
I_n &= G_{\bar{x}}^{-1/2} J_h(\bar{x})^T G_{h\bullet\bar{x}}^{1/2} G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) G_{\bar{x}}^{-1/2} \\
&= (G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) G_{\bar{x}}^{-1/2})^T G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) G_{\bar{x}}^{-1/2},
\end{aligned}$$

which means that $G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) G_{\bar{x}}^{-1/2}$ is an orthonormal matrix for all $\bar{x} \in \bar{\mathcal{M}}$. We have

$$\begin{aligned}
&(I_n - \tilde{N}_{h\bullet\bar{x}} \tilde{N}_{h\bullet\bar{x}}^T) \tilde{N}_{h\bullet\bar{y}} \\
&= (I_n - G_{h\bullet\bar{x}}^{1/2} N_{h\bullet\bar{x}} N_{h\bullet\bar{x}}^T G_{h\bullet\bar{x}}^{1/2}) G_{h\bullet\bar{y}}^{1/2} N_{h\bullet\bar{y}} \\
&= (I_n - G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) N_{\bar{x}} N_{\bar{x}}^T J_h(\bar{x})^T G_{h\bullet\bar{x}}^{1/2}) G_{h\bullet\bar{y}}^{1/2} J_h(\bar{y}) N_{\bar{y}} \text{ (by assumption)} \\
&= (I_n - G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) G_{\bar{x}}^{-1/2} G_{\bar{x}}^{1/2} N_{\bar{x}} N_{\bar{x}}^T G_{\bar{x}}^{1/2} G_{\bar{x}}^{-1/2} J_h(\bar{x})^T G_{h\bullet\bar{x}}^{1/2}) G_{h\bullet\bar{y}}^{1/2} J_h(\bar{y}) G_{\bar{y}}^{-1/2} G_{\bar{y}}^{1/2} N_{\bar{y}} \\
&= G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) G_{\bar{x}}^{-1/2} (I_n - G_{\bar{x}}^{1/2} N_{\bar{x}} N_{\bar{x}}^T G_{\bar{x}}^{1/2}) G_{\bar{y}}^{1/2} N_{\bar{y}} \text{ (by assumption and (9.3.4))} \\
&= G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) G_{\bar{x}}^{-1/2} (I_n - \tilde{N}_{\bar{x}} \tilde{N}_{\bar{x}}^T) \tilde{N}_{\bar{y}}.
\end{aligned}$$

Similarly, we have

$$(I_n - \tilde{N}_{h\bullet\bar{y}} \tilde{N}_{h\bullet\bar{y}}^T) \tilde{N}_{h\bullet\bar{x}} = G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) G_{\bar{x}}^{-1/2} (I_n - \tilde{N}_{\bar{y}} \tilde{N}_{\bar{y}}^T) \tilde{N}_{\bar{x}}.$$

Using the two equations above and noting that $O_h = G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) G_{\bar{x}}^{-1/2}$ is an orthonormal matrix, we have

$$\begin{aligned}
Q_{h\bullet\bar{x}} &= O_h Q_{\bar{x}} \\
Q_{h\bullet\bar{y}} &= O_h Q_{\bar{y}}.
\end{aligned}$$

We have also that

$$\begin{aligned}
\bar{\mathcal{T}}_{\eta_{\uparrow_{h\bullet\bar{x}}}} \xi_{\uparrow_{h\bullet\bar{x}}} &= G_{h\bullet\bar{y}}^{\boldsymbol{-1/2}} (I_n - Q_{h\bullet\bar{x}} Q_{h\bullet\bar{x}}^T - Q_{h\bullet\bar{y}} Q_{h\bullet\bar{x}}^T) G_{h\bullet\bar{x}}^{\boldsymbol{1/2}} \xi_{\uparrow_{h\bullet\bar{x}}} \\
&= G_{h\bullet\bar{y}}^{-1/2} (I_n - Q_{h\bullet\bar{x}} Q_{h\bullet\bar{x}}^T - Q_{h\bullet\bar{y}} Q_{h\bullet\bar{x}}^T) G_{h\bullet\bar{x}}^{1/2} \xi_{\uparrow_{h\bullet\bar{x}}} \\
&= G_{h\bullet\bar{y}}^{-1/2} (I_n - O_h Q_{\bar{x}} Q_{\bar{x}}^T O_h^T - O_h Q_{\bar{y}} Q_{\bar{x}}^T O_h^T) G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) \xi_{\uparrow_{\bar{x}}} \text{ (by (iv) of Theorem 9.3.1)} \\
&= G_{h\bullet\bar{y}}^{-1/2} O_h (I_n - Q_{\bar{x}} Q_{\bar{x}}^T - Q_{\bar{y}} Q_{\bar{x}}^T) O_h^T G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) \xi_{\uparrow_{\bar{x}}} \\
&= G_{h\bullet\bar{y}}^{-1/2} G_{h\bullet\bar{y}}^{1/2} J_h(\bar{y}) G_{\bar{y}}^{-1/2} (I_n - Q_{\bar{x}} Q_{\bar{x}}^T - Q_{\bar{y}} Q_{\bar{x}}^T) G_{\bar{x}}^{-1/2} J_h(\bar{x})^T G_{h\bullet\bar{x}}^{1/2} G_{h\bullet\bar{x}}^{1/2} J_h(\bar{x}) \xi_{\uparrow_{\bar{x}}} \\
&= J_h(\bar{y}) G_{\bar{y}}^{-1/2} (I_n - Q_{\bar{x}} Q_{\bar{x}}^T - Q_{\bar{y}} Q_{\bar{x}}^T) G_{\bar{x}}^{-1/2} J_h(\bar{x})^T G_{h\bullet\bar{x}} J_h(\bar{x}) \xi_{\uparrow_{\bar{x}}} \\
&= J_h(\bar{y}) G_{\bar{y}}^{-1/2} (I_n - Q_{\bar{x}} Q_{\bar{x}}^T - Q_{\bar{y}} Q_{\bar{x}}^T) G_{\bar{x}}^{1/2} \xi_{\uparrow_{\bar{x}}} \text{ (by (9.3.4))} \\
&= J_h(\bar{y}) (\bar{\mathcal{T}}_{\eta_{\uparrow_{\bar{x}}}} \xi_{\uparrow_{\bar{x}}})
\end{aligned}$$

which is the desired result. □

Another method to satisfy (9.3.11) is to fix a representation when an equivalence class is given. First of all, we define a section of a quotient manifold.

Definition 9.3.3. *$\mathcal{M}$ is a quotient manifold with the total manifold $\bar{\mathcal{M}}$. $\mathcal{S} \subset \bar{\mathcal{M}}$ is called a section of the quotient manifold $\mathcal{M}$ if it satisfies*

(i) $\mathcal{S}$ is a submanifold of $\bar{\mathcal{M}}$;

(ii) $\mathcal{S}$ intersects each equivalence class at most once.

If $\mathcal{S}$ intersects every equivalence class once, then $\mathcal{S}$ is called a global section of $\mathcal{M}$, otherwise, it is called a local section of $\mathcal{M}$.

If a section $\mathcal{S}$ of the quotient manifold $\mathcal{M}$ is known, then a vector transport $\bar{\mathcal{T}}$ of $\bar{\mathcal{M}}$ with associated retraction $\bar{R}$ can define a vector transport $\mathcal{T}$ of $\mathcal{M}$ by

$$\mathcal{T}_{\eta_x} \xi_x = \zeta_y,$$

where $y = R_x(\eta_x)$, the horizontal lift of ζ_y at $\bar{R}_{\bar{x}}(\eta_{\uparrow \bar{x}})$ is $\bar{\mathcal{T}}_{\eta_{\uparrow \bar{x}}} \xi_{\uparrow \bar{x}}$ and $\bar{x} \in \mathcal{S}$.

9.4 Product of Manifolds

9.4.1 General Discussion

A product of manifolds is a manifold and is denoted by $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2 \times \cdots \times \mathcal{M}_m$, where $\mathcal{M}_i$ is a manifold. In this section, we assume $\mathcal{M}_i$ is a d_i-dimensional manifold that can be represented by n_i-dimensional vector. The dimension of the manifold $\mathcal{M}$ is $d = \sum_{i=1}^m d_i$. The product space in which the representations of elements of $\mathcal{M}$ reside is $\mathbb{R}^n$ where $n = \sum_{i=1}^m n_i$. However, the metric on $\mathcal{M}$ is not necessarily related to a particular metric on $\mathbb{R}^n$. An element x in $\mathcal{M}$ is denoted by $x = (x_1^T, x_2^T, \cdots, x_m^T)^T$, where $x_i \in \mathcal{M}_i$. The tangent space of $\mathcal{M}$ is

$$\mathrm{T}_x \mathcal{M} = \mathrm{T}_{x_1} \mathcal{M}_1 \times \mathrm{T}_{x_2} \mathcal{M}_2 \times \cdots \times \mathrm{T}_{x_m} \mathcal{M}_m.$$

The tangent vector η_x is denoted by $(\eta_{x_1}^T, \eta_{x_2}^T, \cdots, \eta_{x_m}^T)^T$.

The metric of $\mathcal{M}_i$ is denoted by $g_i(\eta_{x_i}, \xi_{x_i}) = \eta_{x_i}^T G_{x_i} \xi_{x_i}$, where G_{x_i} is an n_i by n_i matrix. The metric of $\mathcal{M}$ is

$$g(\eta_x, \xi_x) = \sum_{i=1}^m g_i(\eta_{x_i}, \xi_{x_i}) = \eta_x^T \operatorname{diag}(G_{x_1}, G_{x_2}, \cdots, G_{x_m}) \xi_x = \eta_x^T G_x \xi_x,$$

where $G_x = \operatorname{diag}(G_{x_1}, G_{x_2}, \cdots, G_{x_m}) \in \mathbb{R}^{n \times n}$ is a block diagonal matrix. The definitions of $\flat$, adjoint and vector transport are the same as in Section 9.2.1.

9.4.2 Construction of Isometric Vector Transports

We have discussed methods of constructing isometric vector transports for each $\mathcal{M}_i$. Let $\mathcal{T}_i$ denote the associated isometric vector transports. An isometric vector transport for $\mathcal{M}$ can be defined as

$$\begin{aligned}\mathcal{T}_{\eta_x}\xi_x &= (((\mathcal{T}_1)_{\eta_{x_1}}\xi_{x_1})^T, ((\mathcal{T}_2)_{\eta_{x_2}}\xi_{x_2})^T, \cdots, ((\mathcal{T}_m)_{\eta_{x_m}}\xi_{x_m})^T)^T \\ &= \mathrm{diag}((\mathcal{T}_1)_{\eta_{x_1}}, (\mathcal{T}_2)_{\eta_{x_2}}, \cdots, (\mathcal{T}_m)_{\eta_{x_m}})\xi_x.\end{aligned}$$

9.5 The Intrinsic Dimensional Approach

As discussed in Section 9, a d-dimensional representation for a tangent vector requires a choice of basis. If the function B_x that builds an orthonormal basis is smooth, then we can use this function to assign an orthonormal basis to each point on an open set of the manifold. Under this framework, many operations become computationally inexpensive.

Only the implementation for a manifold in $\mathbb{R}^n$ is discussed in this section. When working on a quotient manifold with the total manifold in $\mathbb{R}^n$, vector transport, the metric, and linear operators on the tangent spaces are, in fact, related to the horizontal space of the particular representative element of the equivalence class associated with elements of the quotient manifold. This horizontal space is simply a d-dimensional subspace of $\mathbb{R}^n$ and therefore the implementation is easily adapted from the implementation discussed in the remainder of this section.

9.5.1 General Discussion

B is a smooth function to build an orthonormal basis, i.e., $B : x \to B_x$ is smooth. From the discussion of Section 4.6, this function always exists at least locally. In practice, we assume that the region on which it exists contains the level set $\{x | f(x) \leq f(x_0)\}$. If $\eta_x, \xi_x \in \mathrm{T}_x \mathcal{M}$, then, since $\mathrm{T}_x \mathcal{M}$ is a d-dimensional subspace of $\mathbb{R}^n$, both η_x and ξ_x are n-dimensional vectors. By the function B_x, we have that the d-dimensional representations of η_x and ξ_x are

$$v_x = B_x^\flat \eta_x \text{ and } u_x = B_x^\flat \xi_x. \tag{9.5.1}$$

The inner product has a simple form, $g(v_x, u_x) = v_x^T u_x$. From (9.5.1), we have

$$g(v_x, u_x) = v_x^T u_x = \eta_x^T (B_x^\flat)^T u_x = \eta_x^T G_x B_x u_x = \eta_x^T G_x \xi_x,$$

which is consistent with (9.2.9). The linear operator $\mathcal{A}$ on a tangent space, $\mathrm{T}_x\,\mathcal{M}$, is a $d \times d$ matrix. $\mathcal{A}$ is self-adjoint if and only if $\mathcal{A} = \mathcal{A}^T$.

9.5.2 Computational Benefits

When d-dimensional representations are used, the matrix expression of an inner product becomes inexpensive and simple since $\hat{G}_x = I_d$. The vector transports (9.2.19) and (9.3.19) are both

$$\mathcal{T} = B_y B_x^\flat.$$

The d-dimensional representation of the vector transport is

$$\begin{aligned}\mathcal{T}^d v_x &= B_y^\flat \mathcal{T} \eta_x \text{ (by (9.5.1) and notice } v_x \text{ represents } \eta_x) \\ &= B_y^\flat B_y B_x^\flat B_x v_x \\ &= v_x, \qquad (9.5.2)\end{aligned}$$

where $\mathcal{T}^d \in \mathbb{R}^{d\times d}$ is a d-dimensional representation of vector transport. We can see that the matrix representations of the vector transports (9.2.19) and (9.3.19) become the identity. In other words, the vector transport are implicit and automatically done by changing bases. This is the most important advantage of this representation. We avoid many operations especially the expensive one, i.e., $\mathcal{T}_\eta \circ \mathcal{H} \circ \mathcal{T}_\eta^{-1}$. The only requirement is that a smooth function of building an orthonormal basis for $\mathrm{T}_x\,\mathcal{M}$ with acceptable computationally complexity is known.

CHAPTER 10

IMPLEMENTATION FOR SOME MANIFOLDS

10.1 Introduction

Chapter 9 contains a general discussion for manifolds that can be represented by a vector in $\mathbb{R}^n$. In this chapter, specific implementations for common manifolds, the sphere, the Stiefel manifold, the orthogonal group and the Grassmann manifold, are presented. The discussed implementations include matrix expression of metrics, linear operators on a tangent space, retractions and vector transports. For retractions and vector transports, details for constructing pairs that satisfy the locking condition are provided. Methods of computing the cotangent vector $\mathrm{D}\, f_{R_x}(s)$ required by Ring and Wirth's RBFGS [RW12] are also given.

This chapter is organized as follows. In Sections 10.2, 10.3, 10.4, 10.5 and 10.6, the implementation of the four manifolds are presented. Finally, computational complexity is presented in Section 10.7.

10.2 The Stiefel Manifold as an Embedded Submanifold

The Stiefel manifold is a frequently encountered manifold in practice. Edelman, Arias and Smith [EAS98] have discussed its properties. The Stiefel manifold can be viewed as an embedded submanifold or a quotient manifold. Their metrics are not equivalent except when $p = 1$ or $p = n$. When $p = 1$, the manifold is the sphere and when $p = n$, the manifold is the orthogonal group. They are considered in Section 10.4 and Section 10.5 respectively and it is assumed that $1 < p < n$ for all other discussions of the Stiefel manifold. In this section, an embedded submanifold view of the Stiefel manifold is taken. Section 10.3 considers the Stiefel manifold as a quotient manifold.

The Stiefel manifold can be defined as a submanifold of np-dimensional Euclidean space by

$$\mathrm{St}(p, n) := \{X \in \mathbb{R}^{n \times p} : X^T X = I_p\},$$

where I_p denotes the p by p identity matrix. The tangent space is

$$\mathrm{T}_X \mathrm{St}(p, n) = \{X\Omega + X_\perp K : \Omega^T = -\Omega, K \in \mathbb{R}^{(n-p)\times p}\},$$

where $X_\perp$ is any $n \times (n-p)$ orthonormal matrix such that $\begin{bmatrix} X & X_\perp \end{bmatrix}$ is an orthogonal matrix. The metric endowed from the Euclidean space is

$$g(U, V) = \mathrm{trace}(U^T V), \tag{10.2.1}$$

where $U, V \in \mathrm{T}_X \mathrm{St}(p, n)$. Let "vec" denote the operation that stacks the columns of its matrix arguments into a single vector. The inner product can be written in vector form,

$$g(\mathrm{vec}(U), \mathrm{vec}(V)) = \mathrm{vec}(U)^T \mathrm{vec}(V).$$

Therefore, the matrix expression G_X of inner product is $np \times np$ identity matrix for all $X \in \mathrm{St}(p, n)$.

10.2.1 Retractions

The exponential mapping given in [EAS98] is

$$\mathrm{Exp}_X(U) = \begin{pmatrix} X & U \end{pmatrix} \left(\exp \begin{pmatrix} A & -S \\ I_p & A \end{pmatrix} \right) \begin{pmatrix} I_p \\ 0 \end{pmatrix} \exp(-A), \tag{10.2.2}$$

where $X \in \mathrm{St}(p, n)$, $U \in \mathrm{T}_X \mathrm{St}(p, n)$, $A = X^T U$ and $S = U^T U$. Some other retractions are given in [AMS08].

$$R_X(U) = \mathrm{qf}(X + U), \tag{10.2.3}$$

$$R_X(U) = (X + U)(I_p + U^T U)^{-1/2}, \tag{10.2.4}$$

where $\mathrm{qf}(A)$ denotes the Q factor of the QR decomposition with nonnegative elements on the diagonal of R. (10.2.4) uses the polar decomposition.

10.2.2 Vector Transports

The differentiated retraction of the exponential mapping (10.2.2) can be derived using [NH95, Theorems 4.5 and 4.15]. The resulting vector transport, however, is computationally very expensive. Alternatives are required for efficiency.

Two vector transports are given in [AMS08, Section 7]. The first is vector transport by projection denoted by $\mathcal{T}_P$

$$\mathcal{T}_{P_U} V = V - Y \,\mathrm{sym}(Y^T V), \tag{10.2.5}$$

where $U, V \in \mathrm{T}_X \operatorname{St}(p,n)$, $Y = R_X(U)$ and R is the associated retraction. The second is vector transport by differentiated retraction (10.2.3)

$$\mathcal{T}_{R_U} V = Y \rho_{skew}(Y^T V (Y^T(X+U))^{-1}) \\ + (I_n - YY^T) V (Y^T(X+U))^{-1}, \tag{10.2.6}$$

where $U, V \in \mathrm{T}_X\, St(p,n)$, R is (10.2.3), $Y = R_X(U)$ and

$$(\rho_{skew}(B))_{i,j} = \begin{cases} B_{i,j}, & \text{if } i > j; \\ 0, & \text{if } i = j; \\ -B_{j,i}, & \text{if } i < j. \end{cases}$$

The following lemma derives the differentiated retraction of (10.2.4).

Lemma 10.2.1. *The differentiated retraction of* (10.2.4) *is*

$$\mathcal{T}_{R_U} V = Y\Omega + (I_n - YY^T) V (Y^T(X+U))^{-1}, \tag{10.2.7}$$

where $Y = R_X(U)$*,* R *is* (10.2.4) *and* $\operatorname{vec}(\Omega) = ((Y^T(X+U)) \oplus (Y^T(X+U)))^{-1} \operatorname{vec}(Y^T V - V^T Y)$ *and* $\oplus$ *is the Kronecker sum, i.e.,* $A \oplus B = A \otimes I + I \otimes B$*.*

Proof. Let $t \mapsto W(t)$ be a curve on the noncompact Stiefel manifold $\mathbb{R}_*^{n \times p}$, i.e., the set of matrices with full column rank. $\dot{W}(0) = V$ and let $W(t) = Y(t)P(t)$ denote the polar decomposition of $W(t)$. We have

$$\dot{W} = \dot{Y}P + Y\dot{P}. \tag{10.2.8}$$

We also have the decomposition

$$\dot{Y} = YY^T\dot{Y} + (I_n - YY^T)\dot{Y}. \tag{10.2.9}$$

Therefore, we need only obtain $YY^T\dot{Y}$ and $(I_n - YY^T)\dot{Y}$.

Multiplying (10.2.8) by $I_n - YY^T$ on the left and P^{-1} on the right, we obtain

$$(I_n - YY^T)\dot{W}P^{-1} = (I_n - YY^T)\dot{Y},$$

which is the second part of (10.2.9).

Note the expression of the tangent space of the Stiefel manifold. $Y^T\dot{Y}$ is a skew symmetric matrix and is denoted by Ω. Multiplying (10.2.8) by Y^T, we have

$$Y^T\dot{W} = Y^T\dot{Y}P + \dot{P} = \Omega P + \dot{P}.$$

Since the tangent space of a symmetric positive definite matrix is a symmetric matrix, $\dot{P}$ is a symmetric matrix. Hence, we obtain

$$Y^T \dot{W} - \dot{W}^T Y = \Omega P - P^T \Omega^T = \Omega P + P \Omega.$$

Rewriting it as an Kronecker sum expression, we obtain

$$(P \oplus P) \operatorname{vec}(\Omega) = \operatorname{vec}(Y^T \dot{W} - \dot{W}^T Y).$$

By [Lau04, Theorem 13.16] and noting P is positive definite, we know $(P \oplus P)$ is a full rank matrix. Therefore, a unique solution for Ω exists and is given by $\operatorname{vec}(\Omega) = (P \oplus P)^{-1} \operatorname{vec}(Y^T \dot{W} - \dot{W}^T Y)$.

In summary, we have

$$\dot{Y} = Y \Omega + (I_n - Y Y^T) \dot{W} P^{-1}. \tag{10.2.10}$$

Computing $\mathcal{T}_{R_U} V = \frac{d}{dt} R_X(U + tV)|_{t=0}$ is equivalent to computing $\dot{Y}(0)$ when $W(t) = X + U + tV$. Therefore, we have $\dot{W} = V$, $Y = Y(0) = R_X(U)$ and $P = Y(0)^T W(0) = Y^T(X + U)$. Substituting into (10.2.10) yields the desired result. □

The sufficient conditions of the convergence analyses of the RBroyden family and RTR-SR1 require isometric vector transports. However, (10.2.3), (10.2.6), (10.2.7) are non-isometric. This does not prevent their usefulness in practice under certain conditions without theoretical guarantees of convergence. In Section 11.4.5, the performance of these non-isometric vector transports are assessed empirically.

Parallel translation is the most natural isometric vector transport geometrically. Edelman, Arias and Smith [EAS98] indicated that the closed expression for parallel translation is unknown. The parallel translation of U along the geodesic $\gamma(t)$ in direction $\dot{\gamma}(t) = V$ satisfies the ordinary differential equation

$$\omega'(t) = -1/2 \gamma(t)(\gamma'(t)^T \omega(t) + \omega(t)^T \gamma'(t)), \tag{10.2.11}$$

where $\omega(t) = P_\gamma^{t \leftarrow 0} U$. Solving the differential equation is expensive and parallel translation is not a good choice of isometric vector transport in practice. Fortunately, the idea of Section 9.2.3 can be used to construct isometric vector transports. The only remaining requirement is the construction of functions that build a basis for the tangent spaces.

Note that $\mathrm{T}_X \operatorname{St}(p,n) = \{X\Omega + X_\perp K : \Omega^T = -\Omega, K \in \mathbb{R}^{(n-p)\times p}\}$. An orthonormal basis of $\mathrm{T}_x \operatorname{St}(p,n)$ denoted by B_x is given by

$$\{\frac{1}{\sqrt{2}} X(e_i e_j^T - e_j e_i^T) : i = 1,\dots,p, j = i+1,\dots,p\} \cup \{X_\perp \tilde{e}_i e_j^T, i = 1,\dots,n-p, j = 1,\dots,p\}, \tag{10.2.12}$$

where $(e_1,\dots,e_p)$ is the canonical basis of $\mathbb{R}^p$ and $(\tilde{e}_1,\dots,\tilde{e}_{n-p})$ is the canonical basis of $\mathbb{R}^{n-p}$. Similarly, note that $\mathrm{N}_X\mathrm{St}(p,n) = \{XS : S \in \mathbb{R}^{p\times p},\ S = S^T\}$ (see, e.g., [AMS08, Example 3.6.2]), hence an orthonormal basis of $\mathrm{N}_X\mathrm{St}(p,n)$ denoted by N_x is given by

$$\{X e_i e_i^T : i = 1,\dots,p\} \cup \{\frac{1}{\sqrt{2}} X(e_i e_j^T + e_j e_i^T) : i = 1,\dots,p, j = i+1,\dots,p\}. \tag{10.2.13}$$

The columns of B_X and N_X are thus chosen as the "vec" of the basis elements.

We can see that the function of constructing N_x is smooth. Meanwhile, if $X_\perp$ is smoothly dependent on X, B_x is also a smooth function. The vector transports (9.2.17), (9.2.18), (9.2.19) and (9.2.20) are all isometric vector transports.

10.2.3 Pairs of Retraction and Isometric Vector Transport Satisfying Locking Condition

The RBroyden family of algorithms require the retraction and isometric vector transport used to satisfy the locking condition (4.2.6). Chapter 4.4 provides three methods. The first two are straightforward and are not discussed in detail. We discuss the implementation of the third method in this section.

With the exception of (9.2.19), for the Stiefel manifold, the vector transports discussed in Section 9.2.3 are not linear functions with respect to y. Therefore, an efficient retraction can be derived based on the isometric vector transport (9.2.19) as follows. Let $X \in \operatorname{St}(p,n)$, $U \in \mathrm{T}_X \operatorname{St}(p,n)$ and denote the unknown $R_X(tU)$ as $X(t)$. Equation (4.4.10) implies

$$\frac{d}{dt} \operatorname{vec}(X(t)) = B_{X(t)} B_X^T \operatorname{vec}(U), \tag{10.2.14}$$

where $B_{X(t)}$ is the orthonormal basis of $\mathrm{T}_{X(t)} \operatorname{St}(p,n)$ given by (10.2.12) and $X = X(0)$. Using the expression of B_X, we have that

$$B_X^T \operatorname{vec}(U) = \begin{pmatrix} \sqrt{2}\operatorname{vectriu}(\Omega_U) \\ \operatorname{vec}(K_U) \end{pmatrix},$$

where $\Omega_U = X^T U$, $K_U = X_\perp^T U$ and for $M \in \mathbb{R}^{p\times p}$,

$$\mathrm{vectriu}(M) = (M_{12}, M_{13}, \dots, M_{1p}, M_{23} \dots, M_{2p}, \dots, M_{(p-1)p})^T.$$

Substituting into (10.2.14), we have

$$\frac{d}{dt}\,\mathrm{vec}(X(t)) = B_{X(t)} \begin{pmatrix} \sqrt{2}\,\mathrm{vectriu}(\Omega_U) \\ \mathrm{vec}(K_U) \end{pmatrix},$$

which yields

$$\frac{d}{dt}X(t) = \begin{pmatrix} X(t) & X_\perp(t) \end{pmatrix} \begin{pmatrix} \Omega_U \\ K_U \end{pmatrix}. \tag{10.2.15}$$

Note $(X(t), X_\perp(t))$ is a smooth curve on the orthogonal group and the tangent space of the orthogonal group has form (10.5.1). We have that

$$\frac{d}{dt}\begin{pmatrix} X(t) & X_\perp(t) \end{pmatrix} = \begin{pmatrix} X(t) & X_\perp(t) \end{pmatrix} \Omega_n,$$

where $\Omega_n \in \mathbb{R}^{n\times n}$ is a skew symmetric matrix. Since (10.2.15) holds, we have

$$\frac{d}{dt}\begin{pmatrix} X(t) & X_\perp(t) \end{pmatrix} = \begin{pmatrix} X(t) & X_\perp(t) \end{pmatrix} \begin{pmatrix} \Omega_U & -K^T \\ K & \Omega_{(n-p)} \end{pmatrix},$$

where $\Omega_{(n-p)} \in \mathbb{R}^{(n-p)\times(n-p)}$ is some skew symmetric matrix. Without loss of generality, we choose $\Omega_{(n-p)}$ to be a zero matrix and we obtain

$$\frac{d}{dt}\begin{pmatrix} X(t) & X_\perp(t) \end{pmatrix} = \begin{pmatrix} X(t) & X_\perp(t) \end{pmatrix} \begin{pmatrix} \Omega_U & -K^T \\ K & 0 \end{pmatrix}.$$

By rewriting it in the form of Kronecker product, we have

$$\frac{d}{dt}\,\mathrm{vec}\left(\begin{pmatrix} X(t) & X_\perp(t) \end{pmatrix}\right) = \left(\begin{pmatrix} \Omega_U & -K^T \\ K & 0 \end{pmatrix}^T \otimes I_n\right) \mathrm{vec}\left(\begin{pmatrix} X(t) & X_\perp(t) \end{pmatrix}\right).$$

The solution is

$$\begin{aligned} \mathrm{vec}\left(\begin{pmatrix} X(t) & X_\perp(t) \end{pmatrix}\right) &= \exp\left(\begin{pmatrix} \Omega_U & -K^T \\ K & 0 \end{pmatrix}^T \otimes I_n\right) \mathrm{vec}\left(\begin{pmatrix} X & X_\perp \end{pmatrix}\right) \\ &= \left(\exp\begin{pmatrix} \Omega_U & -K^T \\ K & 0 \end{pmatrix}^T \otimes I_n\right) \mathrm{vec}\left(\begin{pmatrix} X & X_\perp \end{pmatrix}\right). \end{aligned}$$

It follows that

$$\begin{pmatrix} X(t) & X_\perp(t) \end{pmatrix} = \begin{pmatrix} X & X_\perp \end{pmatrix} \exp\begin{pmatrix} \Omega_U & -K^T \\ K & 0 \end{pmatrix}.$$

Therefore, we have the desired retraction

$$Y = R_X(U) = X(1) = \begin{pmatrix} X & X_\perp \end{pmatrix} \exp \begin{pmatrix} \Omega_U & -K^T \\ K & 0 \end{pmatrix} \begin{pmatrix} I_p \\ 0 \end{pmatrix} \tag{10.2.16}$$

and the $Y_\perp$ in the basis B_Y of the isometric vector transport (9.2.19) is $X_\perp(1)$. One interesting phenomenon is that the retraction (10.2.16) is equivalent to the exponential mapping (10.3.2) with the Stiefel manifold canonical metric (10.3.1).

The differentiated retraction of the retraction (10.2.16) can be derived as follows. Let $U, V \in \mathrm{T}_X \mathrm{St}(p,n)$, $\Omega_U = X^T U$, $K_U = X_\perp^T U$, $\Omega_V = X^T V$, $K_V = X_\perp^T V$. We have

$$\begin{aligned} \mathcal{T}_{R_U} V &= \frac{d}{dt} \mathrm{Exp}_X(U + tV)|_{t=0} \\ &= \begin{pmatrix} X & X_\perp \end{pmatrix} \frac{d}{dt} \exp \begin{pmatrix} \Omega_U + t\Omega_V & -K_U^T - tK_V^T \\ K_U + tK_V & 0 \end{pmatrix} |_{t=0} \begin{pmatrix} I_p \\ 0 \end{pmatrix}. \end{aligned}$$

Let

$$M_1 = \begin{pmatrix} \Omega_U & -K_U^T \\ K_U & 0 \end{pmatrix} \text{ and } M_2 = \begin{pmatrix} \Omega_V & -K_V^T \\ K_V & 0 \end{pmatrix}.$$

It follows that

$$\begin{aligned} \mathcal{T}_{R_U} V &= \begin{pmatrix} X & X_\perp \end{pmatrix} \frac{d}{dt} \exp(M_1 + tM_2)|_{t=0} \begin{pmatrix} I_p \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} X & X_\perp \end{pmatrix} Z((Z^{-1} M_2 Z) \odot \Phi) Z^{-1} \begin{pmatrix} I_p \\ 0 \end{pmatrix} \text{ (by [NH95, Theorem 4.5])}, \end{aligned} \tag{10.2.17}$$

where $M_1 = Z \Lambda Z^{-1}$ is the spectral decomposition, $\lambda_i = \Lambda_{ii}$, $\odot$ denotes the Hadamard product, i.e. $(A \odot B)_{ij} = A_{ij} B_{ij}$ and

$$\Phi_{ij} = \Phi_{ji} = \begin{cases} \frac{e^{\lambda_i} - e^{\lambda_j}}{\lambda_i - \lambda_j}, & \text{if } \lambda_i \neq \lambda_j; \\ e^{\lambda_i}, & \text{if } \lambda_i = \lambda_j. \end{cases}$$

10.2.4 Cotangent Vector Required by Ring and Wirth's RBFGS

Ring and Wirth's RBFGS [RW12] requires $\mathrm{D} f_{R_x}(s)$ which is a cotangent vector of the tangent space $\mathrm{T}_x \mathcal{M}$, i.e., $\mathrm{D} f_{R_x}(s)[\eta] = g(\mathrm{grad} f(R_x(s)), \mathcal{T}_{R_s} \eta)$, $\eta \in \mathrm{T}_x \mathcal{M}$. In general, if $\eta, \xi \in \mathrm{T}_x \mathcal{M}$, $y = R_x(\xi)$ and $\zeta \in \mathrm{T}_y \mathcal{M}$ then the cotangent vector in $\mathrm{T}_x \mathcal{M}$ defined by x, ξ, ζ is $C(x, \xi, \zeta)[\eta] = g(\zeta, \mathcal{T}_{R_\xi} \eta)$ and, thus, $\mathrm{D} f_{R_x}(s)$ is $C(x, s, \mathrm{grad} f(R_x(s)))$. In this section, we discuss the methods of computing the cotangent vector $C(\cdot, \cdot, \cdot)$ for the embedded Stiefel manifold. The methods of computing cotangent vector for the quotient Stiefel manifold and Grassmann manifold are given in Sections 10.3.3 and 10.7.

Let $U, V \in \mathrm{T}_X \operatorname{St}(p, n)$ and $W \in \mathrm{T}_Y \operatorname{St}(p, n)$, $Y = R_X(U)$. Considering the retraction (10.2.3), we have that

$$\begin{aligned}
&C(X, U, W)[V] \\
&= \operatorname{trace}(W^T(Y\rho_{skew}(Y^TV(Y^T(X+U))^{-1}) + (I_n - YY^T)V(Y^T(X+U))^{-1})) \\
&= \operatorname{trace}(W^TY(\operatorname{tril}(Y^TV(Y^T(X+U))^{-1}) - \operatorname{tril}(Y^TV(Y^T(X+U))^{-1})^T)) \\
&\quad + \operatorname{trace}(W^T(I_n - YY^T)V(Y^T(X+U))^{-1}) \\
&= \operatorname{trace}(W^TY \operatorname{tril}(Y^TV(Y^T(X+U))^{-1})) - \operatorname{trace}(\operatorname{tril}(Y^TV(Y^T(X+U))^{-1})Y^TW) \\
&\quad + \operatorname{trace}((Y^T(X+U))^{-1}W^T(I_n - YY^T)V) \\
&= \operatorname{trace}(\operatorname{tril}(Y^TV(Y^T(X+U))^{-1})(W^TY - Y^TW)) \\
&\quad + \operatorname{trace}((Y^T(X+U))^{-1}W^T(I_n - YY^T)V)
\end{aligned}$$

where $\operatorname{tril}(M)$ is the lower triangular part of the matrix M without including the diagonal elements. Noting that W^TY is a skew symmetric matrix and $\operatorname{trace}(\operatorname{tril}(A)B) = \operatorname{trace}(A \operatorname{triu}(B))$, we have

$$\begin{aligned}
&C(X, U, W)[V] \\
&= \operatorname{trace}(Y^TV(Y^T(X+U))^{-1} \operatorname{triu}(2W^TY)) + \operatorname{trace}((Y^T(X+U))^{-1}W^T(I_n - YY^T)V) \\
&= \operatorname{trace}((Y^T(X+U))^{-1}(\operatorname{triu}(2W^TY)Y^T + W^T(I_n - YY^T))V).
\end{aligned}$$

where $\operatorname{triu}(M)$ is the upper triangular part of the matrix M without including the diagonal elements. Therefore, the cotangent vector for the retraction (10.2.3) is

$$C(X, U, W) = (Y^T(X+U))^{-1}(\operatorname{triu}(2W^TY)Y^T + W^T(I_n - YY^T)). \tag{10.2.18}$$

We also can compute the cotangent vector corresponding to the retraction (10.2.4). Using the same notations, we have

$$\begin{aligned}
C(X, U, W)[V] &= \operatorname{trace}(W^T(Y\Omega + (I_n - YY^T)V(Y^T(X+U))^{-1})) \\
&= \operatorname{trace}(W^TY\Omega) + \operatorname{trace}((Y^T(X+U))^{-1}W^T(I_n - YY^T)V),
\end{aligned}$$

where $\operatorname{vec}(\Omega) = ((Y^T(X+U)) \oplus (Y^T(X+U)))^{-1} \operatorname{vec}(Y^TV - V^TY)$. This expression does not lend itself to rewriting $\operatorname{trace}(W^TY\Omega)$ as $\operatorname{trace}(MV)$, where M is some matrix, therefore, we use a

Kronecker product form. Specifically, we have

$$\begin{aligned}
&\mathrm{trace}(W^T Y \Omega) \\
&= \mathrm{vec}(Y^T W)^T \mathrm{vec}(\Omega) \\
&= \mathrm{vec}(Y^T W)^T ((Y^T(X+U)) \oplus (Y^T(X+U)))^{-1} \mathrm{vec}(Y^T V - V^T Y) \\
&= \mathrm{vec}(Y^T W)^T ((Y^T(X+U)) \oplus (Y^T(X+U)))^{-1} ((I_p \otimes Y^T) - (Y^T \otimes I_p) L) \mathrm{vec}(V),
\end{aligned}$$

where $L \in \mathbb{R}^{np \times np}$ satisfies $L \, \mathrm{vec}(V) = \mathrm{vec}(V^T)$, i.e.,

$$(L)_{i,j} = \begin{cases} 1, & i = (k-1)p + h,\ j = (h-1)n + k \text{ where } h = 1, \ldots, p,\ k = 1, \ldots, n; \\ 0, & \text{otherwise.} \end{cases}$$

Therefore, we have

$$\begin{aligned}
&C(X, U, W)[V] \\
&= \mathrm{vec}(Y^T W)^T ((Y^T(X+U)) \oplus (Y^T(X+U)))^{-1} ((I_p \otimes Y^T) - (Y^T \otimes I_p) L) \mathrm{vec}(V) \\
&\quad + \mathrm{trace}((Y^T(X+U))^{-1} W^T (I_n - Y Y^T) V),
\end{aligned}$$

which means that the cotangent vector is

$$C(X, U, W) = Z^T + (Y^T(X+U))^{-1} W^T (I_n - Y Y^T), \tag{10.2.19}$$

where $\mathrm{vec}(Z)^T = \mathrm{vec}(Y^T W)^T ((Y^T(X+U)) \oplus (Y^T(X+U)))^{-1} ((I_p \otimes Y^T) - (Y^T \otimes I_p) L)$.

The cotangent vector corresponding to the retraction (10.2.16) satisfies

$$\begin{aligned}
C(X, U, W)[V] &= \mathrm{trace}(W^T \begin{pmatrix} X & X_\perp \end{pmatrix} Z((Z^{-1} M_2 Z) \odot \Phi) Z^{-1} \begin{pmatrix} I_p \\ 0 \end{pmatrix}) \\
&= \mathrm{trace}(Z^{-1} \begin{pmatrix} I_p \\ 0 \end{pmatrix} W^T \begin{pmatrix} X & X_\perp \end{pmatrix} Z((Z^{-1} M_2 Z) \odot \Phi)) \\
&= \mathrm{trace}(((Z^{-1} \begin{pmatrix} I_p \\ 0 \end{pmatrix} W^T \begin{pmatrix} X & X_\perp \end{pmatrix} Z) \odot \Phi^T)(Z^{-1} M_2 Z)) \\
&= \mathrm{trace}(M_3 M_2),
\end{aligned}$$

where

$$M_3 = (Z((Z^{-1} \begin{pmatrix} I_p \\ 0 \end{pmatrix} W^T \begin{pmatrix} X & X_\perp \end{pmatrix} Z) \odot \Phi^T) Z^{-1}) \in \mathbb{R}^{n \times n}.$$

Splitting M_3 into a 2 by 2 block matrix,

$$M_3 = \begin{pmatrix} M_3^{(11)} & M_3^{(12)} \\ M_3^{(21)} & M_3^{(22)} \end{pmatrix},$$

where $M_3^{(11)} \in \mathbb{R}^{p\times p}$, $M_3^{(12)} \in \mathbb{R}^{p\times(n-p)}$, $M_3^{(21)} \in \mathbb{R}^{(n-p)\times p}$, and $M_3^{(22)} \in \mathbb{R}^{(n-p)\times(n-p)}$ and using the expression of M_2, we have

$$\begin{aligned} C(X,U,W)[V] &= \mathrm{trace}(M_3 M_2) \\ &= \mathrm{trace}(M_3^{(11)} X^T V + M_3^{(12)} X_\perp^T V - M_3^{(21)} V^T X_\perp) \\ &= \mathrm{trace}(\begin{pmatrix} M_3^{(11)} & M_3^{(12)} - (M_3^{(21)})^T \end{pmatrix} \begin{pmatrix} X^T \\ X_\perp^T \end{pmatrix} V). \end{aligned}$$

Therefore, the cotangent vector is

$$C(X,U,W) = \begin{pmatrix} M_3^{(11)} & M_3^{(12)} - (M_3^{(21)})^T \end{pmatrix} \begin{pmatrix} X^T \\ X_\perp^T \end{pmatrix}. \tag{10.2.20}$$

The cotangent vector (10.2.19) contains some big matrix operations. Whether it has efficient implementations is still unknown. Therefore, the cotangent vectors (10.2.18) and (10.2.20) are used in some experiments in the experiment Chapter 11.

10.3 The Stiefel Manifold as a Quotient Manifold

When the Stiefel manifold is viewed as a quotient manifold, the total space is the orthogonal group $\mathcal{O}(n)$. Two elements $O_1, O_2 \in \mathcal{O}(n)$ are called equivalent if there exists $Z \in \mathcal{O}(n-p)$ such that

$$O_1 = O_2 \begin{pmatrix} I_p & \\ & Z \end{pmatrix}.$$

The equivalence class is

$$[O] = \{O \begin{pmatrix} I_p & \\ & Z \end{pmatrix} | Z \in \mathcal{O}(n-p)\} = O\mathcal{O}(n-p).$$

The quotient Stiefel manifold can be written in term of $\mathcal{O}(n)/\mathcal{O}(n-p)$. The first p columns of any element in an equivalence class are the same and the last $n-p$ columns are an arbitrary orthonormal matrix. The first p columns are therefore used to represent the equivalence class. The set of representations is

$$\mathrm{St}(p,n) = \{X \in \mathbb{R}^{n\times p} : X^T X = I_p\},$$

which is the same as the embedded Stiefel manifold. Edelman, Arias and Smith [EAS98, Section 2.4.1] pointed out that the quotient Stiefel manifold is equivalent to the submanifold of $\mathbb{R}^{n\times p}$ with the canonical metric (10.3.1). Therefore, even though the Stiefel manifold is treated as a quotient manifold, we still work on a submanifold of $\mathbb{R}^{n\times p}$ but with a different metric.

The tangent space of $\mathrm{St}(p,n)$ with the metric defined below is

$$\mathrm{T}_X \mathrm{St}(p,n) = \{X\Omega + X_\perp K : \Omega^T = -\Omega, K \in \mathbb{R}^{(n-p)\times p}\},$$

where $X_\perp$ is any $n \times (n-p)$ orthonormal matrix such that $\begin{bmatrix} X & X_\perp \end{bmatrix}$ is an orthogonal matrix. The canonical metric is

$$g(U,V) = \mathrm{trace}(U^T(I_n - \frac{1}{2}XX^T)V), \tag{10.3.1}$$

where $U, V \in \mathrm{T}_X \mathrm{St}(p,n)$. Rewriting it as a vector expression, we have

$$g(\mathrm{vec}(U), \mathrm{vec}(V)) = \mathrm{vec}(U)^T \,\mathrm{diag}(I_n - \frac{1}{2}XX^T, \cdots, I_n - \frac{1}{2}XX^T)\,\mathrm{vec}(V).$$

Hence G_X is a block diagonal matrix, $\mathrm{diag}(I_n - \frac{1}{2}XX^T, \cdots, I_n - \frac{1}{2}XX^T)$, where the number of blocks is p.

10.3.1 Retractions

The exponential mapping given in [EAS98] is

$$\mathrm{Exp}_X(U) = \begin{pmatrix} X & Q \end{pmatrix} \left(\exp \begin{pmatrix} A & -R^T \\ R & 0 \end{pmatrix} \right) \begin{pmatrix} I_p \\ 0 \end{pmatrix}, \tag{10.3.2}$$

where Q and R is the compact QR factorization of $(I - XX^T)U$. The retractions (10.2.3) and (10.2.4) also can be used here since a retraction is independent of a metric.

10.3.2 Vector Transports

Noting that

$$I_n - \frac{1}{2}XX^T = \begin{pmatrix} X & X_\perp \end{pmatrix} \begin{pmatrix} \frac{1}{2}I_p & 0 \\ 0 & I_{n-p} \end{pmatrix} \begin{pmatrix} X & X_\perp \end{pmatrix}^T$$

is positive definite for all $X \in \mathrm{St}(p,n)$, we can extend the metric (10.3.1) to be on $\mathbb{R}^{n\times p}$. Therefore, the "orthonormal" based on this metric is well-defined on $\mathbb{R}^n$ and thus we obtain a projection to a tangent space that turns out to be identical to the projection when the metric is endowed from the Euclidean space. The vector transport by projection is then identical to (10.2.5), i.e.,

$$\mathcal{T}_{P_U} V = V - Y\,\mathrm{sym}(Y^T V).$$

The vector transport by differentiated retraction (10.2.3) is

$$\begin{aligned}\mathcal{T}_{R_U} V = Y \rho_{skew}(Y^T V (Y^T (X+U))^{-1}) \\ + (I_n - YY^T) V (Y^T (X+U))^{-1},\end{aligned}$$

and the vector transport by differentiated retraction (10.2.4) is

$$\mathcal{T}_{R_U} V = Y\Omega + (I_n - YY^T) V (Y^T (X+U))^{-1},$$

where $Y = R_X(U)$, $\mathrm{vec}(\Omega) = ((Y^T(X+U)) \oplus (Y^T(X+U)))^{-1} \mathrm{vec}(Y^T V - V^T Y)$ and $\oplus$ is the Kronecker sum, i.e., $A \oplus B = A \otimes I + I \otimes B$. Both of them are the same as (10.2.6) and (10.2.7) since they are independent of metrics.

Parallel translation is dependent on the metric and is different from (10.2.11). The parallel translation of U along the geodesic $\gamma(t)$ in direction $\dot{\gamma}(t) = V$ satisfies the ordinary differential equation

$$\begin{aligned}\omega'(t) = -1/2(\omega(t)\gamma'(t)^T + \gamma'(t)\omega(t)^T)\gamma(t) \\ - 1/2\gamma(t)(\gamma'(t)^T(I - \gamma(t)\gamma(t)^T)\omega(t) + \omega(t)^T(I - \gamma(t)\gamma(t)^T)\gamma'(t)),\end{aligned} \tag{10.3.3}$$

where $\omega(t) = P_\gamma^{t \leftarrow 0} U$. As with (10.2.11), the closed form is unknown. Therefore, the parallel translation is also not a good choice of an isometric vector transport in practice.

As with the embedded manifold view of the Stiefel manifold, the idea of Section 9.2.3 can be used to construct isometric vector transports. This requires the consideration of generating bases for the tangent and normal spaces. Even though the tangent space is the same as that of the embedded Stiefel manifold, the orthonormal basis is different due to the difference of metric. By simple derivation, we have an orthonormal basis of $\mathrm{T}_x \mathrm{St}(p,n)$ denoted by B_x is

$$\{X(e_i e_j^T - e_j e_i^T) : i = 1, \dots, p, j = i+1, \dots, p\} \cup \{X_\perp \tilde{e}_i e_j^T, i = 1, \dots, n-p, j = 1, \dots, p\},$$

where $(e_1, \dots, e_p)$ is the canonical basis of $\mathbb{R}^p$, $(\tilde{e}_1, \dots, \tilde{e}_{n-p})$ is the canonical basis of $\mathbb{R}^{n-p}$. An orthonormal basis $\mathrm{N}_x \mathrm{St}(p,n)$ denoted by N_x is then

$$\{\sqrt{2} X e_i e_i^T : i = 1, \dots, p\} \cup \{X(e_i e_j^T + e_j e_i^T) : i = 1, \dots, p, j = i+1, \dots, p\}.$$

The columns of B_X and N_X are then given by the "vec" of the basis elements.

Although G_X is not identity, $G_X^{1/2}$ and $G_X^{-1/2}$ are not difficult to compute. We have $G_X^{1/2} = \mathrm{diag}(I - (1 - 1/\sqrt{2})XX^T, \cdots, I - (1 - 1/\sqrt{2})XX^T)$ and $G_X^{-1/2} = \mathrm{diag}(I + (\sqrt{2} - 1)XX^T, \cdots, I + (\sqrt{2} - 1)XX^T)$ where the number of blocks is p. One can find that $\tilde{B}_x$ and $\tilde{N}_x$ are (10.2.12) and (10.2.13).

The basic requirements are satisfied with these vector transports. It is clear that the function constructing N_x is smooth. If $X_\perp$ is a smoothly dependent on X, B_x is also a smooth function. The vector transports (9.2.17), (9.2.18), (9.2.19) and (9.2.20) are all isometric vector transports.

Using the procedures in Section 10.2.3, we can obtain a retraction corresponding to the vector transport (9.2.19) such that the pair satisfies the locking condition (4.2.6). In fact, the result is identical to that in Section 10.2.3.

We have

$$\begin{pmatrix} X(t) & X_\perp(t) \end{pmatrix} = \begin{pmatrix} X & X_\perp \end{pmatrix} \exp \begin{pmatrix} \Omega_U & -K^T \\ K & 0 \end{pmatrix}.$$

The desired retraction is

$$Y = R_X(U) = X(1) = \begin{pmatrix} X & X_\perp \end{pmatrix} \exp \begin{pmatrix} \Omega_U & -K^T \\ K & 0 \end{pmatrix} \begin{pmatrix} I_p \\ 0 \end{pmatrix}. \tag{10.3.4}$$

and $Y_\perp$ in the basis B_Y of the isometric vector transport (9.2.19) is $X_\perp(1)$.

The differentiated retraction of (10.3.4) is the same as (10.2.17),

$$\mathcal{T}_{R_U} V = \begin{pmatrix} X & X_\perp \end{pmatrix} Z((Z^{-1} M_2 Z) \odot \Phi) Z^{-1} \begin{pmatrix} I_p \\ 0 \end{pmatrix},$$

where $M_1 = Z \Lambda Z^{-1}$ is the spectral decomposition, $\lambda_i = \Lambda_{ii}$, $\odot$ denotes the Hadamard product, i.e. $(A \odot B)_{ij} = A_{ij} B_{ij}$ and

$$\Phi_{ij} = \Phi_{ji} = \begin{cases} \frac{e^{\lambda_i} - e^{\lambda_j}}{\lambda_i - \lambda_j}, & \text{if } \lambda_i \neq \lambda_j; \\ e^{\lambda_i}, & \text{if } \lambda_i = \lambda_j. \end{cases}$$

10.3.3 Cotangent Vector Required by Ring and Wirth's RBFGS

The discussion in this section follows that in Section 10.2.4 while noting the differences between the metrics (10.3.1) and (10.2.1). The cotangent vectors $C(\cdot, \cdot, \cdot)$ corresponding to different retractions are not the same as those in Section 10.2.4 since they are dependent on the metric.

Let $U, V \in \mathrm{T}_X \operatorname{St}(p,n)$ and $W \in \mathrm{T}_Y \operatorname{St}(p,n)$, $Y = R_X(U)$. For the retraction (10.2.3), we have that

$$\begin{aligned}
&C(X,U,W)[V] \\
={}& \operatorname{trace}(W^T(I_n - \frac{1}{2}YY^T)(Y\rho_{skew}(Y^TV(Y^T(X+U))^{-1}) + (I_n - YY^T)V(Y^T(X+U))^{-1})) \\
={}& \operatorname{trace}((Y^T(X+U))^{-1}(\operatorname{triu}(W^TY)Y^T + W^T(I_n - YY^T))V)
\end{aligned}$$

Therefore, the cotangent vector for the retraction (10.2.3) is

$$C(X,U,W) = (Y^T(X+U))^{-1}(\operatorname{triu}(W^TY)Y^T + W^T(I_n - YY^T)).$$

Similarly for the retraction (10.2.6), we have

$$\begin{aligned}
&C(X,U,W)[V] \\
={}& \operatorname{trace}(W^T(I_n - \frac{1}{2}YY^T)(Y\Omega + (I_n - YY^T)V(Y^T(X+U))^{-1})) \\
={}& \frac{1}{2}\operatorname{vec}(Y^TW)^T((Y^T(X+U)) \oplus (Y^T(X+U)))^{-1}((I_p \otimes Y^T) - (Y^T \otimes I_p)L)\operatorname{vec}(V) \\
&+ \operatorname{trace}((Y^T(X+U))^{-1}W^T(I_n - YY^T)V),
\end{aligned}$$

where $\operatorname{vec}(\Omega) = ((Y^T(X+U)) \oplus (Y^T(X+U)))^{-1}\operatorname{vec}(Y^TV - V^TY)$ and $L \in \mathbb{R}^{np \times np}$ satisfies $L\operatorname{vec}(V) = \operatorname{vec}(V^T)$, i.e.,

$$(L)_{i,j} = \begin{cases} 1, & i = (k-1)p + h,\ j = (h-1)n + k \text{ where } h = 1, \ldots, p,\ k = 1, \ldots, n; \\ 0, & \text{otherwise.} \end{cases}$$

The cotangent vector is

$$C(X,U,W) = Z^T + (Y^T(X+U))^{-1}W^T(I_n - YY^T),$$

where $\operatorname{vec}(Z)^T = \frac{1}{2}\operatorname{vec}(Y^TW)^T((Y^T(X+U)) \oplus (Y^T(X+U)))^{-1}((I_p \otimes Y^T) - (Y^T \otimes I_p)L)$.

The cotangent vector corresponding to the retraction (10.3.4) is

$$\begin{aligned}
C(X,U,W)[V] &= \operatorname{trace}(W^T(I_n - \frac{1}{2}YY^T)\begin{pmatrix} X & X_\perp \end{pmatrix} Z((Z^{-1}M_2Z) \odot \Phi)Z^{-1}\begin{pmatrix} I_p \\ 0 \end{pmatrix}) \\
&= \operatorname{trace}(M_4M_2),
\end{aligned}$$

where

$$M_4 = (Z((Z^{-1}\begin{pmatrix} I_p \\ 0 \end{pmatrix} W^T(I_n - \frac{1}{2}YY^T)\begin{pmatrix} X & X_\perp \end{pmatrix} Z) \odot \Phi^T)Z^{-1}) \in \mathbb{R}^{n \times n}.$$

Splitting M_4 into a 2 by 2 block matrix,

$$M_4 = \left(\begin{array}{cc} M_4^{(11)} & M_4^{(12)} \\ M_4^{(21)} & M_4^{(22)} \end{array} \right),$$

where $M_4^{(11)} \in \mathbb{R}^{p\times p}$, $M_4^{(12)} \in \mathbb{R}^{p\times(n-p)}$, $M_4^{(21)} \in \mathbb{R}^{(n-p)\times p}$, and $M_4^{(22)} \in \mathbb{R}^{(n-p)\times(n-p)}$ and noting the expression of M_2, we have

$$\begin{aligned} C(X,U,W)[V] &= \operatorname{trace}(M_4 M_2) \\ &= \operatorname{trace}(M_4^{(11)} X^T V + M_4^{(12)} X_\perp^T V - M_4^{(21)} V^T X_\perp) \\ &= \operatorname{trace}(\left(\begin{array}{cc} M_4^{(11)} & M_4^{(12)} - (M_4^{(21)})^T \end{array} \right) \left(\begin{array}{c} X^T \\ X_\perp^T \end{array} \right) V). \end{aligned}$$

Therefore, the cotangent vector is

$$C(X,U,W) = \left(\begin{array}{cc} M_4^{(11)} & M_4^{(12)} - (M_4^{(21)})^T \end{array} \right) \left(\begin{array}{c} X^T \\ X_\perp^T \end{array} \right).$$

10.4 The Sphere

Since the sphere is the Stiefel manifold with $p = 1$, i.e.,

$$\mathbb{S}^{n-1} := \{x \in \mathbb{R}^n | x^T x = 1\}$$

the discussions and techniques of the previous sections apply. Therefore, in this section the results that arise from applying those techniques are simply stated. Details are only given when the particular structure of the sphere allows extensions to the earlier results.

The tangent space of $\mathbb{S}^{n-1}$ is

$$\mathrm{T}_x\, \mathbb{S}^{n-1} = \{\eta \in \mathbb{R}^n : x^T \eta = 0\}.$$

The two metrics corresponding to an embedded submanifold and a quotient manifold are

$$g_e(\eta, \xi) = \eta^T \xi \text{ and } g_q(\eta, \xi) = \eta^T (I_n - \frac{1}{2} x x^T) \xi,$$

where $\eta, \xi \in \mathrm{T}_x\, \mathbb{S}^{n-1}$. Since $x^T \eta = x^T \xi = 0$ on the sphere, the two metrics are identical, i.e.,

$$g(\eta, \xi) = g_e(\eta, \xi) = g_q(\eta, \xi). \qquad (10.4.1)$$

10.4.1 Retractions

Two retractions of sphere are given in [AMS08]. The simplest retraction is achieved by normalizing the length of the vector,

$$R_x(\eta) = \frac{x + \eta}{\|x + \eta\|_2}. \tag{10.4.2}$$

The exponential mapping is given by

$$\mathrm{Exp}_x(\eta) = x \cos(\|\eta\|) + \frac{\eta}{\|\eta\|} \sin(\|\eta\|), \tag{10.4.3}$$

where $x \in \mathbb{S}^{n-1}$ and $\eta \in \mathrm{T}_x\,\mathbb{S}^{n-1}$.

10.4.2 Vector Transports

There are two non-isometric vector transports in [AMS08]. Vector transport by projection is given by

$$\mathcal{T}_{P_\eta}\xi = \xi - yy^T\xi,$$

where $\eta, \xi \in \mathbb{S}^{n-1}$, $y = R_x(\eta)$ and R is the associated retraction. On the sphere, there is considerable simplification of vector transport by differentiated retraction (10.4.2) compared to the general Stiefel case. It is simply a scaling of vector transport by projection, i.e.,

$$\mathcal{T}_{R_\eta}\xi = \frac{\xi - yy^T\xi}{\|x + \eta\|_2},$$

where $\eta, \xi \in \mathbb{S}^{n-1}$, $y = R_x(\eta)$ and R is (10.4.2).

For the sphere, the closed form of parallel translation along the geodesic is known,

$$P_\gamma^{1\leftarrow 0}\xi = \xi - \frac{2\xi^T y}{\|x + y\|_2^2}(x + y), \tag{10.4.4}$$

where $\xi \in \mathrm{T}_x\,\mathbb{S}^{n-1}$, γ is the geodesic from x to y and $\gamma(0) = x, \gamma(1) = y$.

Since the codimension of $\mathbb{S}^{n-1}$ is 1, isometric vector transports can be constructed by (9.2.17) and (9.2.20). They produce the same isometric vector transport,

$$\mathcal{T}_{S_\eta}\xi = (I_n - qq^T - \tilde{q}q^T)\xi, \tag{10.4.5}$$

where $q = (I_n - xx^T)y/\|(I_n - xx^T)y\|_2$ and $\tilde{q} = (I_n - yy^T)x/\|(I_n - yy^T)x\|_2$.

In fact, not only (9.2.17) and (9.2.20) produce the same vector transport. The next lemma shows that parallel translation on a sphere is also equivalent.

Lemma 10.4.1. (10.4.4) *and* (10.4.5) *are equivalent.*

Proof. We have

$$\|(I_n - xx^T)y\|_2^2 = y^Ty + (x^Ty)^2x^Tx - 2(x^Ty)^2 = 1 - (x^Ty)^2 = (1 - x^Ty)(1 + x^Ty),$$
$$\|(I_n - yy^T)x\|_2^2 = x^Tx + (x^Ty)^2y^Ty - 2(x^Ty)^2 = 1 - (x^Ty)^2 = (1 - x^Ty)(1 + x^Ty).$$

Therefore, we have $\|(I - xx^T)y\|_2 = \|(I - yy^T)x\|_2$. It follows that

$$\begin{aligned}(I - qq^T - \tilde{q}q^T)\xi &= \left(I - \frac{(I - xx^T)yy^T(I - xx^T)}{\|(I - xx^T)y\|_2^2} - \frac{(I - yy^T)xy^T(I - xx^T)}{\|(I - xx^T)y\|_2\|(I - yy^T)x\|_2}\right)\xi \\ &= \left(I - \frac{(I - xx^T)yy^T}{(1 - x^Ty)(1 + x^Ty)} - \frac{(I - yy^T)xy^T}{(1 - x^Ty)(1 + x^Ty)}\right)\xi \\ &= \left(I - \frac{(y - xx^Ty)y^T}{(1 - x^Ty)(1 + x^Ty)} - \frac{(x - yy^Tx)y^T}{(1 - x^Ty)(1 + x^Ty)}\right)\xi \\ &= \left(I - \frac{(y + x - xx^Ty - yy^Tx)y^T}{(1 - x^Ty)(1 + x^Ty)}\right)\xi \\ &= \left(I - \frac{(1 - x^Ty)(y + x)y^T}{(1 - x^Ty)(1 + x^Ty)}\right)\xi \\ &= \left(I - \frac{2(y + x)y^T}{2 + 2x^Ty}\right)\xi \\ &= \left(I - \frac{2(y + x)y^T}{\|x + y\|_2^2}\right)\xi,\end{aligned}$$

which is (10.4.4). □

10.5 The Orthogonal Group

The orthogonal group is the Stiefel manifold with $p = n$, i.e.,

$$\mathcal{O}(n) := \{X \in \mathbb{R}^{n\times n} : X^TX = I_n\}.$$

as with the sphere results we highlight the differences that arise due to the specific structure of the group.

The tangent space of $\mathcal{O}(n)$ is

$$\mathrm{T}_X\,\mathcal{O}(n) = \{X\Omega : \Omega^T = -\Omega\}. \tag{10.5.1}$$

The two metrics corresponding to an embedded submanifold and a quotient manifold are

$$g_e(U, V) = \mathrm{trace}(U^TV), \text{ and } g_q(U, V) = \mathrm{trace}(U^T(I_n - \frac{1}{2}XX^T)V)$$

where $U, V \in \mathrm{T}_X \mathcal{O}(n)$. Notice that $U^T V = U^T X X^T V$, we have

$$g_e(U, V) = 2 g_q(U, V).$$

Since the difference is scaling by a constant, they are equivalent and we choose the metric

$$g(U, V) = g_e(U, V). \tag{10.5.2}$$

By rewriting $g(U, V)$ in vector form, G_x is seen to be the identity for all $x \in \mathcal{O}(n)$.

10.5.1 Retractions

As with the Stiefel manifold, three retractions are considered. The exponential mapping is

$$\mathrm{Exp}_X(U) = X \exp(X^T U),$$

where $X \in \mathcal{O}(n)$ and $U \in \mathrm{T}_X \mathcal{O}(n)$. The retractions based on the QR and polar decompositions are

$$\begin{aligned} R_X(U) &= \mathrm{qf}(X + U), \\ R_X(U) &= (X + U)(I_p + U^T U)^{-1/2}, \end{aligned}$$

10.5.2 Vector Transports

Since the orthogonal group is a special case of the Stiefel manifold, the vector transport by projection and vector transports by differentiated retractions are (10.2.5), (10.2.6), (10.2.7) and (10.2.17) with $p = n$. However, as with the sphere, the closed form of the parallel translation is known,

$$P_\gamma^{1 \leftarrow 0} \xi = X e^{\frac{X^T \eta}{2}} X^T \xi e^{\frac{X^T \eta}{2}},$$

where γ is a geodesic from X to Y and $\gamma(0) = X, \gamma(1) = Y$, $\mathrm{Exp}_X(\eta) = Y$.

Even though, for $\mathcal{O}(n)$, the codimension, $n(n+1)/2$, and dimension, $n(n+1)/2$, are not too different using a basis of the tangent space is preferred to using a basis of the normal space since the latter requires an extra QR decomposition. The procedure for using the basis of $\mathrm{T}_X \mathcal{O}(n)$ to construct isometric vector transports is the same as that discussed in Section 10.2.2. Note however in this case, since the matrix X is square, $X_\perp$ does not exist which results in some simplification of the computational complexity. Equations (9.2.17), (9.2.18), (9.2.19) and (9.2.20) are all isometric vector transports.

10.6 The Grassmann Manifold

The Grassmann manifold, $\mathrm{Gr}(p, n)$, is the set of all p-dimensional subspaces of $\mathbb{R}^n$. There are several ways to describe it [EAS98], [AMS08]. This section uses the framework in [EAS98].

The Grassmann manifold is $\mathcal{O}(n)/(\mathcal{O}(p) \times \mathcal{O}(n-p))$. Since the quotient Stiefel manifold is $\mathcal{O}(n)/\mathcal{O}(n-p)$, the Grassmann manifold can be written also as $\mathrm{St}(p, n)/\mathfrak{G}$ where

$$\mathfrak{G} = \mathcal{O}(p) \tag{10.6.1}$$

and the group action, denoted $\cdot$, is matrix multiplication on the right, i.e., $O_1, O_2 \in \mathcal{O}(p)$,

$$O_1 \cdot O_2 = O_2 O_1 \tag{10.6.2}$$

and the group action on $\mathrm{St}(p, n)$ is also matrix multiplication on the right, i.e., $X \in \mathrm{St}(p, n)$ and $O \in \mathcal{O}(p)$,

$$O \bullet X = XO. \tag{10.6.3}$$

Therefore, the equivalence class of $X \in \mathrm{St}(p, n)$ that corresponds to an element of the $\mathrm{Gr}(p, n)$ is

$$[X] = \{XO : O \in \mathcal{O}(p)\} = X\mathcal{O}(p).$$

In order to apply Theorem 9.3.1, an element and a tangent vector of the total manifold $\mathrm{St}(p, n)$ must be expressed as a vector. Let X^v denote $\mathrm{vec}(X)$ and $\eta_{\uparrow X^v}$ denote $\mathrm{vec}(\eta_{\uparrow X})$, where $X \in \mathrm{St}(p, n)$ and $\eta_{[X]} \in \mathrm{T}_{[X]} \mathrm{Gr}(p, n)$. By the definition of Kronecker product, we obtain that the group is

$$\mathfrak{G}^v = \{O \otimes I_n | O \in \mathcal{O}(p)\}, \tag{10.6.4}$$

the group action, denoted $\cdot$, is matrix multiplication on the left, i.e., $O_1, O_2 \in \mathcal{O}(p)$,

$$(O_1 \otimes I_n) \cdot (O_2 \otimes I_n) = (O_1 \otimes I_n)(O_2 \otimes I_n) = O_1 O_2 \otimes I_n, \tag{10.6.5}$$

and the group action on $\mathrm{St}(p, n)$ is also matrix multiplication on the left, i.e., $O \in \mathcal{O}(p), X \in \mathrm{St}(p, n)$,

$$(O \otimes I_n) \bullet X^v = (O \otimes I_n) X^v = \mathrm{vec}(XO^T). \tag{10.6.6}$$

Equations (10.6.1), (10.6.2) and (10.6.3) are equivalent to (10.6.4), (10.6.5) and (10.6.6) respectively. The only difference is that the former group action is on $\mathrm{St}(p, n)$ and the latter is on $\mathrm{vec}(\mathrm{St}(p, n)) = \{\mathrm{vec}(X) | X \in \mathrm{St}(p, n)\}$.

For any $O \in \mathcal{O}(p)$, $O \otimes I_n$ defines a function $O \otimes I_n : \mathrm{vec}(\mathrm{St}(p, n)) \to \mathrm{vec}(\mathrm{St}(p, n)) : X^v \mapsto (O \otimes I_n)X^v$. It is smooth and therefore differentiable with Jacobian

$$J_{(O\otimes I_n)}(X) = (O \otimes I_n). \tag{10.6.7}$$

Hence, we can apply (iv) of Theorem 9.3.1 and obtain the relationship between horizontal lifts of a tangent vector,

$$\eta_{\uparrow_{(O\times I_n)\bullet \mathrm{vec}(X)}} = \mathrm{vec}(\eta_{\uparrow_{XO^T}}) = \mathrm{vec}(\eta_{\uparrow_X} O^T),$$

which is essentially identical to [AMS08, Proposition 3.6.1].

From [AMS08, Example 3.6.4] and $X \in \mathrm{St}(p, n)$, the horizontal space at X is

$$\mathcal{H}_X = \{X_\perp K : K \in \mathbb{R}^{(n-p)\times p}\}, \tag{10.6.8}$$

where $X_\perp$ is the same as in the definition of tangent space of $\mathrm{St}(p, n)$. The metric is

$$g_{[X]}(\eta_{[X]}, \xi_{[X]}) = g_X(\eta_{\uparrow_X}, \xi_{\uparrow_X}) = \mathrm{trace}(\eta_{\uparrow_X}^T (I - \frac{1}{2}XX^T)\xi_{\uparrow_X}) = \mathrm{trace}(\eta_{\uparrow_X}^T \xi_{\uparrow_X}), \tag{10.6.9}$$

where $\eta_{[X]}, \xi_{[X]} \in \mathrm{T}_{[X]} \mathrm{Gr}(p, n)$, $\eta_{\uparrow_X}, \xi_{\uparrow_X}$ are horizontal lifts of $\eta_{[X]}, \xi_{[X]}$ at X. By rewriting (10.6.9) in vector form, it is clear that G_X is an np by np identity matrix for all $X \in \mathrm{St}(p, n)$. This result is consistent with (iii) of Theorem 9.3.1,

$$G_X = I_{np} = (O \otimes I_n)(O \otimes I_n)^T = J_{(O\otimes I_n)}(X)^T G_{(O\otimes I_n)\bullet X} J_{(O\otimes I_n)}(X)^T.$$

A linear operator $\mathcal{A}_{[X]}$ on a tangent space $\mathrm{T}_{[X]} \mathrm{Gr}(p, n)$ at different horizontal spaces has different expressions and they satisfy the general formula (9.3.6) which in this case is

$$\mathcal{A}_{\uparrow_{(O\otimes I_n)\bullet X}} (O \otimes I_n) = (O \otimes I_n)\mathcal{A}_{\uparrow_X}.$$

10.6.1 Retractions

The exponential mapping of $\mathrm{Gr}(p, n)$ [EAS98, Theorem 2.3] is

$$\mathrm{Exp}_{[X]}(\eta_{[X]}) = [\begin{pmatrix} XV & U \end{pmatrix} \begin{pmatrix} \cos\Sigma \\ \sin\Sigma \end{pmatrix} V^T],$$

where $X \in [X]$, $\eta_{\uparrow_X}$ is the horizontal lift of $\eta_{[X]}$ at X and $\eta_{\uparrow_X} = U\Sigma V^T$ is the singular value decomposition.

The matrix representation of the exponential mapping is

$$\mathrm{Exp}_X(\eta_{\uparrow_X}) = \begin{pmatrix} XV & U \end{pmatrix} \begin{pmatrix} \cos\Sigma \\ \sin\Sigma \end{pmatrix} V^T. \tag{10.6.10}$$

Two retractions, (10.2.3) and (10.2.4), are on the total space $\mathrm{St}(p,n)$. Therefore, they are retractions of the total space of the Grassmann manifold and we restate them for completeness,

$$\bar{R}_X(\eta_X) = \mathrm{qf}(X + \eta_{\uparrow_X}), \tag{10.6.11}$$

$$\bar{R}_X(\eta_X) = (X + \eta_{\uparrow_X})(I_p + \eta_{\uparrow_X}^T \eta_{\uparrow_X})^{-1/2}. \tag{10.6.12}$$

Both satisfy (9.3.7) and therefore define two retractions of $\mathrm{Gr}(p,n)$ by (9.3.8).

Lemma 10.6.1. *The retraction* (10.6.11) *satisfies the condition* (9.3.7) *with* $\tilde{h}$ *not necessarily equal to* h *and the retraction* (10.6.12) *satisfies the condition* (9.3.7) *with* $\tilde{h} = h$.

Proof. To show (10.6.11) and (10.6.12) satisfy (9.3.7), we must show that there exists a matrix $\tilde{O} \in \mathcal{O}(p)$ such that $\bar{R}_X(\eta_{\uparrow_X})\tilde{O} = \bar{R}_{XO}(\eta_{\uparrow_X} O)$, $X \in \mathrm{St}(p,n)$, $O \in \mathcal{O}(p)$ and $\eta_{\uparrow_X} \in \mathcal{H}_X$.

Consider (10.6.11) first. We have $\bar{R}_X(\eta_{\uparrow_X}) = \mathrm{qf}(X + \eta_{\uparrow_X})$ and

$$\bar{R}_{XO}(\eta_{\uparrow_X} O) = \mathrm{qf}(XO + \eta_{\uparrow_X} O) = \mathrm{qf}((X + \eta_{\uparrow_X})O).$$

Since $\mathrm{span}(X + \eta_{\uparrow_X}) = \mathrm{span}((X + \eta_{\uparrow_X})O)$, $\bar{R}_X(\eta_{\uparrow_X})$ and $\bar{R}_{XO}(\eta_{\uparrow_X} O)$ are orthonormal bases of the same space. Therefore, there exists an orthonormal matrix, O_{qf}, such that $\bar{R}_X(\eta_{\uparrow_X})O_{\mathrm{qf}} = \bar{R}_{XO}(\eta_{\uparrow_X} O)$. Since O_{qf} is not equal to O in general, the retraction (10.6.11) satisfies (9.3.7) with $\tilde{h}$ not necessarily equal to h.

For (10.6.12), we have

$$\bar{R}_{XO}(\eta_{\uparrow_X} O) = U_1 \text{ and } \bar{R}_X(\eta_{\uparrow_X}) = U_2,$$

where both $(X + \eta_{\uparrow_X})O = U_1 P$ and $X + \eta_{\uparrow_X} = U_2 P$ are the unique polar decompositions. From the property of the polar decomposition, we have $U_2 O = U_1$. Therefore, we obtain

$$\bar{R}_X(\eta_{\uparrow_X})O = \bar{R}_{XO}(\eta_{\uparrow_X} O).$$

Since $\tilde{O}$ is equal to O, the retraction (10.6.12) satisfies (9.3.7) with $\tilde{h} = h$. □

It is also easy to verify the exponential mapping (10.6.10) satisfies (9.3.7) with $\tilde{h} = h$.

10.6.2 Vector Transports

A mapping $\bar{\mathcal{T}} : \mathcal{H}_X \to \mathcal{H}_Y$ is called a vector transport of $\mathrm{Gr}(p,n)$ if it defines a vector transport of $\mathrm{Gr}(p,n)$ by (9.3.12). The vector transport by projection is given in [AMS08],

$$\bar{\mathcal{T}}_{P_{\eta_{\uparrow X}}} \xi_{\uparrow X} = (I - YY^T)\xi_{\uparrow X}, \tag{10.6.13}$$

where $\eta_{[X]}, \xi_{[X]} \in \mathrm{Gr}(p,n)$, $\eta_{\uparrow X}, \xi_{\uparrow X}$ are horizontal lifts of $\eta_{[X]}, \xi_{[X]}$ at X, and $Y = \bar{R}_X(\eta_{\uparrow X})$. The next lemma shows the requirement of the associated retraction of the vector transport by projection.

Lemma 10.6.2. *The associated retraction of* (10.6.13) *must satisfy* (9.3.7) *with* $\tilde{h} = h$. *The retractions* (10.6.10) *and* (10.6.12) *do but retraction* (10.6.11) *does not.*

Proof. We have

$$\begin{aligned}\bar{\mathcal{T}}_{P_{\eta_{\uparrow XO}}} \xi_{\uparrow XO} &= (I - Y\tilde{O}^T\tilde{O}Y^T)\xi_{\uparrow XO} \\ &= (I - YY^T)\xi_{\uparrow X} O \\ &= \bar{\mathcal{T}}_{P_{\eta_{\uparrow X}}} \xi_{\uparrow X} O, \end{aligned} \tag{10.6.14}$$

where $O \in \mathcal{O}(p)$, $Y = \bar{R}_X(\eta_{\uparrow X})$ and $Y\tilde{O} = \bar{R}_{XO}(\eta_{\uparrow XO})$. Comparing (10.6.14) with (9.3.11) and using (10.6.7), we obtain $\tilde{O} = O$. Since the retractions (10.6.10) and (10.6.12) satisfy (9.3.7) with $\tilde{h} = h$, i.e., $\tilde{O} = O$ in this case, the vector transport by projection satisfies (9.3.11) with retractions (10.6.10) and (10.6.12), but not with (10.6.11). □

The vector transport by differentiated retraction (10.6.11) is given in the next lemma.

Lemma 10.6.3. *The vector transport by differentiated retraction* (10.6.11) *is*

$$\bar{\mathcal{T}}_{R_{\eta_{\uparrow X}}} \xi_{\uparrow X} = (I_n - YY^T)\xi_{\uparrow X}(Y^T(X + \eta_{\uparrow X}))^{-1}, \tag{10.6.15}$$

where $\eta_{[X]}, \xi_{[X]} \in \mathrm{Gr}(p,n)$, $\eta_{\uparrow X}, \xi_{\uparrow X}$ *are horizontal lifts of* $\eta_{[X]}, \xi_{[X]}$ *at* X, *and* $Y = \bar{R}_X(\eta_{\uparrow X})$.

Proof. Let $t \mapsto W(t)$ be a curve on the noncompact Stiefel manifold $\mathbb{R}_*^{n\times p}$, i.e., the set of matrix that has full columns rank. $\dot{W}(0) = V$ and let $W(t) = Y(t)P(t)$ denote the qf decomposition of $W(t)$. We have

$$\dot{W} = \dot{Y}P + Y\dot{P}. \tag{10.6.16}$$

We also have the decomposition

$$\dot{Y} = YY^T\dot{Y} + (I_n - YY^T)\dot{Y}.$$

Using the expression for the horizontal space (10.6.8), we have $Y^T\dot{Y} = 0$ which yields

$$\dot{Y} = (I_n - YY^T)\dot{Y}.$$

Multiplying (10.6.16) by $I_n - YY^T$ on the left and multiplying P^{-1} on the right, we obtain

$$(I_n - YY^T)\dot{W}P^{-1} = (I_n - YY^T)\dot{Y}.$$

Therefore, we have

$$\dot{Y} = (I_n - YY^T)\dot{W}P^{-1}. \tag{10.6.17}$$

Computing $\bar{\mathcal{T}}_{R_{\eta_{\uparrow X}}}\xi_{\uparrow X} = \frac{d}{dt}\bar{R}_X(\eta_{\uparrow X} + t\xi_{\uparrow X})|_{t=0}$ is equivalent to computing $\dot{Y}(0)$ when $W(t) = X + \eta_{\uparrow X} + t\xi_{\uparrow X}$. Therefore, we have $\dot{W} = \xi_{\uparrow X}$, $Y = Y(0) = \bar{R}_X(\eta_{\uparrow X})$ and $P = Y(0)^T W(0) = Y^T(X + \eta_{\uparrow X})$. Substituting into (10.6.17) gives the desired result. □

Even though the differentiated retraction (10.6.15) is derived from the retraction (10.6.11), it is not necessarily associated with the retraction. The next lemma shows that there is no requirement on the relationship between $\tilde{h}$ and h for the associated retraction of the vector transport (10.6.15).

Lemma 10.6.4. *The associated retraction of the vector transport* (10.6.15) *can be any retraction of* $\mathrm{Gr}(p, n)$*, e.g.,* (10.6.10)*,* (10.6.11) *and* (10.6.12)*.*

Proof. Suppose $\bar{R}_{XO}(\eta_X O) = R_X(\eta_X)\tilde{O}$ holds, e.g., $\tilde{O} = O_{\mathrm{qf}}$ for retraction (10.6.11) and $\tilde{O} = O$ for retractions (10.6.10) and (10.6.12). We have

$$\begin{aligned}\bar{\mathcal{T}}_{R_{\eta_{\uparrow XO}}}\xi_{\uparrow XO} &= (I_n - Y\tilde{O}\tilde{O}^T Y^T)\xi_{\uparrow X} O(\tilde{O}^T Y^T(XO + \eta_{\uparrow X}O))^{-1} \\ &= (I_n - YY^T)\xi_{\uparrow X}(Y^T(X + \eta_{\uparrow X}))^{-1}\tilde{O} \\ &= (\bar{\mathcal{T}}_{R_{\eta_{\uparrow X}}}\xi_{\uparrow X})\tilde{O},\end{aligned}$$

where $Y = \bar{R}_X(\eta_{\uparrow X})$. Therefore, the differentiated retraction (10.6.15) satisfies (9.3.11) with an arbitrary retraction. □

The differentiated retraction of (10.6.12) is identical to (10.6.15). This is due to the fact that (10.6.11) and (10.6.12) are the same essentially. The main difference is that they choose different representations of an equivalent class. This is shown in the following lemma.

Lemma 10.6.5. *The differentiated retraction of* (10.6.12) *is*

$$\bar{\mathcal{T}}_{R_{\eta_{\uparrow X}}} \xi_{\uparrow X} = (I_n - YY^T)\xi_{\uparrow X}(Y^T(X + \eta_{\uparrow X}))^{-1},$$

where $\eta_{[X]}, \xi_{[X]} \in \mathrm{Gr}(p,n)$, $\eta_{\uparrow X}, \xi_{\uparrow X}$ *are horizontal lifts of* $\eta_{[X]}, \xi_{[X]}$ *at* X, *and* $Y = \bar{R}_X(\eta_{\uparrow X})$.

Proof. The proof is basically the same as that in Lemma 10.6.3 and we do not repeat it here. □

The vector transports of $\mathrm{Gr}(p,n)$ given above are non-isometric. The natural isometric vector transport, the parallel translation [EAS98, Theorem 2.4]. is

$$P_{\bar{\gamma}(t)}^{1 \leftarrow 0} \xi_{\uparrow X} = \left(\begin{pmatrix} XV & U \end{pmatrix} \begin{pmatrix} -\sin \Sigma \\ \cos \Sigma \end{pmatrix} U^T + (I - UU^T) \right) \xi_{\uparrow X}$$

where $Y = \mathrm{Exp}_X \eta_{\uparrow X}$, $\eta_{\uparrow X}$ is the horizontal lift of $\eta_{[X]}$ at X, $\eta_{\uparrow X} = U \Sigma V^T$ is the singular value decomposition and $\bar{\gamma}(t)$ is a geodesic from X to Y such that $\bar{\gamma}(0) = X, \bar{\gamma}(1) = Y$, i.e.,

$$\bar{\gamma}(t) = \begin{pmatrix} XV & U \end{pmatrix} \begin{pmatrix} \cos \Sigma t \\ \sin \Sigma t \end{pmatrix} V^T.$$

The associated retraction of the parallel translation must be the exponential mapping since $P_{\bar{\gamma}(t)}^{1 \leftarrow 0} \xi_{\uparrow X}$ is only in the tangent space of $\mathrm{Exp}_X(\eta_{\uparrow X})$.

The matrix expression G_X of the metric (10.6.9) is an identity matrix. Hence, it is a positive definite matrix. Therefore, a perpendicular space of any horizontal space $\mathcal{H}_X$ based on the metric (10.6.9) is well-defined. An orthonormal basis B_X of $\mathrm{T}_X \mathcal{S}_X$ is given by

$$\{X_\perp \tilde{e}_i e_j^T, i = 1, \ldots, n-p, j = 1, \ldots, p\}, \tag{10.6.18}$$

where $(e_1, \ldots, e_p)$ is the canonical basis of $\mathbb{R}^p$, $(\tilde{e}_1, \ldots, \tilde{e}_{n-p})$ is the canonical basis of $\mathbb{R}^{n-p}$. An orthonormal basis N_X of $\mathrm{N}_X \mathcal{S}_X = (\mathcal{H}_X)_\perp$ is given by

$$\{X(e_i e_j^T), i = 1, \ldots, p, j = 1, \ldots, p\}.$$

The columns of B_X and N_X are chosen as the "vec" of the basis elements. The function constructing N_X is smooth. If $X_\perp$ is smoothly dependent on X then B_X is also a smooth function.

Given the functions constructing B_X and N_X, four mappings (9.3.17), (9.3.18), (9.3.19) and (9.3.20) can be defined and Lemma 9.3.1 used to determine whether they are vector transports or not.

From (10.6.7), we can see that $J_{O\otimes I_n}(X)$ is independent on X. In addition, noticing G_X is identity for all $X \in \mathrm{St}(p,n)$, we know (9.3.4) holds. Therefore, by (i) of Lemma 9.3.1, (9.3.17) is a vector transport. Since $G^{1/2}_{(O\otimes I_n)\bullet X^v} J_{O\otimes I_n}(X) G_X^{1/2} = J_{O\otimes I_n}(X)$ is independent of X, we also obtain (9.3.18) is a vector transport. However, it is difficult to guarantee (9.3.19) and (9.3.20) are vector transports since $J_{O\otimes I_n}(X) B_X = B_{(O\otimes I_n)\bullet X^v}$ and $J_{O\otimes I_n}(X) N_X = N_{(O\otimes I_n)\bullet X^v}$ do not hold in general for any element $X \in \mathrm{Gr}(p,n)$.

We can define a section of the Grassmann manifold to overcome the difficulties of (9.3.19) and (9.3.20). Given a $X \in \mathrm{St}(p,n)$, a section $\mathcal{S}_X$ is defined as

$$\begin{aligned}\mathcal{S}_X &= \{Y \in \mathrm{St}(p,n) | \text{there exists a } \eta_{\uparrow_X} \in \mathcal{H}_X \text{ such that } \mathrm{Exp}_X(\eta_{\uparrow_X}) = Y \text{ and } X^TY \text{ is full rank.}\}\\ &= \{Y \in \mathrm{St}(p,n) | X^TY \text{ is a full rank and symmetric matrix. }\}.\end{aligned}$$

If X^TY is not full rank, then $\mathcal{S}_X$ intersects each equivalent class more than once but in a neighborhood only once. $\mathcal{S}_X$ is therefore only a local section of the Grassmann manifold. In practice, one can randomly choose a $X_0 \in \mathrm{St}(p,n)$ and use $\mathcal{S}_{X_0}$. If a iterate X^+ makes $X_0^TX^+$ not full rank, then the section can be changed to be a new one $\mathcal{S}_{X^+}$.

In practice, it is worthwhile to try the mappings (9.3.19) and (9.3.20) directly. First, due to the smoothness of the mappings, both of them are close to a vector transport locally, i.e., when the two elements of the manifold defining the two tangent spaces are sufficiently close. Second, using a section requires some extra work and may increase computational complexity. Third, the mapping (9.3.19) is linear with respect to Y, which allows the use of the idea in Section 4.4.3 to construct a cheap retraction that satisfies the locking condition with (9.3.19) in contrast to modifying the mapping (9.3.19) based on a local section $\mathcal{S}_X$ which does not give an vector transport that is linear with respect to Y. Therefore, the mappings (9.3.19) and (9.3.20) give what can be informally called local vector transports. This idea may be worth exploiting in the later iterations of an algorithm due to savings in computational cost associated with the local vector transports. A similar phenomenon occurs when non-isometric transports are locally near isometric vector transports and allow convergence despite violating the conditions of the convergence analyses.

10.6.3 Pairs of Retraction and Isometric Vector Transport Satisfying Locking Condition

Since the vector transport (9.3.19) is linear with respect to Y for the Grassmann manifold, we can construct a retraction that satisfies the locking condition when paired with the vector transport. The derivation is similar to those in Section 10.2.3.

Let $X \in \mathrm{St}(p,n)$, $U \in \mathcal{H}_X$ and denote the unknown $R_X(tU)$ as $X(t)$. Equation (4.4.10) implies

$$\frac{d}{dt}\operatorname{vec}(X(t)) = B_{X(t)} B_X^T \operatorname{vec}(U),$$

where $B_{X(t)}$ is the orthonormal basis of $\mathcal{H}_X$ given by (10.6.18) and $X = X(0)$. Using the form of B_X, we have that

$$B_X^T \operatorname{vec}(U) = \operatorname{vec}(K_U)$$

where $K_U = X_\perp^T U$. Using the same method in Section 10.2.1 and noting the non-existence of Ω_U, we obtain an ordinary differential equation

$$\frac{d}{dt}\begin{pmatrix} X(t) & X_\perp(t) \end{pmatrix} = \begin{pmatrix} X(t) & X_\perp(t) \end{pmatrix}\begin{pmatrix} 0 & -K^T \\ K & 0 \end{pmatrix}.$$

Therefore, the desired retraction is

$$Y = R_X(U) = X(1) = \begin{pmatrix} X & X_\perp \end{pmatrix} \exp\begin{pmatrix} 0 & -K^T \\ K & 0 \end{pmatrix}\begin{pmatrix} I_p \\ 0 \end{pmatrix}. \qquad (10.6.19)$$

and $Y_\perp$ in the basis B_Y of the isometric vector transport (9.2.19) is $X_\perp(1)$. The retraction (10.6.19) is the exponential mapping (10.6.10) [EAS98, Theorem 2.3].

In general, when the retraction associated with a vector transport is the exponential mapping, as above, it does not necessarily imply that the vector transport is parallel translation. However, on the Grassmann manifold, when the associated retraction of the vector transport by parallelization is the exponential mapping and $Y_\perp$ is used in the basis B_Y, the vector transport by parallelization is the parallel translation. This can be seen easily from [EAS98, Theorem 2.4, proof 2]. Therefore, we obtain a method to compute the parallel translation of the Grassmann manifold when X, Y are given but $\mathrm{Exp}_X^{-1} Y$ is unknown.

The differentiated retraction of the retraction (10.6.19) is also straightforward to derive. Let

$$M_1 = \begin{pmatrix} 0 & -K_U^T \\ K_U & 0 \end{pmatrix} \text{ and } M_2 = \begin{pmatrix} 0 & -K_V^T \\ K_V & 0 \end{pmatrix}.$$

It follows that

$$\begin{aligned}\mathcal{T}_{R_U} V &= \begin{pmatrix} X & X_\perp \end{pmatrix} \frac{d}{dt} \exp(M_1 + tM_2)|_{t=0} \begin{pmatrix} I_p \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} X & X_\perp \end{pmatrix} Z((Z^{-1} M_2 Z) \odot \Phi) Z^{-1} \begin{pmatrix} I_p \\ 0 \end{pmatrix} \text{ (by [NH95, Theorem 4.5]),}\end{aligned}$$

where $M_1 = Z \Lambda Z^{-1}$ is the spectral decomposition, $\lambda_i = \Lambda_{ii}$, $\odot$ denotes the Hadamard product, i.e. $(A \odot B)_{ij} = A_{ij} B_{ij}$ and

$$\Phi_{ij} = \Phi_{ji} = \begin{cases} \frac{e^{\lambda_i} - e^{\lambda_j}}{\lambda_i - \lambda_j}, & \text{if } \lambda_i \neq \lambda_j; \\ e^{\lambda_i}, & \text{if } \lambda_i = \lambda_j. \end{cases}$$

10.6.4 Cotangent Vector Required by Ring and Wirth's RBFGS

Since (10.6.11) and (10.6.12) have an identical differentiated retraction, the cotangent vectors are also identical. Therefore, we consider them together. Let $\eta_{\uparrow_X}, \xi_{\uparrow_X} \in \mathcal{H}_X$ and $\zeta_{\uparrow_Y} \in \mathcal{H}_Y$, $Y = \bar{R}_X(\eta_{\uparrow_X})$. We have that

$$\begin{aligned} C(X, \eta_{\uparrow_X}, \zeta_{\uparrow_Y})[\xi_{\uparrow_X}] &= \operatorname{trace}(\zeta_{\uparrow_Y}^T (I_n - YY^T) \xi_{\uparrow_X} (Y^T (X + \eta_{\uparrow_X}))^{-1}) \\ &= \operatorname{trace}((Y^T (X + \eta_{\uparrow_X}))^{-1} \zeta_{\uparrow_Y}^T (I_n - YY^T) \xi_{\uparrow_X}). \end{aligned}$$

Therefore, the cotangent vector for the retraction (10.6.12) and (10.6.12) is

$$C(X, \eta_{\uparrow_X}, \zeta_{\uparrow_Y}) = (Y^T (X + \eta_{\uparrow_X}))^{-1} \zeta_{\uparrow_Y}^T (I_n - YY^T).$$

10.7 Complexity

In this section, we analyze the complexity of updating the Hessian approximation of Riemannian quasi-Newton algorithms and gradient sampling algorithm. When the d-dimensional intrinsic representation is used complexity depends strongly on the particular manifold. Therefore, we restrict the discussion to d-dimensional Riemannian manifolds $\mathcal{M}$ using n-dimensional vector representations.

For the assumed representation, a tangent vector is represented by an n-dimensional vector while the Hessian approximation, inverse Hessian approximation, vector transport and inverse vector transport are represented by $n \times n$ matrices. We assume the complexity of $g(\eta, \xi)$ or $\eta^\flat$ is $O(n)$.

Table 10.1: Complexity of some steps of full version and limited-memory version. – means this step is not explicit.

RBroyden family			
Full version		Limited-memory version	
Action	Complexity	Action	Complexity
get $\tilde{\mathcal{B}}_{k+1}$ from $\mathcal{B}_k$	$O(n^2)$	–	–
$\mathcal{T}_S\tilde{\mathcal{B}}\mathcal{T}_S^{-1}$	$O(n^3)$	transport s_k, y_k	$O(mn^2)$
$\mathcal{B}\eta$	$O(n^2)$	$\mathcal{B}\eta$	$O(mn)$
RTR-SR1			
Full version		Limited-memory version	
Action	Complexity	Action	Complexity
get $\tilde{\mathcal{B}}_{k+1}$ from $\mathcal{B}_k$	$O(n^2)$	–	–
$\mathcal{T}_S\tilde{\mathcal{B}}\mathcal{T}_S^{-1}$	$O(n^3)$	transport s_k, y_k	$O(mn^2)$
$\mathcal{B}\eta$	$O(n^2)$	$\mathcal{B}\eta$	$O(mn) + O(m^3)$

In Riemannian quasi-Newton methods, since $\mathcal{B}_k$ operates on a d-dimensional space, it could be written as $\mathcal{B}_k = PP^T$, where $P \in \mathbb{R}^{n\times d}$. In this way, we could decrease the complexity. Likewise, $\mathcal{T}_S$ could be written as low rank update(see Section 9.2.3 and 9.3.3). However, this can only be exploited in practice when the cost of determining the factor is not excessive. When the factor is available at an acceptable cost, the factored form saves computation compared to the complexity listed above.

Table 10.2: Complexity of some steps of RGS.

Action	Complexity
Transport grad $f(x_{kj})$ to a same tangent space	$O(mn^2)$
Solving a convex quadratic program	polynomial in n

In the Riemannian gradient sampling algorithm, even though solving a convex quadratic program is polynomial in time complexity, it can still be quite time consuming in practice. Burke, Lewis and Overton point this out and show empirically that once $n > 200$ the performance of their implementation for Euclidean problems becomes problematic. They use MOSEK to solve the convex optimization problem at each step (see http://www.mosek.com).

CHAPTER 11

EXPERIMENTS

11.1 Introduction

There are many aspects of the proposed algorithms and their performance that must be assessed empirically. We must consider the effects of: the manifold defining the constraints; the representation of manifold elements and tangent spaces; retractions, vector transports, their implementations, and their relationships to the representations of manifold elements; the smoothness of the cost function; and the properties of the problems. In this chapter, we discuss experiments that are designed to isolate and address these aspects influencing the performance of the proposed Riemannian algorithms. Empirical results for more complex application problems are presented in Chapters 12 through 14.

11.2 Test Problems

We consider four basic well-known optimization problems defined on manifolds: minimization of the Brockett cost function on the Stiefel manifold, minimization of the Rayleigh quotient on the Grassmann manifold, and two minmax problems on the sphere. The first two problems have smooth cost functions and detailed discussions can be found in [AMS08]. One minmax problem has a partly smooth Lipschitz cost function and the other has a partly smooth non-Lipschitz cost function.

The basic properties of our algorithms on an embedded manifold will be discussed using the minimization of the Brockett cost function on the Stiefel manifold with a metric endowed from the embedding Euclidean space (10.2.1). Since the Rayleigh quotient minimization problem can be defined on the Stiefel and Grassmann manifolds, it is used to compare the properties of our algorithms on a quotient representation and an embedded representation. The Lipschitz and non-Lipschitz minmax problems on the sphere are used to investigate the relationships in performance characteristics of RGS and RBFGS.

11.2.1 Brockett Cost Function Minimization on the Stiefel Manifold

The Brockett cost function is

$$f : \mathrm{St}(p,n) \to \mathbb{R} : X \mapsto \mathrm{trace}(X^T BXN),$$

where $N = \mathrm{diag}(\mu_1, \cdots, \mu_p)$ with $0 < \mu_1 < \cdots < \mu_p$ and $B \in \mathbb{R}^{n\times n}$ and $B = B^T$. The gradient with respect to the metric (10.2.1) is

$$\mathrm{grad}\, f(X) = P_X(2BXN),$$

where $\mathrm{sym}(M) = (M + M^T)/2$ and $P_X(V) = V - X\,\mathrm{sym}(X^T V)$ is the projection onto $\mathrm{T}_X\,\mathrm{St}(p,n)$. The action of the Hessian on $\eta \in \mathrm{T}_X\,\mathrm{St}(p,n)$ with respect to the metric (10.2.1) is

$$\mathrm{Hess}\, f(X)[\eta] = P_X(2B\eta N - \eta\,\mathrm{sym}(X^T(2BXN))).$$

These are in [AMS08].

Additionally, we consider the gradient with respect to the metric (10.3.1). It is

$$\mathrm{grad}\, f(X) = 2BXN - X(2BXN)^T X.$$

The action of the Hessian on $\eta \in \mathrm{T}_X\,\mathrm{St}(p,n)$ with respect to the metric (10.3.1) is

$$\begin{aligned}\mathrm{Hess}\, f(X)[\eta] = {} & 2B\eta N - X(2B\eta N)^T X - X\,\mathrm{skew}((2BXN)^T\eta) - \mathrm{skew}(\eta(2BXN)^T)X \\ & - \frac{1}{2}(I_n - XX^T)(\eta X^T(2BXN)),\end{aligned}$$

where $\mathrm{skew}(M) = (M - M^T)/2$.

It is known that the columns of a global minimizer, $X^* e_i$, are eigenvectors for the p smallest eigenvalues, λ_i, ordered so that $\lambda_1 \geq \ldots \geq \lambda_p$ [AMS08, §4.8].

11.2.2 Rayleigh Quotient Minimization on the Grassmann Manifold

The Rayleigh quotient is

$$f : \mathrm{Gr}(p,n) \to \mathbb{R} : [X] \mapsto \mathrm{trace}(X^T CX),$$

where C is a n by n matrix and not necessary positive-definite. The gradient with respect to the metric (10.6.9) satisfies

$$(\mathrm{grad}\, f([X]))_{\uparrow X} = P_X(2CX),$$

where $P_X(V) = (I_n - XX^T)V$ is the projection onto $\mathcal{H}_X$. The action of the Hessian on $\eta \in \mathrm{T}_{[X]} \mathrm{Gr}(p, n)$ is

$$(\operatorname{Hess} f([X])[\eta])_{\uparrow X} = (\operatorname{Hess} f([X]))_{\uparrow X}[\eta_{\uparrow X}] = P_X(2C\eta_{\uparrow X} - \eta_{\uparrow X} X^T(2CX)).$$

The minimizer is a space such that the eigenvectors of p smallest eigenvalues form an orthonormal basis of the space.

11.2.3 Lipschitz Minmax Problem on the Sphere

The cost function is

$$f : \mathbb{S}^{n-1} \to \mathbb{R} : x \mapsto \|x\|_\infty = \max(|x_1|, \dots, |x_n|).$$

This is a Lipschitz continuous function defined on the sphere. The gradient with respect to the metric (10.4.1) is

$$\operatorname{grad} f(x) = P_x(v), v = \begin{cases} \operatorname{sign}(x_i), & \text{where } |x_i| \text{ is the largest;} \\ 0, & \text{otherwise,} \end{cases}$$

where $P_x(v) = (I_n - xx^T)v$. If x has more than one maximal magnitude component $|x_i|$, then $\operatorname{grad} f(x)$ does not exist. The minimizer is a vector where all components have magnitude $1/\sqrt{n}$.

11.2.4 Non-Lipschitz Minmax Problem on the Sphere

The cost function is

$$f : \mathbb{S}^{n-1} \to \mathbb{R} : x \mapsto \|x\|_\infty = \max(|x_1 - \frac{1}{\sqrt{n}}|^{\frac{1}{3}}, \dots, |x_n - \frac{1}{\sqrt{n}}|^{\frac{1}{3}}).$$

The gradient with respect to the metric (10.4.1) is

$$\operatorname{grad} f(x) = P_x(v), v = \begin{cases} \frac{1}{3}\operatorname{sign}(x_i - \frac{1}{\sqrt{n}})|x_i - \frac{1}{\sqrt{n}}|^{-\frac{2}{3}}, & \text{where } |x_i - \frac{1}{\sqrt{n}}|^{\frac{1}{3}} \text{ is the largest;} \\ 0, & \text{otherwise.} \end{cases}$$

If x has more than one maximal magnitude component $|x_i - \frac{1}{\sqrt{n}}|^{\frac{1}{3}}$, then $\operatorname{grad} f(x)$ does not exist. The minimizer is a vector where all components have magnitude $1/\sqrt{n}$. If x approaches x^*, then the norm of the gradient at x goes to infinity. Therefore, this function is non-Lipschitz continuous at the minimizer.

11.3 Notation, Algorithm Parameters and Test Data Parameters

Ten algorithms are used in the experiments in this chapter. Six are combined with a line search algorithm: RBFGS using an inverse Hessian approximation, the restricted RBroyden algorithm using an inverse Hessian approximation and a problem specific $\tilde{\phi}_k$, the Davidon's update RBroyden algorithm using an inverse Hessian approximation, limited-memory RBFGS (LRBFGS), Riemannian steepest descent with line search (RSD) and RGS. Four of them are combined with a trust region: RTR-SR1, LRTR-SR1, Riemannian trust region with steepest descent (RTR-SD), and RTR-Newton [Bak08].

The line search algorithm used with RBroyden family methods is [DS83, Algorithm A6.3.1mod] for optimizing smooth functions and, is given as Algorithm 7 after appropriate modifications for partly smooth functions. The line search algorithm in RGS is the one described in Algorithm 6. The constants c_1 and c_2 in the Wolfe conditions are 1e-4 and 0.999 respectively.

When the n-dimensional embedded representation of a d-dimensional manifold is used, the system $\mathcal{B}_k \eta_k = -\operatorname{grad} f(x_k)$ may have multiple solutions. To solve $\mathcal{B}_k \eta_k = -\operatorname{grad} f(x_k)$ such that $\eta_k \in \mathrm{T}_{x_k} S^{n-1}$, we add the constraints $x_k^T \eta_k = 0$ for the sphere and the Grassmann manifold, $x_k^T \eta_k + \eta_k^T x_k^T = 0$ for the Stiefel manifold and the orthogonal group, to the system $\mathcal{B}_k \eta_k = -\operatorname{grad} f(x_k)$. The QR decomposition is used to find η_k. It is not difficult to show there is a unique solution η_k.

For RGS, when random tangent vectors in a given $\mathrm{T}_x \mathbb{S}^{n-1}$ are required, a random orthonormal basis of $\mathrm{T}_x \mathbb{S}^{n-1}$ is generated. Each required random tangent vector is produced by generating a random vector of coefficients in a unit box $[-0.5, 0.5]^{n-1}$ and computing the associated linear combination of the basis vectors. The number of samples in RGS and J for RBFGS (see Section 7.3.2) are both ceil($1.3d+5$). The constants τ_x and τ_d in Section 7.3.2 are 1e-5 and 1e-6 respectively. The initial sampling radius ϵ_0, sampling radius reduction factor μ, initial optimality tolerance ν_0, optimality tolerance reduction factor θ, tolerance τ_ν and Armijo parameter c_1 in RGS are 1e-3, 0.1, 1e-3, 0.1, 1e-5 and 0 respectively .

The trust region inner iteration algorithm is the truncated CG inner iteration in [AMS08, §7.3.2]. The θ, κ parameters in the inner iteration stopping criterion [AMS08, (7.10)] are set to 0.1, 0.9 respectively for RTR-SR1 and LRTR-SR1 and to 1, 0.1 respectively for RTR-Newton. The constants τ_1 and τ_2 in trust region are 0.25 and 2 respectively. The initial radius Δ_0 is 1, c in RTR-SR1 and LRTR-SR1 is 0.1, and ν is the square root of machine epsilon.

Table 11.1: Notation for reporting the experimental results.

gf_0	Riemannian metric value of the initial gradient
gf_f	Riemannian metric value of the final gradient
$iter$	number of iterations
nf	number of function evaluations
ng	number of gradient evaluations
nH	number of operations of the form $\mathcal{H}\eta$
nV	number of vector transports
nR	number of retraction evaluations
t	average time (seconds)

The comparisons are performed in Matlab 7.0.0 on a 32 bit Windows platform with 2.4 GHz CPU (T8300).

Unless otherwise indicated in the description of the experiments, the following test data parameters are used. The problems are defined by setting $B = R_1 + R_1^T$ and $C = R_2 + R_2^T$ where the elements of R_1 and R_2 are drawn from the standard normal distribution using Matlab's RANDN with seed 1, N is a diagonal matrix whose diagonal elements are integers from 1 to p, i.e., $N = \mathrm{diag}(1, 2, \dots, p)$. The initial iterate X_0 is given by applying Matlab's function ORTH to a matrix whose elements are drawn from the standard normal distribution using Matlab's RANDN with seed 1. The identity is used as the initial Hessian inverse approximation. The stopping criterion requires that the ratio of the norm of final gradient and the norm of initial gradient is less than 10^{-6}.

The methods of constructing an isometric vector transport are discussed in Section 9. The Matlab's function ODE45 with "RelTol" and "AbsTol" both set to 1e-5 is used to solve the ODE of the parallel transport of the Stiefel manifold.

To obtain sufficiently stable timing results, an average time is taken of several runs with identical parameters for a total runtime of at least 1 minute. The notation used when reporting the experimental results is given in Table 11.1.

There are some relationships among the operations. For RBroyden family methods $nH = 2(iter - 1)$ holds; for RTR-SR1 we have $iter = nf = ng = nv + 1 = nR + 1$; for LRTR-SR1 we have $iter = nf = ng = nR + 1$; for RTR-SD $iter = nf = nR + 1$; nH does not appear in LRBFGS, RSD, RTR-SR1, RTR-SD and RGS; and, finally, nV does not appear in RTR-SD.

11.4 Results and Conclusions

11.4.1 Performance for Different ϕ in the RBroyden Family

We use the inverse Hessian approximation update for the RBroyden family algorithms

$$\mathcal{H}_{k+1} = \tilde{\mathcal{H}}_k - \frac{\tilde{\mathcal{H}}_k y_k (\tilde{\mathcal{H}}_k^* y_k)^\flat}{(\tilde{\mathcal{H}}_k^* y_k)^\flat y_k} + \frac{s_k s_k^\flat}{s_k^\flat y_k} + \tilde{\phi}_k g(y_k, \tilde{\mathcal{H}}_k y_k) u_k u_k^\flat,$$

where

$$u_k = \frac{s_k}{g(s_k, y_k)} - \frac{\tilde{\mathcal{H}}_k y_k}{g(y_k, \tilde{\mathcal{H}}_k y_k)},$$

are tested with $\tilde{\phi}_k = \tilde{\phi} = 1.0, 0.8,\ 0.6,\ 0.4,\ 0.2,\ 0.1,\ 0.01,\ 0$ and with a variable $\tilde{\phi}_k$ set to Davidon's value $\tilde{\phi}_k^D$. The inverse Hessian approximation update tends to be preferred to the Hessian approximation update since it avoids solving a linear system.

Most of the existing literature investigates the effects of the coefficient ϕ_k in the Hessian approximation update formula. In [BNY87], Byrd et al. provide empirical evidence that, in a Euclidean space, the ability to correct eigenvalues of the Hessian approximation that are much larger than the eigenvalues of the true Hessian degrades for larger ϕ values. Our experiments show the same trend on manifolds (see Table 11.2) and RBFGS is seen to be the best at such a correction among the restricted RBroyden family methods.

Strategies for choosing ϕ_k and allowing it to be outside $[0, 1]$ have been investigated. Davidon [Dav75] defines an update for ϕ_k by minimizing the condition number of $\mathcal{B}_k^{-1}\mathcal{B}_{k+1}$. We have generalized this update to Riemannian manifolds for both the Hessian approximation and inverse Hessian approximation forms to obtain

$$\phi_k^D = \begin{cases} \frac{g(y_k,s_k)(g(y_k,\tilde{\mathcal{B}}_k^{-1}y_k)-g(y_k,s_k))}{g(s_k,\tilde{\mathcal{B}}_k s_k)g(y_k,\tilde{\mathcal{B}}_k^{-1}y_k)-g(y_k,s_k)^2}, & \text{if } g(y_k, s_k) \leq \frac{2g(s_k,\tilde{\mathcal{B}}_k s_k)g(y_k,\tilde{\mathcal{B}}_k^{-1}y_k)}{g(s_k,\tilde{\mathcal{B}}_k s_k)+g(y_k,\tilde{\mathcal{B}}_k^{-1}y_k)}; \\ \frac{g(y_k,s_k)}{g(y_k,s_k)-g(s_k,\tilde{\mathcal{B}}_k s_k)}, & \text{otherwise} \end{cases}$$

$$\tilde{\phi}_k^D = \begin{cases} \frac{g(y_k,s_k)(g(s_k,\tilde{\mathcal{H}}_k^{-1}y_k)-g(y_k,s_k))}{g(y_k,\tilde{\mathcal{H}}_k y_k)g(s_k,\tilde{\mathcal{H}}_k^{-1}s_k)-g(y_k,s_k)^2}, & \text{if } g(y_k, s_k) \leq \frac{2g(s_k,\tilde{\mathcal{H}}_k^{-1}s_k)g(y_k,\tilde{\mathcal{H}}_k y_k)}{g(s_k,\tilde{\mathcal{H}}_k^{-1}s_k)+g(y_k,\tilde{\mathcal{H}}_k y_k)}; \\ \frac{g(y_k,s_k)}{g(y_k,s_k)-g(y_k,\tilde{\mathcal{H}}_k y_k)}, & \text{otherwise} \end{cases}.$$

When the "if" conditions are satisfied in these definitions the Hessian and inverse Hessian approximations are symmetric positive definite. The "otherwise" clauses in the definitions correspond to the two forms of the Riemannian SR1 method (see Chapter 3 and [HAG13]).

Byrd et al. [BLN92] use negative values of ϕ to improve the performance of the Hessian approximation form. However, their experiments require solving a linear system to find $z_k = \operatorname{Hess} f(x_k)^{-1} v_k$. Their purpose was, of course, to demonstrate a theoretical value of ϕ_k and not to recommend the specific form for computation which by involving the Hessian is inconsistent with the goal of quasi-Newton methods. In the Riemannian setting, the action of the Hessian is often known rather than the Hessian itself, i.e., given $\eta \in \mathrm{T}_x \mathcal{M}$, $\operatorname{Hess} f(x)[\eta]$ is known. So z_k could be approximated by applying a few steps of an iterative method such as CG to the system of equations. Also, the Hessian could be recovered given a basis for $\mathrm{T}_x \mathcal{M}$ and the linear system solved but this is an excessive amount of work. Therefore, we test only the generalization of Davidon's update, $\tilde{\phi}_k^D$.

Since we use the inverse Hessian approximation update, $\tilde{\phi}_k \equiv 1$ corresponds to RBFGS and $\tilde{\phi}_k \equiv 0$ corresponds to RDFP. Also note we are testing the restricted RBroyden family since $0 \leq \tilde{\phi}_k \leq 1$ implies $0 \leq \phi_k \leq 1$. The parameters n and p are chosen to be 5 and 2 respectively. In [BNY87], Byrd et al. set the initial Hessian approximation $\mathcal{B}_0$ $\operatorname{diag}(1,\ 1,\ \dots\ ,\ 1,\ \lambda_2, \lambda_1) \in \mathbb{R}^{n \times n}$ to demonstrate the correction properties of the different members of the Broyden family. Similarly, to show the differences among RBroyden family with different ϕ_k, in our experiments the initial inverse Hessian approximation $\mathcal{H}_0$ is set to $\operatorname{diag}(1,\ 1,\ \dots\ ,\ 1,\ 1/50,\ 1/10000) \in \mathbb{R}^{d \times d}$.

The intrinsic dimension representation is used for a tangent vector. The retraction is chosen to be (10.2.16) and the vector transport is defined by parallelization.

The results in Table 11.2 and Figure 11.1 show trends typically observed in our experiments with the Brockett cost function and others. There is a clear preference in performance for choosing the constant $\tilde{\phi}$ near 1.0 to yield RBFGS or a nearby method. For variable $\tilde{\phi}_k$, Davidon's update performs somewhat better than RBFGS for the Brockett cost function and is usually comparable to RBFGS on other cost functions. The problem of choosing $\tilde{\phi}_k$ or ϕ_k is still an open question in Riemannian optimization research.

11.4.2 Retractions and Vector Transports for RBFGS

In this section, we show the results of RBFGS when using different pairs of retractions and vector transports that satisfy the locking condition. Table 11.3 shows the pairs that are tested and the notation used. Besides using the exponential mapping and parallel translation, we consider the qf retraction and the corresponding isometric vector transports that satisfies the locking condition with

Table 11.2: Comparison of RBroyden family for $\tilde{\phi}_k^D$ and several constant $\tilde{\phi}_k$. The subscript $-k$ indicates a scale of 10^{-k}.

$\tilde{\phi}_k$	$\tilde{\phi}_k^D$	1.0	0.8	0.6	0.4	0.2	0.1	0.01	0
$iter$	27	30	31	32	35	44	56	166	320
nf	29	38	38	38	42	49	60	168	322
ng	27	30	31	32	35	44	56	166	320
nH	52	58	60	62	68	86	110	330	638
nV	78	87	90	93	102	129	165	495	957
nR	28	37	37	37	41	48	59	167	321
gf_f	7.97_{-7}	3.05_{-6}	2.74_{-6}	5.51_{-6}	7.09_{-6}	6.14_{-6}	5.80_{-6}	7.87_{-7}	4.56_{-6}
gf_f/gf_0	8.58_{-8}	3.28_{-7}	2.96_{-7}	5.94_{-7}	7.64_{-7}	6.61_{-7}	6.26_{-7}	8.48_{-8}	4.92_{-7}
t	1.66_{-2}	1.81_{-2}	1.82_{-2}	1.83_{-2}	1.97_{-2}	2.48_{-2}	3.11_{-2}	9.06_{-2}	1.74_{-1}

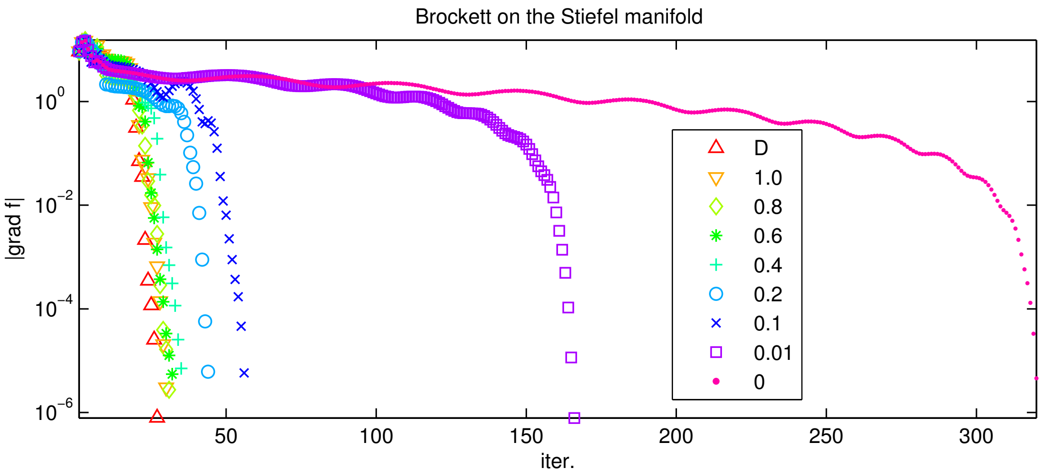


Figure 11.1: Comparison of RBroyden family for $\tilde{\phi}_k^D$ and several constant $\tilde{\phi}_k$.

Table 11.3: Notations of retractions and vector transports.

EP	Exponential mapping (10.2.2)	parallel translation (10.2.11)
QfDT	qf retraction (10.2.3)	vector transport (4.4.2) with $\mathcal{T}_I$ be vector transport by direction rotation based on tangent space(9.2.18)
QfDN	qf retraction (10.2.3)	vector transport (4.4.2) with $\mathcal{T}_I$ be vector transport by direction rotation based on normal space(9.2.17)
QfP	qf retraction (10.2.3)	vector transport (4.4.2) with $\mathcal{T}_I$ be vector transport by parallelization (9.2.19)
QfR	qf retraction (10.2.3)	vector transport (4.4.2) with $\mathcal{T}_I$ be vector transport by rigging (9.2.20)
QfC	qf retraction (10.2.3)	vector transport by constructing (4.4.9)
QfInC	qf retraction (10.2.3)	vector transport by constructing (4.4.9) using intrinsic representation
CPIn	retraction (10.2.16)	vector transport by parallelization (9.2.19) using intrinsic representation

qf. In Section 10.2, we also discuss an another retraction based on the polar decomposition (10.2.4). We do not show the results of this retraction due to two reasons. First, the polar decomposition is more expensive than the QR decomposition. Second, finding the Ω in the differentiated retraction (10.2.7) requires significant extra work solving a large linear system.

The parameters (p, n) are taken as $(4, 12)$ and $(8, 12)$. Table 11.4 and Figure 11.2 show the results of the experiments.

Unsurprisingly, RBFGS with EP takes much more time than others due to the high cost of the parallel translation even though the number of iterations are not too different from the other algorithms. RBFGS with QfDT and RBFGS with QfDN have exactly the same number of iterations since the vector transports are theoretically equivalent (see Lemma 9.2.3). In the experiments, as long as the numerical behaviors of the two vector transports are close, the numbers of iterations they require are identical. However, their computation times of are usually different. For example, when $(p, n) = (4, 12)$, RBFGS with vector transport based on the tangent space requires more computation time than RBFGS with the vector transport based on the normal space and the performance reverses when $(p, n) = (8, 12)$. This phenomenon illustrates the discussions in Section 9.2.3, i.e., a smaller ratio of codimension and dimension of the manifold implies a less computationally costly vector transport based on the tangent space and a more computationally costly vector transport based on the normal space.

RBFGS with QfP and RBFGS with QfR require less computation time than RBFGS with QfDT and RBFGS with QfDN respectively. The reason is that the vector transports by direct rotation are more expensive when compared with the vector transports by rigging or parallelization since the vector transports by direct rotation require the singular value decomposition, especially when both the dimension and codimension of a manifold are not sufficiently small in the required the singular value decomposition.

RBFGS with QfP, RBFGS with QfC and RBFGS with QfInC have identical numbers of iterations. The difference between the vector transports used in QfC and QfInC is only in their implementations since they are identical theoretically. Noting that there is no difference between doing the Householder reflection in intrinsic representation and in embedding space, the vector transport in QfP is equivalent to the other two theoretically. The only differences are the computational implementation. Since working on the intrinsic dimension avoids many expensive operations, RBFGS with QfInC has the smallest computational cost of the three.

RBFGS with CPIn is the fastest in terms of number of iterations and time. One reason is that the locking condition is satisfied by definition and the computations of differentiated retraction along a direction and Householder reflection are avoided. In addition, the number of iterations is the smallest or the second smallest one in this example. This pair of retraction and vector transport is preferred and is used in other comparisons of this chapter.

11.4.3 Comparison of RBFGS and Ring and Wirth's Algorithm

The retraction and vector transport used for our RBFGS are (10.2.16) and vector transport by parallelization which are shown, in Section 11.4.2, to be preferred. For Ring and Wirth's RBFGS (RW), retractions (10.2.3) and (10.2.16) are used. As we discussed in Section 10.2.3, these two retractions have relatively efficient computational forms of cotangent vector. The vector transport for RW is chosen to be by parallelization. The intrinsic representation for a tangent vector is used for both algorithms.

Table 11.5 contains the results for the Brockett cost function with multiple sizes of the Stiefel manifold for the efficient RW algorithm and RBFGS. Since the number of iterations among RBFGS and RW with different retractions are not significantly different, the computational time depends

Table 11.4: Comparison of retraction and vector transports for RBFGS. The subscript $-k$ indicates a scale of 10^{-k}.

p, n		RBFGS							
		EP	QfDT	QfDN	QfP	QfR	QfC	QfInC	CPIn
4, 12	$iter$	59	56	56	73	55	73	73	49
	nf	134	131	131	151	130	151	151	112
	ng	59	56	56	73	55	73	73	49
	nH	116	110	110	144	108	144	144	96
	nV	174	165	165	216	162	216	216	144
	nR	133	130	130	150	129	150	150	111
	gf_f	1.20_{-5}	4.65_{-6}	4.65_{-6}	1.71_{-5}	7.13_{-6}	1.71_{-5}	1.71_{-5}	1.56_{-5}
	gf_f/gf_0	3.38_{-7}	1.30_{-7}	1.30_{-7}	4.80_{-7}	2.00_{-7}	4.80_{-7}	4.80_{-7}	4.36_{-7}
	t	5.83_{1}	4.43_{-1}	3.27_{-1}	3.82_{-1}	2.90_{-1}	3.73_{-1}	1.38_{-1}	4.79_{-2}
8, 12	$iter$	68	74	74	84	82	84	84	72
	nf	179	188	188	209	204	209	209	182
	ng	68	74	74	84	82	84	84	72
	nH	134	146	146	166	162	166	166	142
	nV	201	219	219	249	243	249	249	213
	nR	178	187	187	208	203	208	208	181
	gf_f	7.03_{-5}	2.81_{-5}	2.81_{-5}	4.98_{-5}	7.39_{-5}	4.98_{-5}	4.98_{-5}	3.73_{-5}
	gf_f/gf_0	8.00_{-7}	3.20_{-7}	3.20_{-7}	5.66_{-7}	8.41_{-7}	5.66_{-7}	5.66_{-7}	4.24_{-7}
	t	1.55_{2}	1.62	1.92	9.54_{-1}	1.59	9.37_{-1}	2.68_{-1}	9.44_{-2}

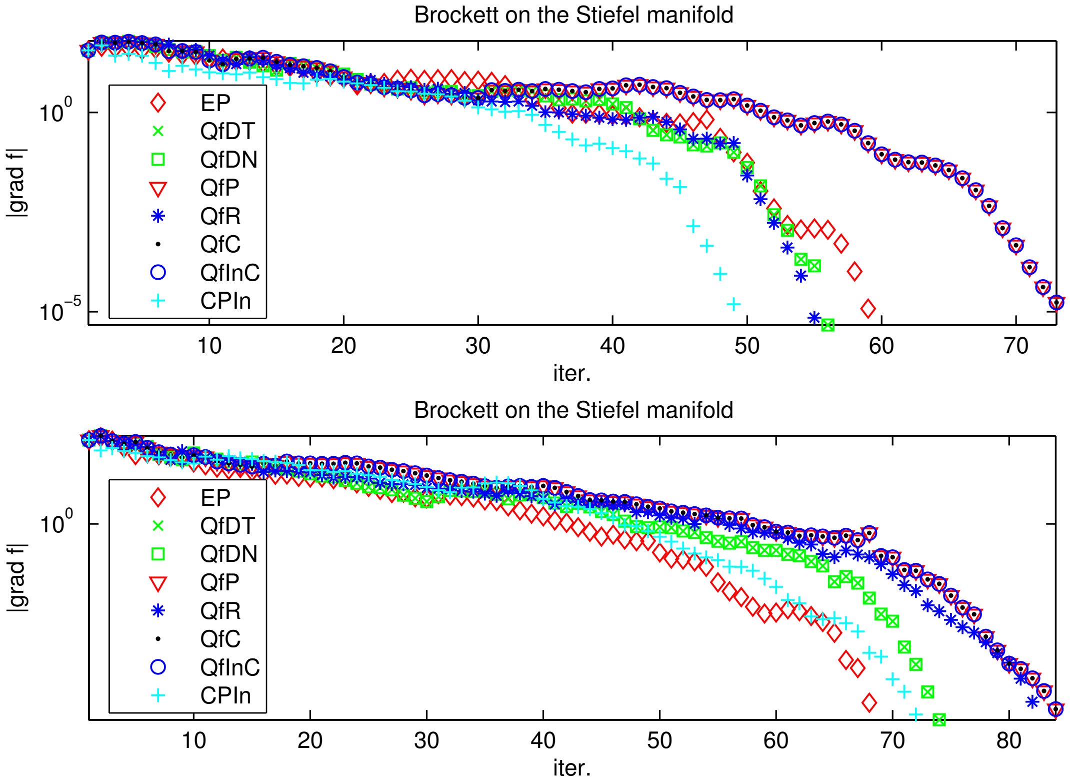


Figure 11.2: Comparison of retraction and vector transports for RBFGS. The top figure corresponds to $(p, n) = (4, 12)$ and the bottom figure corresponds to $(p, n) = (8, 12)$.

Table 11.5: Comparison of RBFGS and RW. RW1 and RW2 denote RW with retractions (10.2.16) and (10.2.3) respectively. The subscript $-k$ indicates a scale of 10^{-k}.

$(n,\ p)$	(12, 6)			(12, 12)			(24, 12)		
method	RBFGS	RW1	RW2	RBFGS	RW1	RW2	RBFGS	RW1	RW2
$iter$	61	62	63	78	74	80	186	207	217
nf	154	155	161	202	182	194	597	630	660
ng	61	62	63	79	74	80	186	207	217
nV	180	122	124	232	146	158	555	412	432
nR	153	154	160	201	181	193	596	629	659
gf_f	5.19_{-5}	3.41_{-5}	4.99_{-5}	4.90_{-5}	7.74_{-5}	3.46_{-5}	2.15_{-4}	1.85_{-4}	2.09_{-4}
gf_f/gf_0	7.60_{-7}	5.00_{-7}	7.31_{-7}	6.18_{-7}	9.75_{-7}	4.36_{-7}	8.02_{-7}	6.88_{-7}	7.79_{-7}
t	6.49_{-2}	1.29_{-1}	9.77_{-2}	1.19_{-1}	1.79_{-1}	1.57_{-1}	1.28	2.51	1.58

on the cost of each iteration. Because the retraction (10.2.16) and the corresponding cotangent vector (10.2.20) are more expensive than the retraction (10.2.3) and the corresponding cotangent vector (10.2.18), it is unsurprising that RW with retraction (10.2.16) takes more time than RW with retraction (10.2.3). This phenomenon is illustrated in Figure 11.3. Even though RW with retraction (10.2.3) has relatively efficient components, the benefit of avoiding differentiated retraction is seen in the time advantage of RBFGS for all three problem sizes. Figure 11.3 shows that RBFGS has a time advantage regardless of the required accuracy.

The noticeable increase of the computation time on the largest problem indicates that the dense matrix computations are beginning to mask the effects of other algorithmic choices. This motivates a comparison with the LRBFGS method intended to limit the use of dense matrices with the full dimension of the problem. Comparisons of LRBFGS and RBFGS for problems of moderate sizes are shown in the next section.

11.4.4 Comparison of LRBFGS and RBFGS

The performance results for RBFGS and LRBFGS with different values of the parameter m are given in Table 11.6 and Figure 11.4 for the Brockett cost function with $n = p = 16$. As expected, the number of iterations required by LRBFGS to achieve the required reduction in the norm of the gradient comparable to RBFGS decreases as m increases but remains higher than the number required by RBFGS. The benefit of LRBFGS is seen from Figure 11.4 in computation times that are better or similar to that of RBFGS for all tested values of m when a high accuracy is not

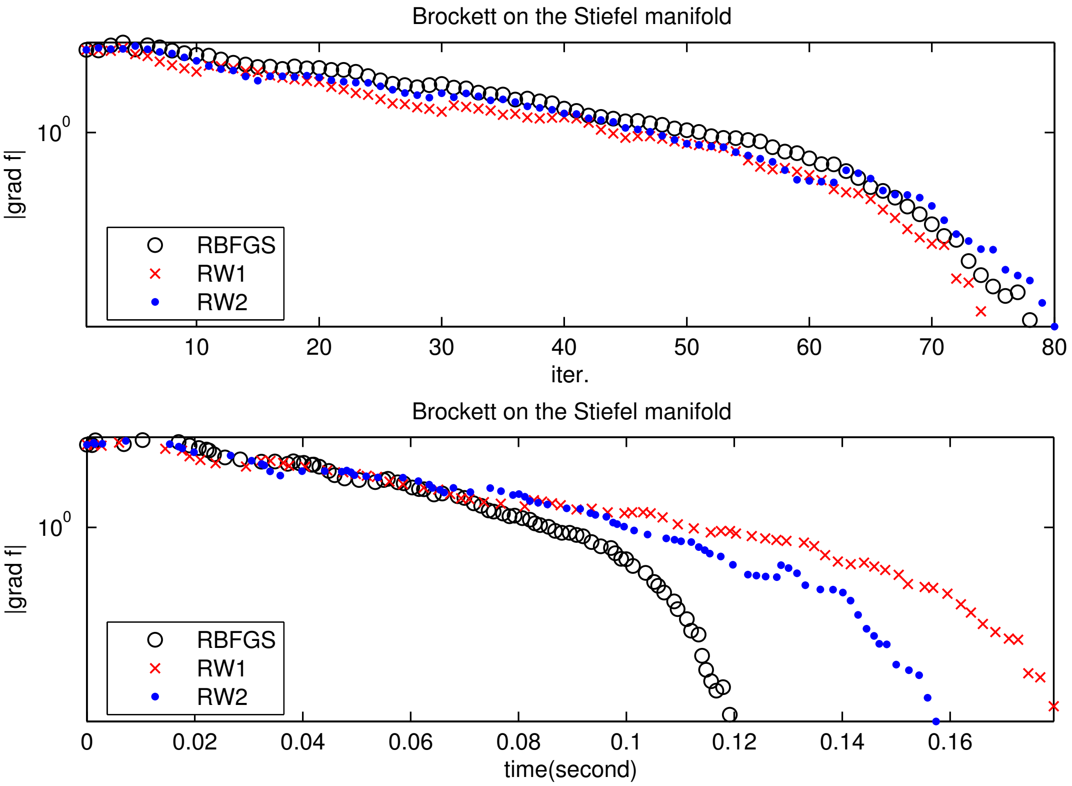


Figure 11.3: Comparison of RBFGS and RW for $(p, n) = (12, 12)$. RW1 and RW2 denote RW with retractions (10.2.16) and (10.2.3) respectively. The top figure is the results of *iter* versus $|gradf|$ and the bottom one is the results of *time* versus $|gradf|$.

Table 11.6: Comparison of LRBFGS and RBFGS. The subscript $-k$ indicates a scale of 10^{-k}.

method	RBFGS	LRBFGS					
m		1	2	4	8	16	32
$iter$	117	245	198	184	160	159	147
nf	330	272	215	200	172	174	160
ng	117	245	198	184	160	159	147
nV	348	732	983	1635	2647	4974	8498
nR	329	271	214	199	171	173	159
gf_f	1.30_{-4}	1.32_{-4}	1.10_{-4}	1.20_{-4}	1.47_{-4}	1.29_{-4}	9.51_{-5}
gf_f/gf_0	8.82_{-7}	8.96_{-7}	7.48_{-7}	8.15_{-7}	9.95_{-7}	8.72_{-7}	6.44_{-7}
t	2.10_{-1}	1.81_{-1}	1.51_{-1}	1.64_{-1}	1.83_{-1}	2.56_{-1}	3.55_{-1}

required. Even though a high accuracy is required, LRBFGS with $m \leq 8$ still shows advantages in computation time, which clearly indicates that, for this range of m, the approximation of the inverse of the Hessian is of suitable quality in LRBFGS so that the number of less complex iterations is kept sufficiently small to solve the problem in an efficient manner. The advantage is lost, as expected, once m becomes too large for the size of the given problem. In practice, for moderately sized problems, exploiting the potential benefits of LRBFGS requires an efficient method of choosing m which depends strongly on the problem. The results are encouraging in the sense of potential for problems large enough to preclude the use of RW, RBFGS, or other RBroyden family members.

11.4.5 Locking Condition and Isometry of Vector Transport in RBFGS

In order for RBFGS to be well-defined, the vector transport needs to be isometric and satisfy the locking condition. In this section, we investigate what happens when these conditions are not necessarily satisfied in our framework of RBFGS. Three vector transports are tested. The first is a non-isometric vector transport that does not satisfy the locking condition, i.e., the vector transport by projection (10.2.5). The second is a non-isometric vector transport satisfying the locking condition, i.e., vector transport (4.4.2) with $\mathcal{T}_I$ be the vector transport by projection (10.2.5). The third is an isometric vector transport without the locking condition, i.e., the vector transport by parallelization (9.2.19). The retractions for all transports are the qf retraction. For completeness, the results of using proposed retraction (10.2.16) and vector transport (9.2.19) that satisfy the isometry constraint and locking condition are included in the results.

The parameter (p, n) is chosen to be $(6, 12)$. The stopping criterion requires the ratio of the

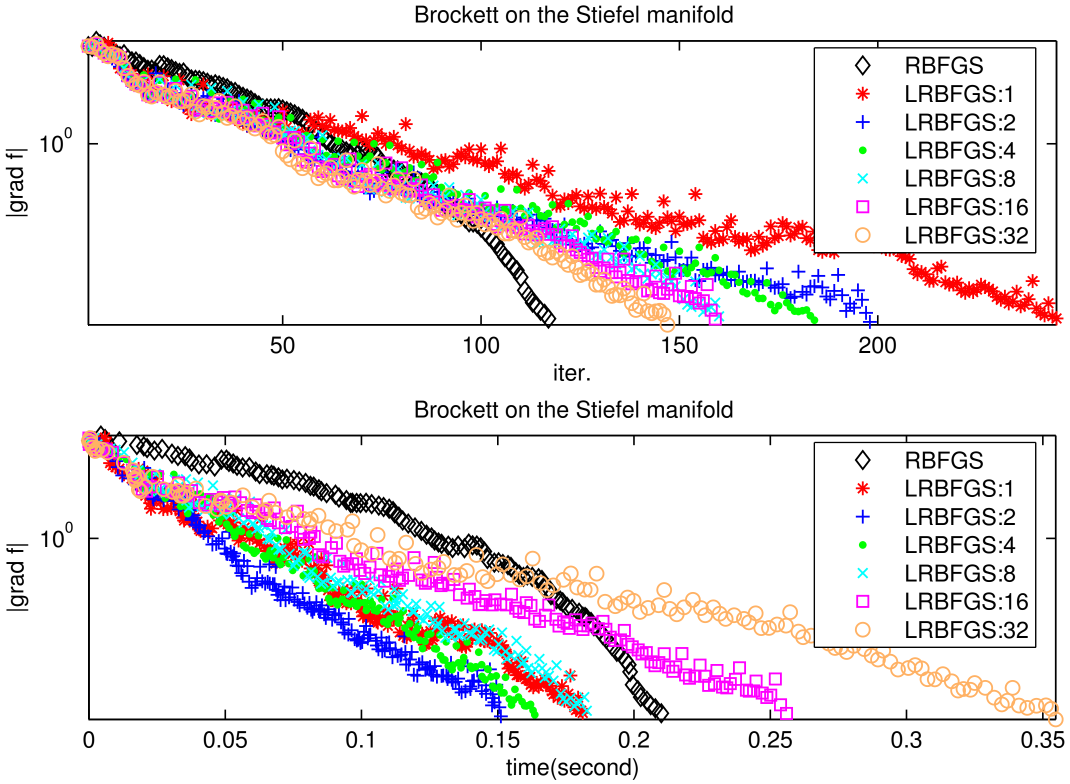


Figure 11.4: Comparison of RBFGS and LRBFGS. The top figure is the results of *iter* versus $|gradf|$ and the bottom one is the results of *time* versus $|gradf|$.

Table 11.7: The numbers of successful runs of RBFGS with different retractions and vector transports, where "a successful run" mean reaching the required accuracy.

δ_o	1	10^{-1}	10^{-2}
nonIso-nonlocking	0	84	100
nonIso-locking	0	88	100
Iso-nonlocking	80	99	100
Iso-locking	100	100	100

norm of final gradient and the norm of initial gradient to be less than 10^{-3}. The matrix B is chosen to be $Q \operatorname{diag}(1, 2, \dots, n) Q^T$ where Q is given by applying Matlab's function ORTH to an n by n matrix whose elements are drawn from the standard normal distribution. The initial iterate X_0 is given by $[q_p, q_{p-1}, \dots, q_1] + \delta_0 R$ where q_i is the i-th column of the matrix Q, the elements of $R \in \mathbb{R}^{n \times n}$ are drawn from the standard normal distribution and δ_0 is specified in Table 11.7.

Table 11.7 shows the numbers of runs that reach the required accuracy among 100 using 1, 2, ..., 100 as the random number generator seed. The reason of the failures of RBFGS in the experiments is that the line search fails due to the search direction not being a descent direction. It is more likely for RBFGS with a non-isometric vector transport to fail when the initial iterate is not close to the minimizer. Even though the locking condition is not satisfied, the RBFGS still works in a quite high probability as long as the vector transport is isometric. The choice of a vector transport reduces in importance as the initial iterate is chosen closer to the minimizer. In particular, when δ_0 is 10^{-2}, all runs of RBFGS with all vector transports reach the required accuracy. This phenomenon is explained by Lemma 4.3.6 that proves all vector transports are close to each locally, i.e., when transporting a tangent vector in $\mathrm{T}_{x_k} \mathcal{M}$ to a fairly close $\mathrm{T}_{x_{k+1}} \mathcal{M}$. Early in the iteration, these tangent spaces will be far apart for algorithms that are converging satisfactorily and the vector transports' behaviors will be different. Nearer the minimizer by necessity the distance between successive iterates is much smaller and the transports' behaviors begin to look very much alike lessening the importance of the satisfying the two conditions. This implies that there is further work to do understanding the effectiveness of non-isometric vector transports and reducing even more the influence of the differentiated retraction.

Table 11.8: Notations of retractions and vector transports.

EP	Exponential mapping (10.2.2)	parallel translation (10.2.11)
QfDT	qf retraction (10.2.3)	vector transport by direction rotation based on tangent space(9.2.18)
QfDN	qf retraction (10.2.3)	vector transport by direction rotation based on normal space(9.2.17)
QfP	qf retraction (10.2.3)	vector transport by parallelization (9.2.19)
QfR	qf retraction (10.2.3)	vector transport by rigging (9.2.20)
QfInP	qf retraction (10.2.3)	vector transport by parallelization (9.2.19) using intrinsic representation
CPIn	retraction (10.2.16)	vector transport by parallelization (9.2.19) using intrinsic representation

11.4.6 Retractions and Vector Transports for RTR-SR1

The performance of RTR-SR1 with different retractions and isometric vector transports is shown in this section. Table 11.3 lists the pairs that are tested and the notation used. Since RTR-SR1 does not require an isometric vector transport to satisfy the locking condition, Methods 1 and 2 in Section 4.4 are not included since they require more computation per iteration and in our experience their convergence rates do not result in overall computational savings.

Table 11.9 and Figure 11.5 show the results of the experiments, when (p, n) is $(4, 12)$ and $(8, 12)$. The conclusions about the cost of vector transports are similar to those in Section 11.4.2. First, RTR-SR1 with EP requires much more time than others since parallel translation is expensive computationally. Second, RTR-SR1 with QfDT and RTR-SR1 with QfP are faster than RTR-SR1 with QfDN and RTR-SR1 with QfR when $(p, n) = (8, 12)$ and the relationship is reversed when $(p, n) = (4, 12)$ due to the ratio of the dimension and codimension of the manifold and its influence on the computational cost of vector transport. Third, RTR-SR1 with CPIn is the fastest algorithm. This pair of retraction and vector transport is preferred and is used in other comparisons in this chapter.

Unlike the results in Section 11.4.2, we do not always observe the vector transports that are equivalent theoretically give identical numbers of iterations, function evaluations, gradient eval-

Table 11.9: Comparison of retraction and vector transports for RTR-SR1. The subscript $-k$ indicates a scale of 10^{-k}.

p, n		RTR-SR1						
		EP	QfDT	QfDN	QfP	QfR	QfInP	CPIn
4, 12	$iter$	82	81	81	92	93	92	81
	nf	82	81	81	92	93	92	81
	ng	82	81	81	92	93	92	81
	nH	292	291	291	330	360	329	289
	nV	81	80	80	91	92	91	80
	nR	81	80	80	91	92	91	80
	gf_f	3.47_{-5}	3.00_{-5}	3.00_{-5}	1.94_{-5}	3.18_{-5}	1.93_{-5}	1.63_{-5}
	gf_f/gf_0	9.73_{-7}	8.41_{-7}	8.41_{-7}	5.42_{-7}	8.90_{-7}	5.42_{-7}	4.58_{-7}
	t	6.01_{1}	3.07_{-1}	2.18_{-1}	2.30_{-1}	2.25_{-1}	6.93_{-2}	5.67_{-2}
8, 12	$iter$	114	113	124	172	137	168	123
	nf	114	113	124	172	137	168	123
	ng	114	113	124	172	137	168	123
	nH	465	476	480	624	540	666	445
	nV	113	112	123	171	136	167	122
	nR	113	112	123	171	136	167	122
	gf_f	6.06_{-5}	3.63_{-5}	3.86_{-5}	8.21_{-5}	4.24_{-5}	3.72_{-5}	7.22_{-5}
	gf_f/gf_0	6.89_{-7}	4.13_{-7}	4.39_{-7}	9.33_{-7}	4.83_{-7}	4.23_{-7}	8.22_{-7}
	t	1.98_{2}	1.20	1.54	9.24_{-1}	1.27	1.71_{-1}	9.50_{-2}

uations, Hessian actions, vector transports and retractions. For example, the differences between QfDT and QfDN, QfP and QfInP for $(p, n) = (8, 12)$ are non-negligible. However, even though theoretically, the pairs QfDT and QfDN, QfP and QfInP are equivalent respectively, when $(p, n) = (8, 12)$ we observed the iterations are identical only when $iter$ is less than 60 for both pairs. So RTR-SR1 is more sensitive to the numerical differences in the implementations of the pairs than RBFGS. This is not too surprising since the RTR-SR1 and the restricted RBroyden family have significantly different properties of approximation of the (inverse) Hessian, different local models to set the direction vector and different convergence properties of the inner iteration on the local models.

11.4.7 Comparison of LRTR-SR1 and RTR-SR1

The performance results for RTR-SR1 and LRTR-SR1 with different values of the parameter m are given in Table 11.10 and Figure 11.6 for the Brockett cost function with $n = p = 16$ (a

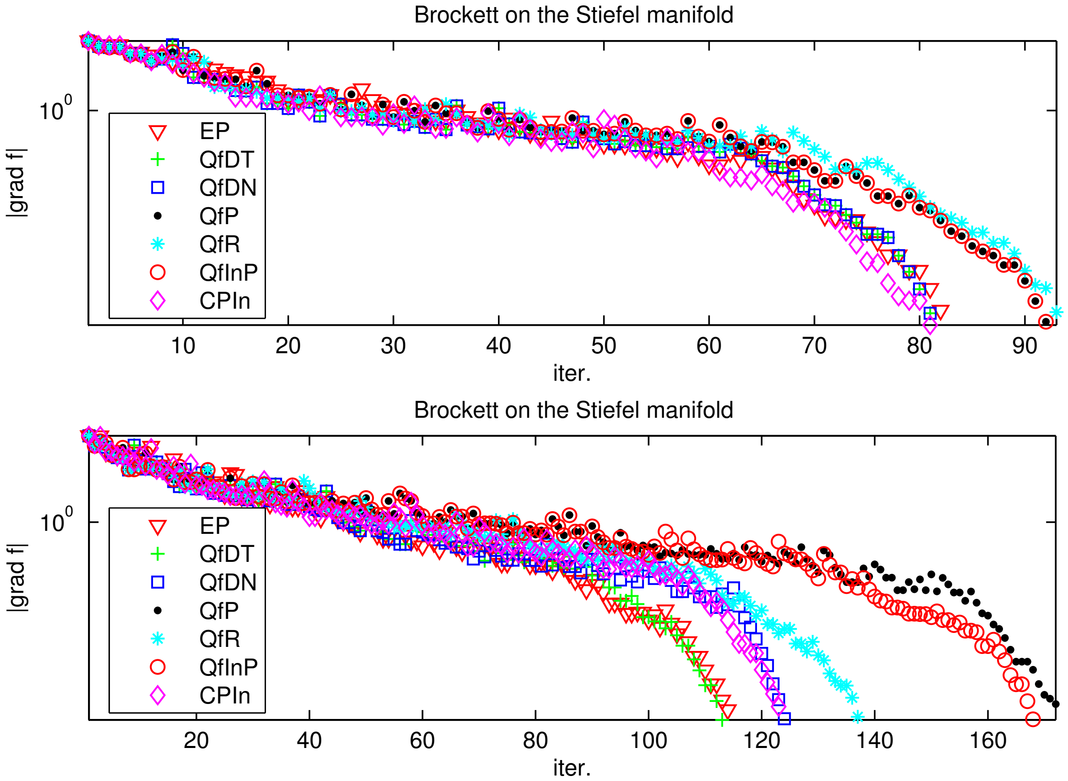


Figure 11.5: Comparison of retraction and vector transports for RTR-SR1. The top figure corresponds to $(p, n) = (4, 12)$ and the bottom figure corresponds to $(p, n) = (8, 12)$.

Table 11.10: Comparison of RTR-SR1 and LRTR-SR1. The subscript $-k$ indicates a scale of 10^{-k}.

method	RTR-SR1	LRTR-SR1					
m		1	2	4	8	16	32
$iter$	180	707	994	399	445	575	542
nf	180	707	994	399	445	575	542
ng	180	707	994	399	445	575	542
nV	179	1938	4315	3042	6186	14964	27621
nR	179	706	993	398	444	574	541
gf_f	8.19_{-5}	1.46_{-4}	1.39_{-4}	1.13_{-4}	1.45_{-4}	1.44_{-4}	1.18_{-4}
gf_f/gf_0	5.55_{-7}	9.91_{-7}	9.44_{-7}	7.66_{-7}	9.82_{-7}	9.73_{-7}	8.02_{-7}
t	1.74_{-1}	6.01_{-1}	9.73_{-1}	4.93_{-1}	7.75_{-1}	1.58	2.68

problem of moderate size from a dense matrix computational point of view). Unlike the results for LRBFGS shown in Table 11.6, the number of iterations required by LRTR-SR1 does not decreases as m increases. Like the results shown in Table 11.6, the number of iterations remains higher than the number required by RTR-SR1. Unfortunately, the computation time of LRTR-SR1 is higher than RTR-SR1 for the range of m considered. Therefore, LRTR-SR1 is not competitive with RTR-SR1 for this moderately sized problem. However, for the large scale versions of the Brockett cost function in Section 11.4.9, LRTR-SR1 does show an advantage in computational efficiency.

11.4.8 Convergence Rate Comparison

Table 11.11 and Figure 11.7 report the observed values of the basic experimental metrics for RSD, RTR-SD, RBFGS, RTR-SR1 and RTR-Newton applied to the Brockett cost function with $(p, n) = (6, 12)$. From this data we can verify our theoretical convergence rate analyses,

Since RSD and RTR-SD are convergent linearly and are the slowest among the tested algorithms, the observed number iterations to achieve the required accuracy are significantly larger than those of the other algorithms. RBFGS requires fewer iterations than RTR-SR1 which is also consistent with their convergence rates, i.e., RBFGS converges superlinearly and RTR-SR1 converges $d+1$-step superlinearly. RTR-Newton requires the fewest iterations due to its quadratic convergence rate. Since the action of the Hessian of the Brockett cost function on a vector is computationally cheap, RTR-Newton requires the smallest computational time among all the tested algorithms. Chapter 12 contains results on applications where the action of the Hessian on a vector is not

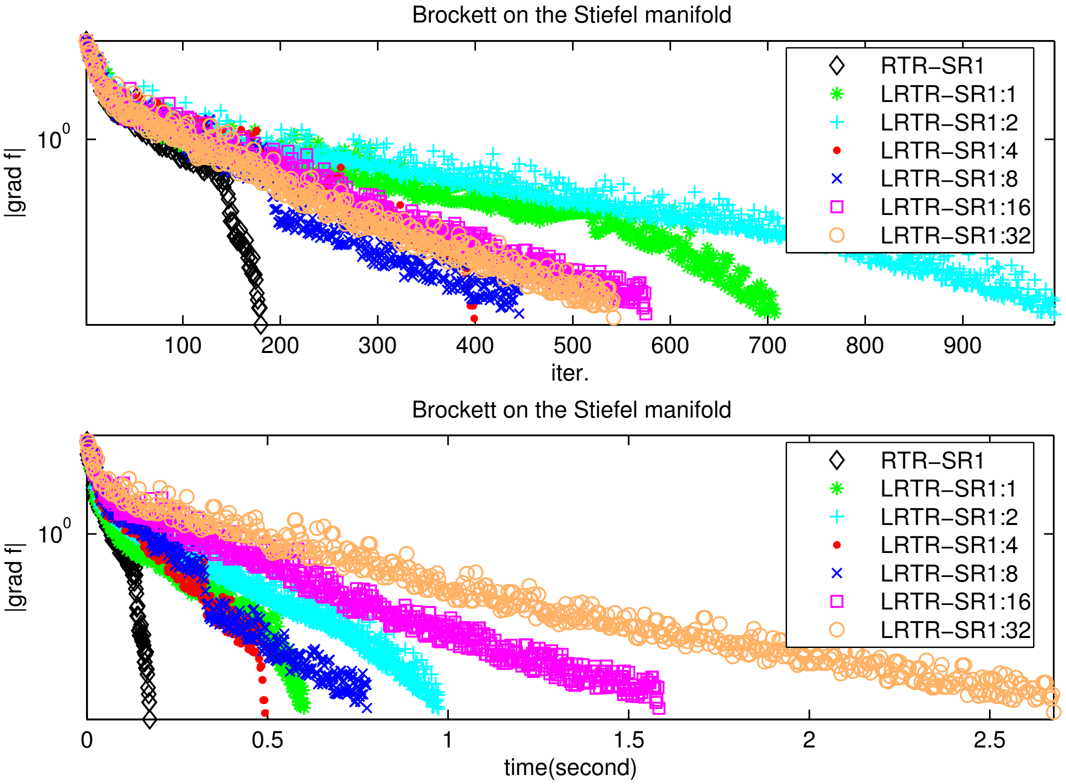


Figure 11.6: Comparison of RTR-SR1 and LRTR-SR1. The top figure is the results of *iter* versus $|gradf|$ and the bottom one is the results of *time* versus $|gradf|$.

Table 11.11: Comparison of RTR-Newton, RBFGS, RTR-SR1, RSD and RTR-SD. The subscript $-k$ indicates a scale of 10^{-k}.

method	RTR-Newton	RBFGS	RTR-SR1	RSD	RTR-SD
$iter$	16	61	137	2888	3127
nf	16	154	137	8691	3127
ng	16	61	137	2888	3114
nH	185	120	509	0	0
nV	0	180	136	2887	0
nR	15	153	136	8690	3126
gf_f	1.43_{-6}	5.19_{-5}	5.25_{-5}	6.79_{-5}	5.20_{-5}
gf_f/gf_0	2.10_{-8}	7.60_{-7}	7.69_{-7}	9.95_{-7}	7.62_{-7}
t	2.98_{-2}	6.57_{-2}	1.00_{-1}	2.48	1.61

computationally cheap and the performance predictions based on the convergence rates and the relative computational efficiency per iteration of methods other than RTR-Newton being fastest are verified with experimental observations.

11.4.9 A Large Scale Problem

Earlier experiments have evaluated the performance of limited memory algorithms LRBFGS and LRTR-SR1 on moderately sized problems. In this section, the potential of the methods is demonstrated by applying them to the Brockett cost function for several sufficiently large values of n. The qf retraction is used and isometric vector transport is defined by rigging modified by two rank-1 updates. The parameter m in LRBFGS is set to 4. Besides comparing the performance of LRBFGS and LRTR-SR1, the performance of another method suitable for large-scale problems, the Riemannian conjugate gradient algorithm (RCG) defined in [AMS08], is included. RCG uses a modified Polak-Ribiére formula (see [NW06, (5.45)]) and imposes the same Wolfe conditions as LRBFGS on the line search for the step size.

The performance results for LRBFGS, LRTR-SR1 and RCG for several values of the pair $(n,\ p)$ are shown in Table 11.12 and Figure 11.8. The reductions of the norm of the initial gradient are comparable so all algorithms provide similar optimization performance. However, the computation time required by LRBFGS to achieve the reduction is smaller than the computation time required by LRTR-SR1, which is smaller than the computation time required by RCG. The number of iterations for LRBFGS is smaller or comparable to RCG and LRTR-SR1 needs the largest number

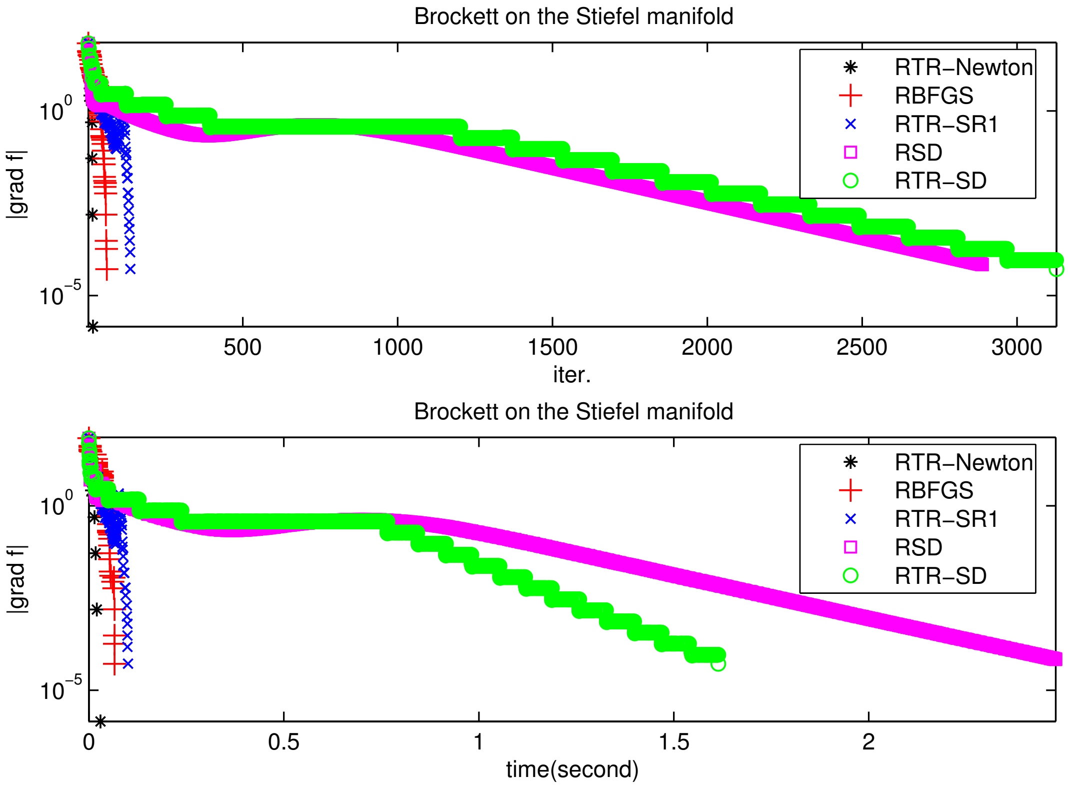


Figure 11.7: Comparison of RTR-Newton, RBFGS, RTR-SR1, RSD and RTR-SD. The top figure is the results of *iter* versus $|gradf|$ and the bottom one is the results of *time* versus $|gradf|$.

Table 11.12: LRBFGS, LRTR-SR1 and RCG for large scale problems. The subscript $-k$ indicates a scale of 10^{-k}.

$(n,\ p)$	(1000, 2)			(1000, 3)		
method	LRBFGS	LRTR-SR1	RCG	LRBFGS	LRTR-SR1	RCG
$iter$	175	399	236	293	778	438
nf	194	399	1079	315	778	1906
ng	175	399	1079	293	778	1906
nV	1554	3018	1078	2616	5909	1905
nR	193	398	843	314	777	1468
gf_f	1.86_{-4}	1.66_{-4}	1.96_{-4}	3.17_{-4}	3.25_{-4}	3.17_{-4}
gf_f/gf_0	9.48_{-7}	8.44_{-7}	9.98_{-7}	9.59_{-7}	9.85_{-7}	9.58_{-7}
t	5.47	1.23_{1}	4.33_{1}	1.62_{1}	4.19_{1}	1.28_{2}

iterations than the other two. The main source of the difference in computation time is seen in the much larger numbers of function and gradient evaluations required by RCG. This is due to the line search having difficulty satisfying the Wolfe conditions and we conclude that LRBFGS and LRTR-SR1 are viable approaches for large scale problems typified by the Brockett cost function. The results also indicate that understanding the properties of application problems that indicated when to prefer either a line search approach for the RBroyden family using one of the restricted members or the trust region with the Riemannian SR1 updated member of the RBroyden family.

11.4.10 Comparison of the Stiefel Manifold and the Grassmann Manifold

If the $\mathcal{M}$ in a minimization problem $\min f(x), x \in \mathcal{M}$ is a quotient manifold, then the minimization problem can be considered as being defined on the total manifold $\bar{\mathcal{M}}$ of $\mathcal{M}$. One may also ask whether there exist differences between using $\bar{\mathcal{M}}$ and $\mathcal{M}$. Noting that the Grassmann manifold is the quotient manifold of the quotient Stiefel manifold and the Rayleigh quotient minimization problem is defined on the Grassmann manifold we consider that problem in this section.

The parameter (p, n) is set to $(6, 12)$. Two sets of vector transports and retractions are tested. The first set is the vector transports by parallelization and the retractions using the idea in Section 4.4.3. The second set is the parallel translations and exponential mappings. Results using the embedded Stiefel manifold are included to complete the comparisons.

The experimental results are reported in Tables 11.13 and 11.14 and Figure 11.9. As seen from Table 11.13 and the top graph of Figure 11.9, algorithms with pairs of metric and transport in

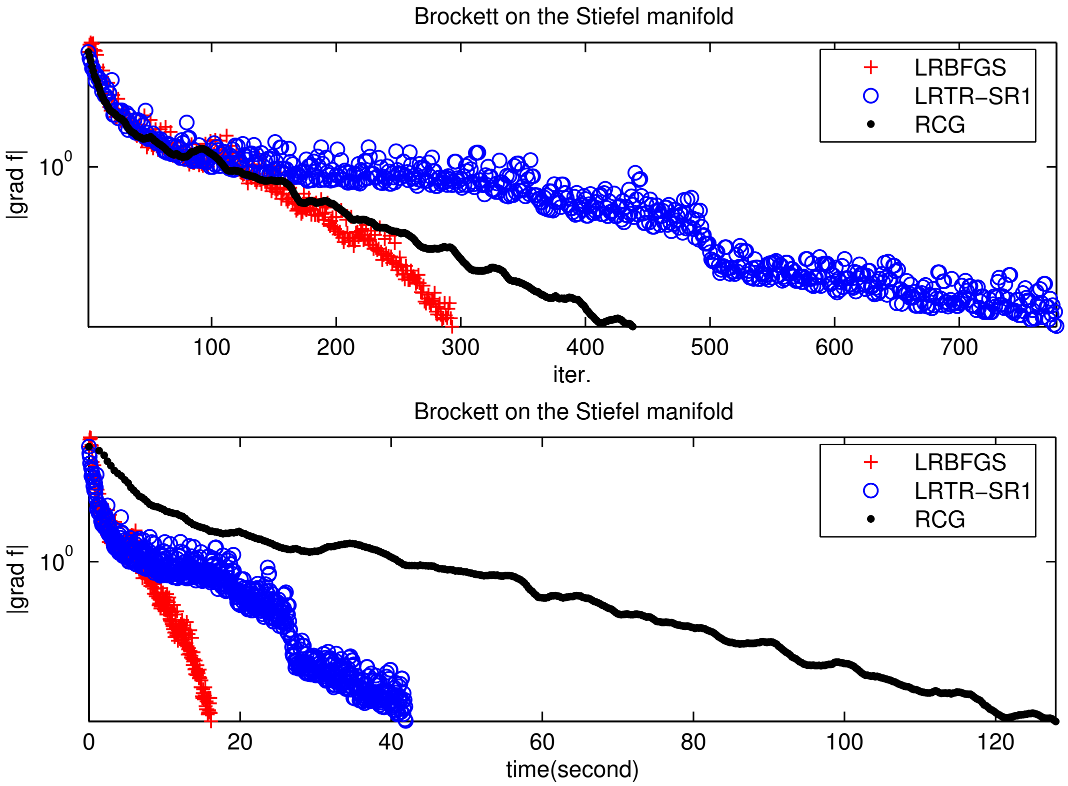


Figure 11.8: Comparison of LRBFGS, LRTR-SR1 and RCG for the Brockett cost function with $(p, n) = (3, 1000)$. The top figure is the results of *iter* versus $|gradf|$ and the bottom one is the results of *time* versus $|gradf|$.

the first set have identical numbers of iterations, function evaluations, gradient evaluations, vector transports and retractions. This behavior confirms theoretical expectations. The retractions and vector transports in the first set of pairs are the same independently of the metric. Additionally, for this cost function and metrics, the gradients are the same. Therefore, the iterations are expected to be the same. Note that for other cost functions and these same metrics, the gradients might differ and one would not expect this behavior.

The second set of pairs of vector transports and retractions comprising exponential mappings and parallel translations are dependent on the metric of the manifold and we do not expect the replication of behavior seen for the first set. The experimental results are in Table 11.14 and the bottom graph of Figure 11.9. Note that across the table the number of iterations is essentially the same as the results from the first set of vector transports and retractions. Since we know that the exponential mappings and parallel translations are at least as expensive computationally as alternative vector transports and retractions and are very often considerably more expensive computationally, we expect the times in the Table to not improve relative to the first set. This is seen to be the case and once again the basic premise of producing more efficient optimization algorithms based on carefully constructed and computationally efficient vector transports and retractions is confirmed. For this manifold, the first set shows that this is possible in a manner that is independent of these typical metrics.

11.4.11 Comparison of RGS and RSD for Smooth Functions

RGS is an algorithm to find an optimum for a partly smooth function, where smooth means differentiable, not infinitely differentiable. It can be used also to find an optimum of a smooth function. In this section, we compare RGS with a classic linearly convergent algorithm, RSD. We choose small $(p, n) = (3, 6)$ since the quadratic programming in each step RGS is expensive computationally. Multiple sampling radii, ϵ_0, are tested. Since the cost function is smooth, the stopping criterion for a partly smooth function is disabled, i.e., tolerance $\tau_\nu = 0$ and the default stopping criterion for smooth cost functions is used.

The numerical results are shown in Tables 11.15 and Figure 11.10. RGS uses gradients of neighbors of the current iterate to approximate the gradient. When the radius of the neighborhood is very small, the gradient approximation is very close to the true gradient and RGS is essentially RSD. RGS almost always take more computation time to achieve the same accuracy as RSD.

Table 11.13: Comparison of the embedded Stiefel manifold(ES), the quotient Stiefel manifold(QS) and the Grassmann manifold(GR) for the Rayleigh quotient problem using RBFGS and RTR-SR1. The retractions used are (10.2.16), (10.3.4), (10.6.19) for ES, QS and GR respectively. The vector transports are by parallelization. The subscript $-k$ indicates a scale of 10^{-k}.

method	RBFGS			RTR-SR1		
manifold	ES	QS	GR	ES	QS	GR
$iter$	41	41	41	32	32	32
nf	85	85	85	32	32	32
ng	41	41	41	32	32	32
nH	80	80	80	86	86	86
nV	120	120	120	31	31	31
nR	84	84	84	31	31	31
gf_f	1.01_{-5}	1.01_{-5}	1.01_{-5}	1.40_{-5}	1.40_{-5}	1.40_{-5}
gf_f/gf_0	6.68_{-7}	6.68_{-7}	6.68_{-7}	9.22_{-7}	9.22_{-7}	9.22_{-7}
t	3.50_{-2}	3.57_{-2}	3.42_{-2}	2.06_{-2}	2.18_{-2}	1.78_{-2}

Table 11.14: Comparison of the embedded Stiefel manifold(ES), the quotient Stiefel manifold(QS) and the Grassmann manifold(GR) for the Rayleigh quotient problem using RBFGS and RTR-SR1. The retractions and vector transports are exponential mapping and parallel translation. The subscript $-k$ indicates a scale of 10^{-k}.

method	RBFGS			RTR-SR1		
manifold	ES	QS	GR	ES	QS	GR
$iter$	42	54	41	29	31	32
nf	88	134	85	29	31	32
ng	42	54	41	29	31	32
nH	82	106	80	75	91	86
nV	123	162	120	28	30	31
nR	87	133	84	28	30	31
gf_f	1.08_{-5}	1.26_{-5}	1.01_{-5}	1.42_{-5}	1.08_{-5}	1.40_{-5}
gf_f/gf_0	7.13_{-7}	8.31_{-7}	6.68_{-7}	9.38_{-7}	7.14_{-7}	9.22_{-7}
t	5.81_{1}	3.82_{1}	1.01_{-1}	3.15_{1}	1.69_{1}	9.44_{-2}

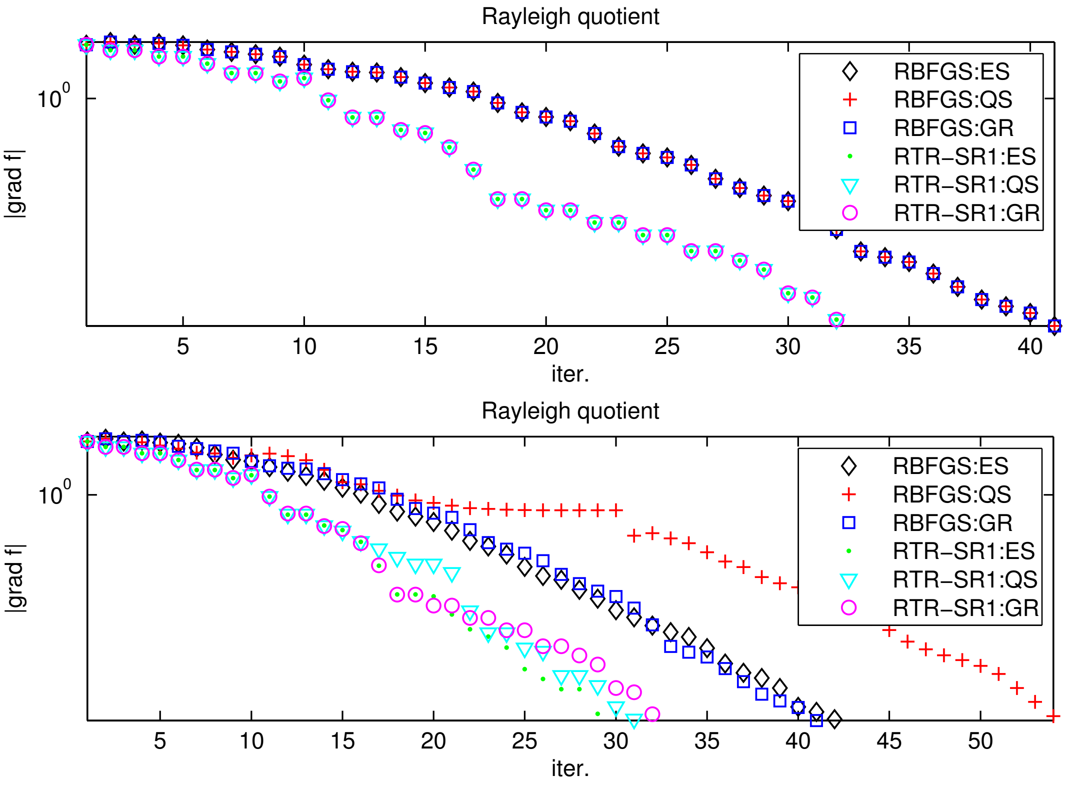


Figure 11.9: Comparison of the embedded Stiefel manifold(ES), the quotient Stiefel manifold(QS) and the Grassmann manifold(GR) for the Rayleigh quotient problem using RBFGS and RTR-SR1. The retractions used in the top figure are (10.2.16), (10.3.4), (10.6.19) for ES, QS and GR respectively. The vector transports are by parallelization. The retractions and vector transports used in the bottom figure are exponential mapping and parallel translation.

Table 11.15: Comparison of RSD and RGS with multiple initial sampling radii. The subscript $-k$ indicates a scale of 10^{-k}.

method	RSD	RGS			
ϵ_0		10	10^{-1}	10^{-3}	10^{-5}
$iter$	149	97	82	70	80
nf	439	1378	1163	692	1142
ng	149	2113	1783	1519	1739
nV	148	2016	1701	1449	1659
nR	438	3489	2944	2209	2879
gf_f	1.86_{-5}	1.17_{-5}	1.83_{-5}	1.01_{-5}	1.13_{-5}
gf_f/gf_0	9.91_{-7}	6.25_{-7}	9.77_{-7}	5.40_{-7}	6.03_{-7}
t	1.06_{-1}	2.34	1.82	1.38	1.63

This is due to the extra work relative to RSD required when computing sample gradients and transporting them to a reference tangent space. In addition, a quadratic program must be solved. These processes are expensive and in this form RGS is seen not to be suitable for large dimensional problems.

11.4.12 Comparison of RGS and RBFGS for Partly Smooth Functions Defined on a Riemannian Manifold

The Lipschitz and non-Lipschitz minmax problems on the sphere are used in this section to test the performance of RGS and RBFGS. The retraction and vector transport are taken as (10.4.2) and (10.4.4).

Table 11.16 shows the performance of RGS and RBFGS for a partly smooth Lipschitz continuous function. Both of the algorithms work well. In addition, this performance illustrates the prediction in Chapter 7 that RBFGS should be faster than RGS.

Chapter 7 also predicted that RBFGS should have difficulties optimizing a partly smooth non-Lipschitz continuous function while RGS, by design, should converge reasonably well. The main concern with RGS is the computational cost required to achieve such reliable convergence. The experimental results in Table 11.17 illustrate these expectations.

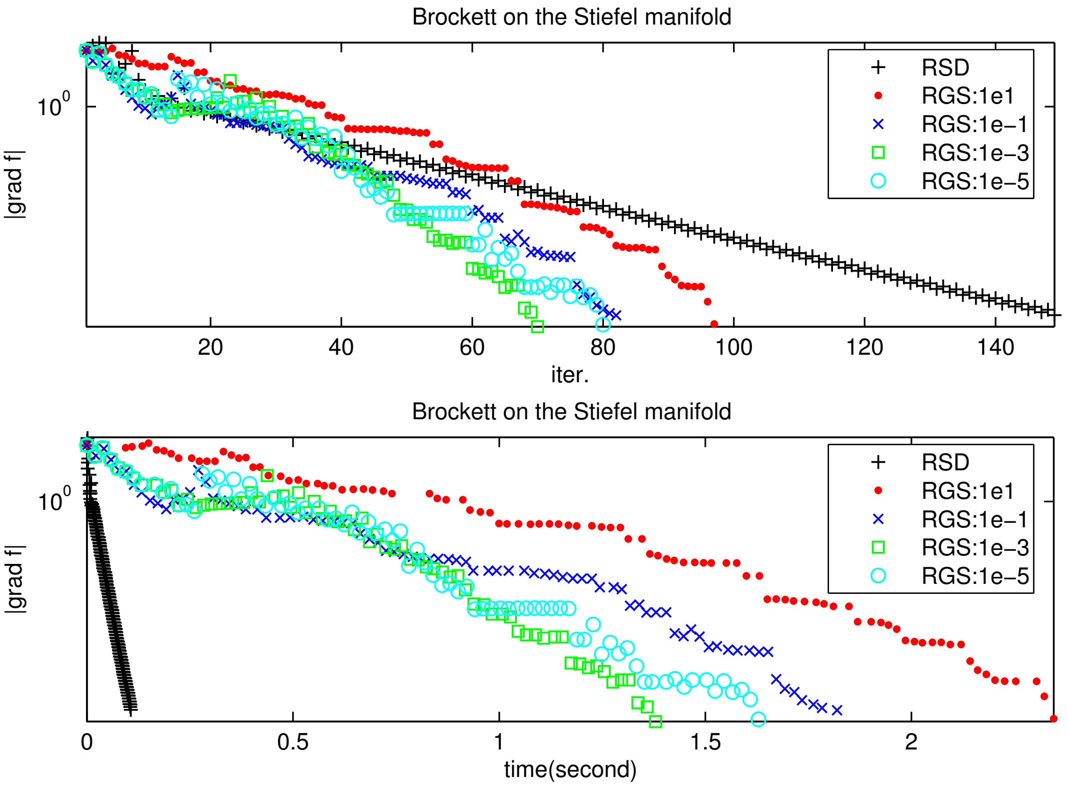


Figure 11.10: Comparison of RSD and RGS with multiple initial sampling radii.

Table 11.16: Comparison of RGS and RBFGS for a partly smooth Lipschitz continuous function.

	n	RGS	RBFGS	n	RGS	RBFGS	n	RGS	RBFGS
$iter$		38	26		107	53		1486	106
nf		534	77		3134	167		63325	378
ng		380	77		1712	167		38636	378
nH		0	50		0	104		0	210
nV	4	342	133	8	1605	347	16	37150	1017
nR		912	76		4844	166		101959	377
gf_f		8.66_{-1}	8.66_{-1}		9.35_{-1}	9.35_{-1}		9.68_{-1}	9.68_{-1}
gf_f/gf_0		1.07	1.07		1.05	1.05		1.04	1.04
t		2.70_{-1}	3.66_{-2}		1.19	8.85_{-2}		3.08_1	3.23_{-1}

Table 11.17: Comparison of RGS and RBFGS for a partly smooth non-Lipschitz continuous function. ”lsf” means line search fails.

	n	RGS	RBFGS	n	RGS	RBFGS	n	RGS	RBFGS
$iter$	4	35	27	8	44	lsf	16	128	lsf
nf		395	104		698	lsf		3678	lsf
ng		350	104		704	lsf		3328	lsf
nH		0	52		0	lsf		0	lsf
nV		315	170		660	lsf		3200	lsf
nR		743	103		1400	lsf		7004	lsf
gf_f		1.05_4	1.08_4		1.30_4	lsf		1.94_4	lsf
gf_f/gf_0		8.37_3	8.60_3		1.10_4	lsf		2.38_4	lsf
t		2.57_{-1}	4.34_{-2}		5.25_{-1}	lsf		3.26	lsf

CHAPTER 12

SOFT DIMENSION REDUCTION FOR INDEPENDENT COMPONENT ANALYSIS AND SYNCHRONIZATION OF ROTATION PROBLEM

12.1 Soft Dimension Reduction for Independent Component Analysis

12.1.1 Introduction

Independent Component Analysis (ICA) is a key task in many statistical data analysis applications. The task is to determine an independent component form of a random vector, typically known through a large number of samples, or to determine a few independent components similar to Principal Component Analysis [HKO01]. The solution to the ICA problem does not lend itself to a simple characterization and therefore a large number of heuristic approaches based on approximate characterizations have been proposed. Joint diagonalization of a set of sample covariance matrices is one popular and effective approximate characterization with which we have some experience [AG06]. Theis et al. [TCA09] explored the problem of extracting a few sources from information that has several data sources mixed using joint diagonalization.

12.1.2 Problem Statement

The cost function of the joint diagonalization problem on the Stiefel manifold is

$$f : \mathrm{St}(p, n) \to \mathbb{R} : Y \longmapsto f(Y) = -\sum_{i=1}^{N} \| \operatorname{diag}(Y^T C_i Y)\|_F^2, \qquad (12.1.1)$$

where C_i are known symmetric matrices and $\operatorname{diag}(M)$ is a vector formed by diagonal entries of a matrix M. The Stiefel manifold can be viewed as a quotient manifold or embedded manifold with each view defining a metric. Since both gradient and Hessian are related to metric, they also have two distinct forms.

The gradient of this function with respect to the metric (10.2.1) is

$$\operatorname{grad} f(Y) = P_Y \operatorname{grad} \hat{f}(Y),$$

and the gradient with respect to the metric (10.3.1) is

$$\operatorname{grad} f(Y) = \operatorname{grad} \hat{f}(Y) - Y(\operatorname{grad} \hat{f}(Y))^T Y.$$

where $P_Y\xi = \xi - Y \operatorname{sym}(Y^T\xi)$, $\operatorname{grad} \hat{f}(Y) = -\sum_{i=1}^N 4C_iY \operatorname{ddiag}(Y^TC_iY)$ and $\operatorname{ddiag}(M)$ is a diagonal matrix whose diagonal entries are the diagonal entries of a matrix M.

The Hessian of this function with respect to the metric (10.2.1) is

$$\operatorname{Hess} f(Y)[\xi] = P_Y[\mathrm{D} \operatorname{grad} \hat{f}(Y)[\xi] - \xi \operatorname{sym}(Y^T \operatorname{grad} \hat{f}(Y))], \tag{12.1.2}$$

and the Hessian with respect to the metric (10.3.1) is

$$\begin{aligned}\operatorname{Hess} f(Y)[\xi] = {} & \mathrm{D} \operatorname{grad} \hat{f}(Y)[\xi] - Y(\mathrm{D} \operatorname{grad} \hat{f}(Y)[\xi])^T Y - Y \operatorname{skew}((\operatorname{grad} \hat{f}(Y))^T \xi) \\ & - \operatorname{skew}(\xi(\operatorname{grad} \hat{f}(Y))^T)Y - \frac{1}{2}(I - YY^T)\xi Y^T \operatorname{grad} \hat{f}(Y)\end{aligned} \tag{12.1.3}$$

where $\mathrm{D} \operatorname{grad} \hat{f}(Y)[\xi]$ is

$$\mathrm{D} \operatorname{grad} \hat{f}(Y)[\xi] = -\sum_{i=1}^N 4C_i(\xi \operatorname{ddiag}(Y^TC_iY) + Y \operatorname{ddiag}(\xi^TC_iY) + Y \operatorname{ddiag}(Y^TC_i\xi)).$$

In [TCA09], Theis et al. used the RTR-Newton method in [Bak08]. RTR-Newton requires the action of the Hessian on a tangent vector and converges quadratically. For this cost function, however, the actions of the Hessians on tangent vectors in (12.1.2) and (12.1.3) are expensive when N is large. In this chapter, we assess the ability of our Riemannian quasi-Newton algorithms to provide sufficiently fast convergence while avoiding evaluation of the action of the Hessian on a tangent vector.

12.1.3 Implementations and Results

The cost function (12.1.1) is defined on the Stiefel manifold and details about the implementation of objects on that manifold are discussed in Section 9. The retraction is chosen to be (10.2.16) and the vector transport is chosen to be by parallelization (9.2.19) with intrinsic representation since they are the preferred pair for the Stiefel manifold as shown in Section 11.4.2 and 11.4.6. From Section 11.4.10, we can see that there is no differences between the metric (10.2.1) and the metric (10.3.1) if the preferred retraction and vector transport are used. Therefore, without loss of generality, we only show the experimental results when using the metric (10.2.1). The algorithmic parameters settings are the same as those in Section 11.3.

The problem size parameters are $(p, n) = (4, 12)$. Table 12.1 and Figure 12.1 present the experimental results obtained for the joint diagonalization problem (12.1.1). The C_i matrices in (12.1.1) are chosen to be approximately jointly diagonalizable, which is normally the case in practical applications. Specifically, the C_i matrices are selected as $C_i = \mathrm{diag}(n, n-1, \dots, 1) + \epsilon_{\mathrm{C}}(R_i + R_i^T)$, where the elements of $R_i \in \mathbb{R}^{n \times n}$ are independently drawn from the standard normal distribution. Table 12.1 and Figure 12.1 correspond to $\epsilon_{\mathrm{C}} = 0.1$, but we have observed similar results for a wide range of values of ϵ_{C}. Table 12.1 indicates that RTR-Newton requires fewer iterations than RTR-SR1, which requires fewer iterations than RBFGS and LRTR-SR1. This was expected since RTR-Newton uses the Hessian of f while RBFGS and RTR-SR1 use an inexact Hessian and LRBFGS and LRTR-SR1 are further constrained by the limited memory. However, the iterations of RTR-Newton tend to be more time-consuming that those of the quasi-Newton methods, all the more so if N gets large since the number of terms in the Hessian of f is linear in N. The experiments reported in Table 12.1 show that the trade-off between the number of iterations and the time per iteration is in favor of LRBFGS and RTR-SR1 for N sufficiently large.

12.2 Synchronization of Rotation Problem

12.2.1 Introduction

The synchronization problem is to find N unknown rotations $R_1, \dots, R_N \in SO(n)$ from M noisy measurements, H_{ij} of $\tilde{H}_{ij} = R_i R_j^T$. In general, noisy measurements are not given for all $\tilde{H}_{ij}$. We can induce a graph $G = (V, E)$, with vertices $V = \{1, 2, \dots, N\}$ and edges

$$E = \{(i, j) | j > i \text{ and } H_{ij} \text{ or } H_{ji} \text{ is given.}\}$$

Boumal et al. [BSAB12] give a overview of the applications and previous work of this problem and propose a Riemannian approach. In this chapter, the Riemannian quasi-Newton algorithms are applied for their approach.

12.2.2 Problem Statement

The noisy measurements H_{ij} satisfy $H_{ij} = Z_{ij} \tilde{H}_{ij}$ where $Z_{ij} \in SO(n)$ is noise. The noise Z_{ij} is sampled from the isotropic Langevin distribution on $SO(n)$ with mean I_n and outliers [BSAB12,

Table 12.1: Comparison of RTR-Newton and Riemannian quasi-Newton algorithms for the joint diagonalization problem: $n = 12, p = 4, \epsilon_{\mathrm{C}} = 0.1$

N		RTR	RBFGS	LRBFGS			RTR	LRTR-SR1		
		Newton		$m:2$	$m:4$	$m:8$	SR1	$m:2$	$m:4$	$m:8$
16	$iter$	12	73	97	97	87	163	300	212	206
	nf	12	328	114	117	99	163	300	212	206
	ng	12	73	97	97	87	163	300	212	206
	nH	96	144	0	0	0	439	0	0	0
	nV	0	216	478	852	1406	162	1429	1811	3187
	nR	11	327	113	116	98	162	299	211	205
	gf_f	5.99_{-4}	2.16_{-3}	1.97_{-3}	2.48_{-3}	1.75_{-3}	1.96_{-3}	2.47_{-3}	1.50_{-3}	2.49_{-3}
	gf_f/gf_0	2.40_{-7}	8.61_{-7}	7.84_{-7}	9.88_{-7}	6.97_{-7}	7.83_{-7}	9.87_{-7}	6.00_{-7}	9.95_{-7}
	t	6.40_{-2}	1.34_{-1}	8.59_{-2}	9.37_{-2}	9.68_{-2}	1.28_{-1}	3.25_{-1}	2.88_{-1}	3.55_{-1}
64	$iter$	16	87	127	108	99	237	587	371	195
	nf	16	391	162	137	126	237	587	371	195
	ng	16	88	127	108	99	237	587	371	195
	nH	120	172	0	0	0	683	0	0	0
	nV	0	259	628	951	1610	236	2860	3218	2998
	nR	15	390	161	136	125	236	586	370	194
	gf_f	2.58_{-3}	3.11_{-3}	8.48_{-3}	5.72_{-3}	7.19_{-3}	5.17_{-3}	8.26_{-3}	7.75_{-3}	6.90_{-3}
	gf_f/gf_0	2.90_{-7}	3.50_{-7}	9.56_{-7}	6.45_{-7}	8.10_{-7}	5.83_{-7}	9.31_{-7}	8.73_{-7}	7.78_{-7}
	t	2.14_{-1}	3.20_{-1}	2.11_{-1}	1.89_{-1}	2.02_{-1}	3.61_{-1}	1.05	7.45_{-1}	4.80_{-1}
256	$iter$	13	72	94	92	83	105	267	217	236
	nf	13	397	119	117	104	105	267	217	236
	ng	13	72	94	92	83	105	267	217	236
	nH	109	142	0	0	0	289	0	0	0
	nV	0	213	463	807	1338	104	1292	1868	3767
	nR	12	396	118	116	103	104	266	216	235
	gf_f	8.08_{-3}	2.00_{-2}	3.21_{-2}	3.30_{-2}	2.14_{-2}	3.49_{-2}	1.88_{-2}	3.43_{-2}	3.63_{-2}
	gf_f/gf_0	2.00_{-7}	4.94_{-7}	7.91_{-7}	8.15_{-7}	5.29_{-7}	8.62_{-7}	4.63_{-7}	8.47_{-7}	8.97_{-7}
	t	6.61_{-1}	8.81_{-1}	4.70_{-1}	4.50_{-1}	4.25_{-1}	4.42_{-1}	1.27	1.05	1.25

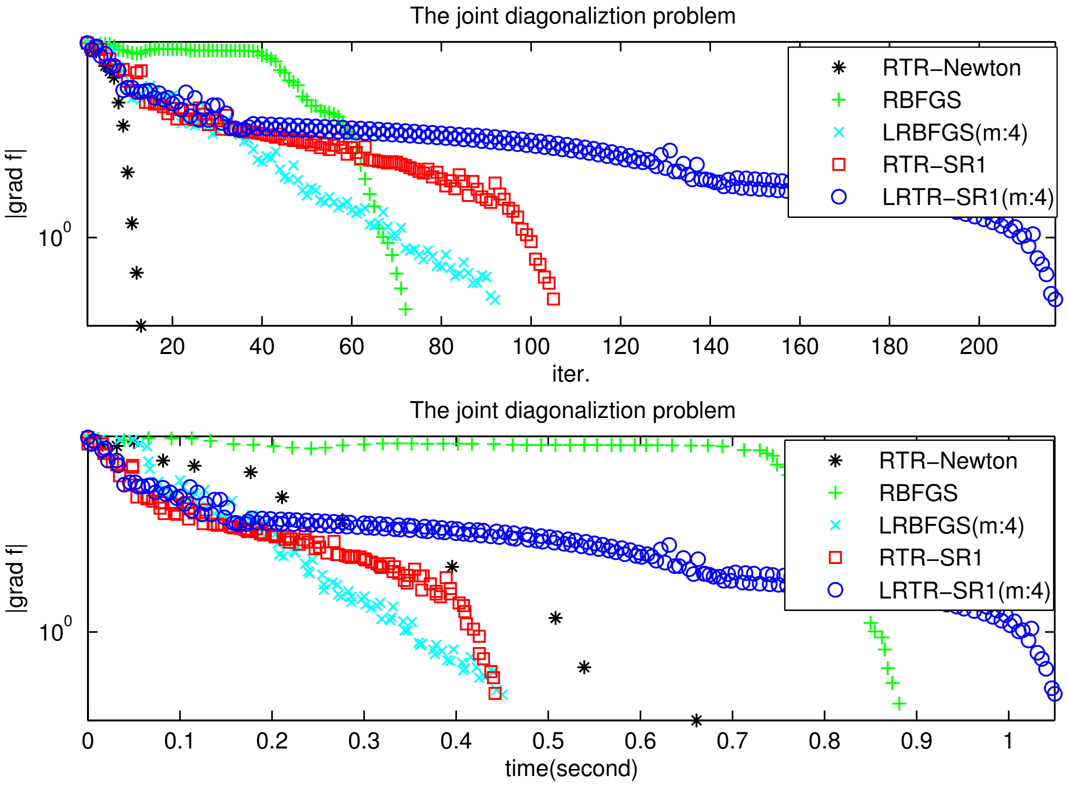


Figure 12.1: Comparison of RTR-Newton and Riemannian quasi-Newton algorithms for the joint diagonalization problem: $n = 12, p = 4, N = 256, \epsilon_{\text{C}} = 0.1$

(4.12)]. The log-likelihood function [BSAB12, (2.5)] is then

$$\begin{aligned} L : SO(n) \times \cdots \times SO(n) \to \mathbb{R} : R = (R_1, \ldots, R_N) \longmapsto L(SO(n) \times \cdots \times SO(n)) \\ = \frac{1}{2} \sum_{(i,j)\in E} \log(\frac{p}{c_n(\kappa)} \exp(\kappa \operatorname{trace}(R_i^T H_{ij} R_j)) + 1 - p), \end{aligned} \tag{12.2.1}$$

where $p \in [0, 1]$, $\kappa > 0$, H_{ij} are the observed matrices in $SO(n)$ and

$$c_n(\kappa) = \int_{SO(n)} \exp(\kappa \operatorname{trace}(O)) d\mu(O).$$

The domain $\mathcal{M} = (SO(n), \ldots, SO(n))$ is a manifold formed by a product of manifolds whose tangent space and metric are discussed in Section 9.4. The orthogonal projection to a tangent space of the manifold is

$$P_R^{\mathcal{M}} V = (P_{R_1}^{SO(n)} V_1, \ldots, P_{R_N}^{SO(n)} V_N),$$

where $V = (V_1, \ldots, V_N) \in \mathbb{R}^{n\times(nN)}$ and $P_{R_i}^{SO(n)} V_i = (V_i - R_i V_i^T R_i)/2$. For any $U \in \mathbb{R}^{n\times(nN)}$, let the U_k denote the matrix formed by $nk + 1$ to $n(k + 1)$ columns of U, i.e., U can be written as $U = (U_1, U_2, \ldots, U_N)$. The gradient of the cost function is given by

$$\operatorname{grad} L(R) = P_R^{\mathcal{M}} \operatorname{grad} \bar{L}(R),$$

where

$$(\operatorname{grad} \bar{L}(R))_k = \frac{1}{2} \sum_{(i,k)\in E} \omega_{ik} H_{ik}^T R_i + \frac{1}{2} \sum_{(k,j)\in E} \omega_{kj} H_{kj} R_j,$$

for all $(i, j) \in E$ and

$$\omega_{ij} = \frac{\kappa l_{ij}}{l_{ij} + 1 - p}, l_{ij} = \frac{p}{c_n(\kappa)} \exp(\kappa \operatorname{trace}(R_i^T H_{ij} R_j)).$$

The action of the Hessian on a tangent vector is

$$\begin{aligned} (\operatorname{Hess} L(R)[\xi])_k = \frac{1}{2} P_{R_k}^{\mathcal{M}} (\mathrm{D}(\operatorname{grad} \hat{L}(R))_k[\xi] - \xi_k (\operatorname{grad} \hat{L}(R))_k^T R_k \\ - R_k (\operatorname{grad} \hat{L}(R))_k^T \xi_k - R_k (\mathrm{D}(\operatorname{grad} \hat{L}(R))_k[\xi])^T R_k), \end{aligned}$$

where

$$\mathrm{D}(\operatorname{grad} \hat{L}(R))_k[\xi] = \frac{1}{2} \sum_{(i,k)\in E} (\omega_{ik} H_{ik}^T \xi_i + \omega'_{ik} H_{ik}^T R_i) + \frac{1}{2} \sum_{(k,j)\in E} (\omega_{kj} H_{kj} \xi_j + \omega'_{kj} H_{kj} R_j),$$

and

$$\omega'_{ij} = \frac{\kappa^2 (1 - p) l_{ij} (\operatorname{trace}(\xi_i^T H_{ij} R_j + R_i^T H_{ij} \xi_j))}{(l_{ij} + 1 - p)^2}.$$

12.2.3 Implementation and Results

Since the goal is to maximize the likelihood function (12.2.1) and the optimization algorithms are to minimize a cost function, the negative of the likelihood function is used to be the cost function.

Details of the implementation of the Riemannian objects needed are discussed in Chapter 9. The retraction and the vector transport are given by (10.2.16) and (9.2.19) since they are shown to be preferred in Sections 11.4.2 and 11.4.6 [1]. We use the intrinsic approach to represent a tangent vector. The algorithmic parameters settings are the same as those in Section 11.3.

The problem dimensions are $(n, p) = (3, 0.5)$, and $\kappa = 0.5$ in our experiments. The initial $R^{(0)} = \{R_1^{(0)}, R_2^{(0)}, \dots, R_N^{(0)}\}$ is chosen such that $R_i^{(0)} = \mathrm{qf}(M_i)$ where the elements of M_i are independently drawn from the standard normal distribution. The H_{ij} are taken to be $H_{ij} = \mathrm{qf}(N_{ij})$, where the elements of N_{ij} are independently drawn from the standard normal distribution. In order to obtain a connected graph, we use recursion. Suppose a k-node graph is connected. A $k+1$-node connected graph can be obtained by adding a node and a edge between $(k+1)$-th node and one of previous nodes. After completing an N-node connected graph, that is in fact a tree given the construction procedure, some edges are added randomly. Specifically, each pair of nodes (i, j) is checked. If an edge connecting i and j exists, the next pair is considered. If a connecting edge does not exist, then one is added with probability q.

Table 12.2 and Figure 12.2 show the results for $q = 1/N$ and Table 12.3 and Figure 12.3 show the results for $q = 0.5$. The number of edges of the former is linear in the number of nodes, while the number of edges in the latter is quadratic in the number of nodes.

RTR-Newton requires the smallest number of iteration since it exploits the Hessian. However, it is relatively slow due to the relatively high computational cost of the action of the Hessian on a tangent vector. From Tables 12.2 and 12.3, we can see that RBFGS and LRBFGS are the two fastest algorithms and RTR-SR1 is competitive when $N = 16$ or 32.

[1] The retraction (10.2.16) is identical to the exponential mapping of the orthogonal group but the vector transport by parallelization (9.2.19) is different from the parallel translation of the orthogonal group. The vector transport (9.2.19) is cheaper than the parallel translation.

Table 12.2: Comparison of RTR-Newton and Riemannian quasi-Newton algorithms for the synchronization of rotation problem with $q = 1/N$.

N		RTR	RBFGS	LRBFGS			RTR	LRTR-SR1		
		Newton		$m:2$	$m:4$	$m:8$	SR1	$m:2$	$m:4$	$m:8$
16	$iter$	21	85	89	89	75	100	193	154	176
	nf	21	89	102	95	82	100	193	154	176
	ng	20	87	91	91	77	100	193	154	176
	nH	134	168	0	0	0	438	0	0	0
	nV	0	254	440	782	1204	99	794	1077	2295
	nR	20	88	101	94	81	99	192	153	175
	gf_f	1.55_{-10}	4.35_{-7}	7.48_{-7}	5.89_{-7}	4.07_{-7}	5.66_{-7}	4.76_{-7}	7.74_{-7}	6.97_{-7}
	gf_f/gf_0	2.00_{-10}	5.62_{-7}	9.66_{-7}	7.61_{-7}	5.25_{-7}	7.31_{-7}	6.15_{-7}	9.99_{-7}	9.00_{-7}
	t	7.45_{-1}	3.48_{-1}	3.64_{-1}	3.61_{-1}	3.26_{-1}	3.81_{-1}	7.92_{-1}	6.71_{-1}	8.53_{-1}
32	$iter$	24	120	157	114	107	140	340	259	269
	nf	24	126	165	122	113	140	340	259	269
	ng	20	120	157	114	107	140	340	259	269
	nH	301	238	0	0	0	599	0	0	0
	nV	0	357	778	1005	1746	139	1449	1870	3402
	nR	23	125	164	121	112	139	339	258	268
	gf_f	4.60_{-10}	1.12_{-6}	1.13_{-6}	8.32_{-7}	1.01_{-6}	1.05_{-6}	8.11_{-7}	9.63_{-7}	8.36_{-7}
	gf_f/gf_0	4.08_{-10}	9.95_{-7}	9.99_{-7}	7.38_{-7}	8.95_{-7}	9.27_{-7}	7.20_{-7}	8.55_{-7}	7.42_{-7}
	t	4.82	1.11	1.36	1.00	9.61_{-1}	1.21	3.00	2.34	2.57
64	$iter$	40	234	301	307	266	391	1270	1072	1282
	nf	40	238	323	317	275	391	1270	1072	1282
	ng	33	235	302	308	267	391	1270	1072	1282
	nH	704	466	0	0	0	2897	0	0	0
	nV	0	700	1499	2743	4450	390	5323	7763	16857
	nR	39	237	322	316	274	390	1269	1071	1281
	gf_f	8.25_{-7}	1.52_{-6}	1.41_{-6}	1.31_{-6}	1.23_{-6}	1.40_{-6}	1.60_{-6}	1.56_{-6}	1.61_{-6}
	gf_f/gf_0	5.04_{-7}	9.29_{-7}	8.59_{-7}	8.00_{-7}	7.52_{-7}	8.56_{-7}	9.79_{-7}	9.50_{-7}	9.83_{-7}
	t	4.12_{1}	6.11	7.53	8.11	6.15	9.32	2.89_{1}	2.43_{1}	2.99_{1}

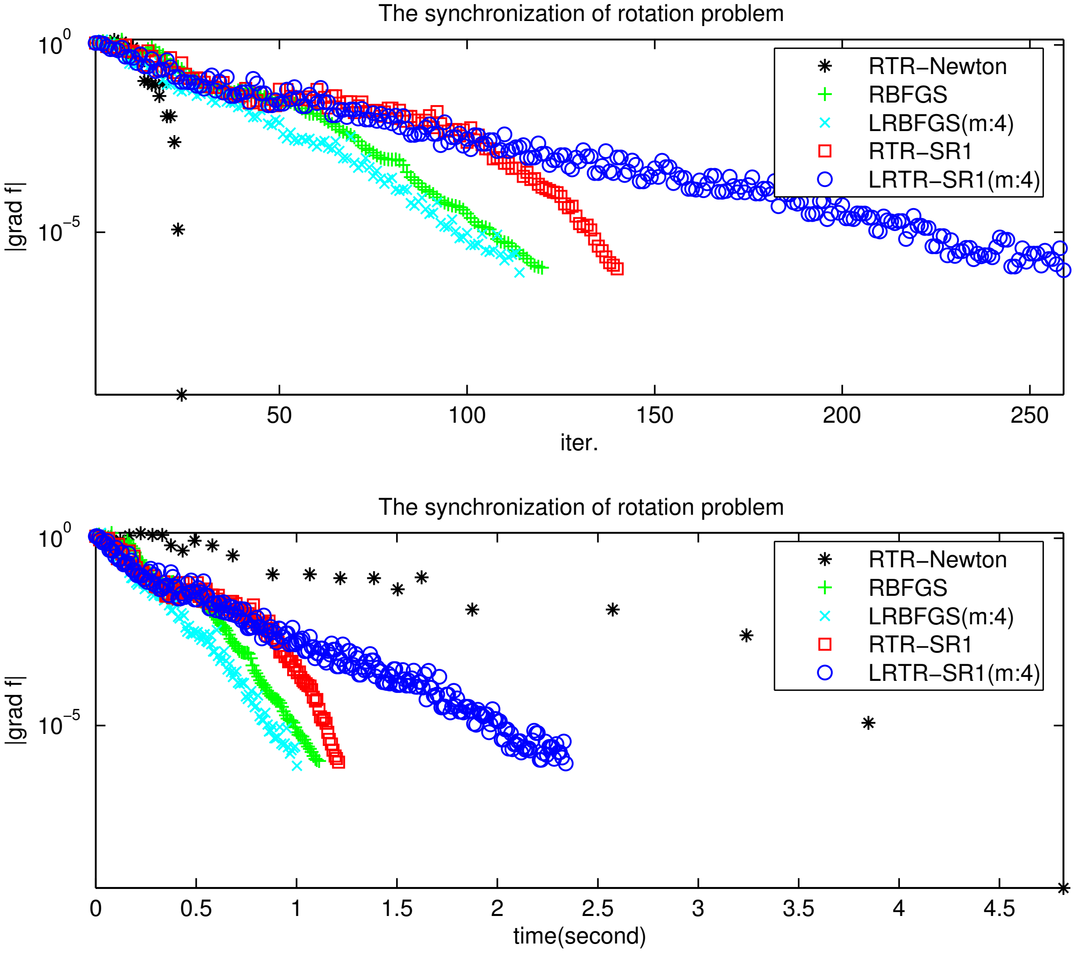


Figure 12.2: Comparison of RTR-Newton and Riemannian quasi-Newton algorithms for the synchronization of rotation problem with $q = 1/N, N = 32$.

Table 12.3: Comparison of RTR-Newton and Riemannian quasi-Newton algorithms for the synchronization of rotation problem with $q = 0.5$.

N		RTR	RBFGS	LRBFGS			RTR	LRTR-SR1		
		Newton		$m:2$	$m:4$	$m:8$	SR1	$m:2$	$m:4$	$m:8$
16	$iter$	17	47	51	49	48	48	73	63	62
	nf	17	51	56	53	54	48	73	63	62
	ng	16	47	51	49	48	48	73	63	62
	nH	73	92	0	0	0	134	0	0	0
	nV	0	138	248	420	743	47	286	434	759
	nR	16	50	55	52	53	47	72	62	61
	gf_f	1.35_{-7}	1.29_{-6}	1.25_{-6}	8.94_{-7}	8.61_{-7}	1.25_{-6}	3.50_{-7}	9.51_{-7}	1.45_{-6}
	gf_f/gf_0	8.92_{-8}	8.48_{-7}	8.21_{-7}	5.90_{-7}	5.68_{-7}	8.22_{-7}	2.31_{-7}	6.27_{-7}	9.57_{-7}
	t	5.44_{-1}	2.38_{-1}	2.51_{-1}	2.39_{-1}	2.51_{-1}	2.22_{-1}	3.57_{-1}	3.27_{-1}	3.47_{-1}
32	$iter$	25	71	125	71	71	105	161	131	140
	nf	25	75	127	73	73	105	161	131	140
	ng	23	71	125	71	71	105	161	131	140
	nH	154	140	0	0	0	350	0	0	0
	nV	0	210	618	618	1134	104	666	926	1811
	nR	24	74	126	72	72	104	160	130	139
	gf_f	7.39_{-9}	1.80_{-6}	2.20_{-6}	1.67_{-6}	2.02_{-6}	1.34_{-6}	2.21_{-6}	1.44_{-6}	2.39_{-6}
	gf_f/gf_0	2.98_{-9}	7.26_{-7}	8.87_{-7}	6.72_{-7}	8.14_{-7}	5.39_{-7}	8.91_{-7}	5.82_{-7}	9.63_{-7}
	t	3.90	9.64_{-1}	1.64	9.48_{-1}	9.64_{-1}	1.42	2.18	1.81	2.00
64	$iter$	42	166	152	120	221	225	351	262	311
	nf	42	181	162	124	247	225	351	262	311
	ng	35	166	152	120	224	225	351	262	311
	nH	309	330	0	0	0	803	0	0	0
	nV	0	495	753	1059	3687	224	1496	1913	4210
	nR	41	180	161	123	246	224	350	261	310
	gf_f	1.57_{-7}	4.15_{-6}	5.63_{-6}	3.97_{-6}	5.58_{-6}	2.21_{-6}	4.78_{-6}	5.66_{-6}	3.28_{-6}
	gf_f/gf_0	2.76_{-8}	7.30_{-7}	9.91_{-7}	6.98_{-7}	9.83_{-7}	3.88_{-7}	8.41_{-7}	9.96_{-7}	5.77_{-7}
	t	2.94_{1}	7.64	6.76	5.30	1.01_{1}	1.00_{1}	1.53_{1}	1.15_{1}	1.40_{1}

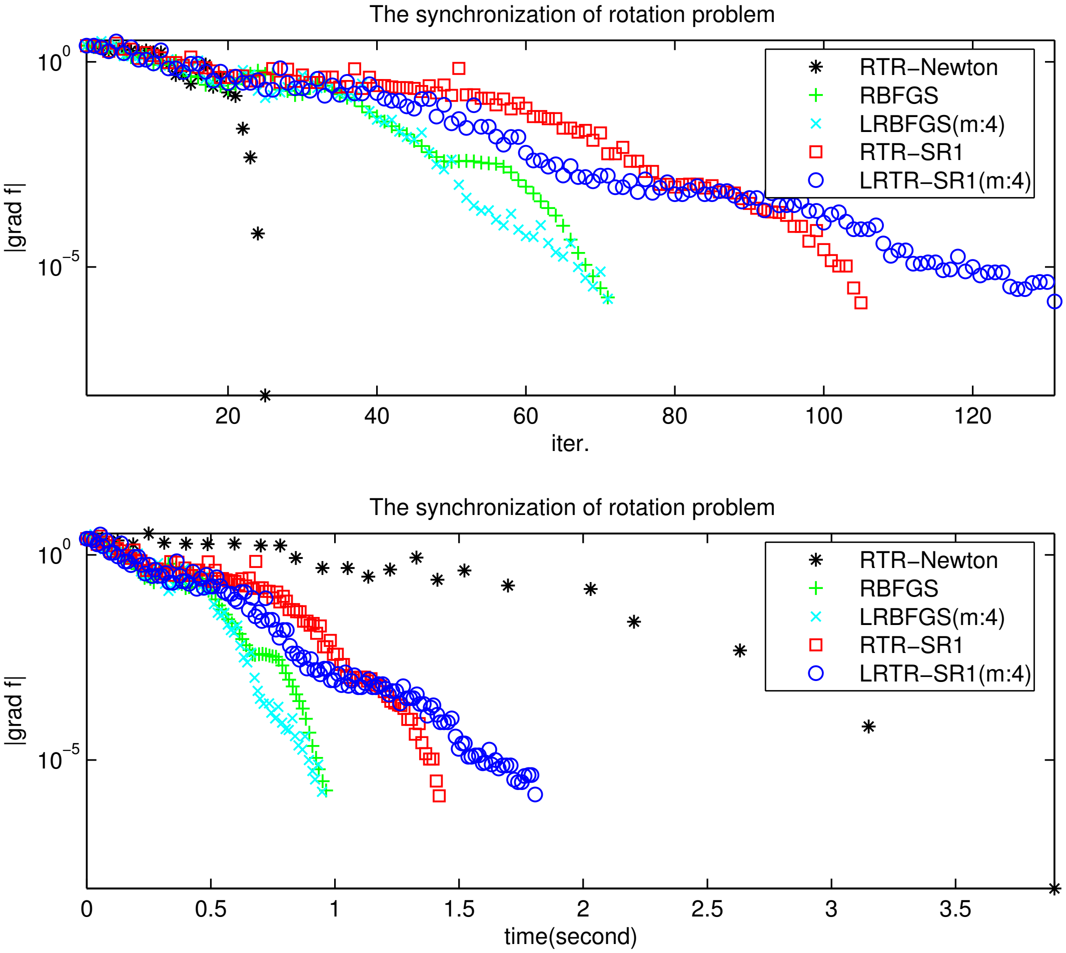


Figure 12.3: Comparison of RTR-Newton and Riemannian quasi-Newton algorithms for the synchronization of rotation problem with $q = 0.5, N = 32$.

CHAPTER 13

RIEMANNIAN OPTIMIZATION FOR ELASTIC SHAPE ANALYSIS

13.1 Introduction

Many approaches to shape analysis have been proposed in the literature and used to varying degrees of success in applications, e.g., point-based methods, domain-based shape representations and parameterized curve representations. Among parameterized curve representation methods, elastic shape analysis has become increasingly important in recent years due to its superior theoretical basis and empirically demonstrated effectiveness. In this chapter, we consider the framework of elastic shape analysis due to Srivastava et. al. [SKJJ11].

A fundamental operation in elastic shape analysis, upon which many other important tasks depend, is the accurate and efficient computation of distance between two curves. We develop and analyze a novel approach to determining optimal reparameterizations and rotations between two curves and evaluate its use in computing the distance and minimal geodesic between two curves.

This chapter is organized as follows. Section 13.2 presents the Riemannian framework for shape analysis including the definition of the elastic metric for open and closed curves in $\mathbb{R}^n$. Section 13.3 presents the algorithm of Srivastava et al. [SKJJ11], the approximations upon which it is based and its core dynamic programming algorithm. The proposed Riemannian approach to the solution of the optimization problem that defines the elastic distance metric evaluation is derived in Section 13.4 and a detailed discussion of its implementation using Riemannian optimization algorithms follows in Section 13.5. Empirical evaluation of the relative efficiency and effectiveness of the methods is presented in Section 13.6 and our conclusions are given in Section 13.7.

13.2 Riemannian Framework and Problem Statement

13.2.1 Curve Representation

The derivation of the basic representation of a shape begins with a parametrized curve, i.e., $\beta(t) : \mathbb{D} \to \mathbb{R}^n$, where $\mathbb{D}$ is the domain of the curve, $\mathbb{D} = [0, 1]$ for an open curve and $\mathbb{D} = \mathbb{S}^1$, i.e., the unit circle in $\mathbb{R}^2$, for a closed curve. The shape is taken to be invariant with respect to rescaling, translation, and rotation for inelastic shape analysis, while elastic shape analysis adds invariance with respect to reparameterization. All four invariants must be taken into account when developing a representation that supports efficient and robust computation.

The framework of Srivastava et al. [SKJJ11] uses the square root velocity (SRV) function

$$q(t) = \frac{\dot{\beta}(t)}{\sqrt{\|\dot{\beta}(t)\|_2}}$$

as the basis for elastic analysis of a shape defined by the parameterized curve $\beta(t)$. Observe that $\dot{\beta}(t)$ can be recovered from $q(t)$ by $\dot{\beta}(t) = \|q(t)\|_2 q(t)$. Translation is removed automatically by the use of $\dot{\beta}(t)$ in the definition. Rescaling is removed by the normalization of the length of the curve to 1. Since the length of a curve, $\beta(t)$, is $\int_{\mathbb{D}} \|\dot{\beta}(t)\|_2 dt = \int_{\mathbb{D}} \|q(t)\|_2^2 dt$, the normalization requires that $\int_{\mathbb{D}} \|q(t)\|_2^2 dt = 1$ and the set of all SRV functions is the unit sphere in $\mathbb{L}^2(\mathbb{D}, \mathbb{R}^n)$. This sphere is called the preshape space. For open curves in $\mathbb{R}^n$, the domain is $\mathbb{D} = [0, 1]$ and the preshape space

$$l_n^o = \{q : [0, 1] \to \mathbb{R}^n | \int_0^1 \|q(t)\|_2^2 dt = 1\},$$

is the unit sphere of $\mathbb{L}^2([0, 1], \mathbb{R}^n)$. For closed curves, the domain is $\mathbb{D} = S^1$ and the preshape space is

$$l_n^c = \{q : \mathbb{S}^1 \to \mathbb{R}^n | \int_{\mathbb{S}^1} \|q(t)\|_2^2 dt = 1, \int_{\mathbb{S}^1} q(t)\|q(t)\|_2 dt = 0\},$$

where $\int_{\mathbb{S}^1} q(t)\|q(t)\|_2 dt = 0$ is the closure condition.

Removing rotation and reparameterization is required to define the shape space. This is done by defining an appropriate quotient operation via isometric group actions. This, in turn, defines the distance between curves, the associated optimization problem, and other key tasks such as determining geodesics containing the two curves. Since the approaches taken differ for open and closed curves they are considered separately below. However, both approaches require the rotation and reparameterization groups, and their actions. In these two definitions, Γ and l_n are used to indicate the reparameterization group and preshape space for both open and closed curves. They are distinguished in later discussions by the addition of a superscript o and c respectively.

Definition 13.2.1. *The rotation group for curves in $\mathbb{R}^n$ is*

$$SO(n) = \{O \in \mathbb{R}^{n \times n} | O^T O = I_n, \det(O) = 1\}.$$

and its action is $SO(n) \times l_n \to l_n : (O, q) \to Oq$.

Definition 13.2.2. *The reparameterization group for curves in $\mathbb{R}^n$ is*

$$\Gamma = \{\gamma : \mathbb{D} \to \mathbb{D} | \gamma \in \mathcal{D}(\mathbb{D}, \mathbb{D})\}$$

and its action is $l_n \times \Gamma \to l_n : (q, \gamma) \to (q \circ \gamma)\sqrt{\dot{\gamma}}$, where $\mathcal{D}(\mathbb{D}, \mathbb{D})$ is the set of diffeomorphisms from $\mathbb{D}$ to itself (an invertible function such that both the function and its inverse are smooth, i.e., in C^∞).

13.2.2 Open Curves in $\mathbb{R}^n$

The preshape space for open curves, l_n^o, is a well-known infinite dimensional manifold. The tangent space of $q \in l_n^o$ is

$$\mathrm{T}_q\, l_n^o = \{v : [0, 1] \to \mathbb{R}^n | \int_0^1 q(t)^T v(t) dt = 0\}.$$

The Riemannian metric can be taken as the endowed metric from the embedding space $\mathbb{L}^2([0, 1], \mathbb{R}^n)$, i.e.,

$$\langle v_1, v_2 \rangle_{l_n^o} = \langle v_1, v_2 \rangle_{\mathbb{L}^2} = \int_0^1 v_1(t)^T v_2(t) dt,$$

where $v_1, v_2 \in \mathrm{T}_q\, l_n^o$ and the distance function on the manifold induced by this Riemannian metric is

$$d_{l_n^o}(x, y) = \cos^{-1} \langle x, y \rangle_{\mathbb{L}^2}. \tag{13.2.1}$$

The resulting shape space for open curves is given by the quotient

$$\mathfrak{L}_n^o = l_n^o/(SO(n) \times \Gamma^o) = \{[q] | q \in l_n^o\},$$

where $[q] = \{O(q \circ \gamma)\sqrt{\dot{\gamma}} | (O, \gamma(t)) \in SO(n) \times \Gamma^o, q \in l_n^o\}$ is the orbit.

If a compact Lie group G acts freely on a Riemannian manifold $\mathcal{M}$ by isometries, and the orbits are closed, then the quotient $\mathcal{M}/G$ is a manifold, and inherits a Riemannian metric from $\mathcal{M}$.

Srivastava et al. [SKJJ11] proved that $SO(n) \times \Gamma^o$ is an isometric group for l_n^o. However, as pointed by Robinson [Rob12] Γ^o is not a closed set since a sequence of γ's can have a limit that is flat on some nontrivial interval and is, therefore, not a diffeomorphism. In order to overcome this difficulty, we can work with the closure of the orbit, $\overline{[q]}$ and the shape space is approximated by

$$\mathfrak{L}_n^o \approx \{\overline{[q]} | q \in l_n^o\}.$$

(See [SKJJ11] for the technical details.) Another way to elaborate this to first introduce a semigroup:

$$\begin{aligned}\Gamma_s^o = \{\gamma : [0,1] \to [0,1] | \gamma(0) = 0, \gamma(1) = 1, \\ \gamma \text{ is an absolutely continuous, non-decreasing and surjective function } \} \ .\end{aligned}$$

It can be shown that Γ_s^o is closed under composition, $\overline{[q]}$ is the orbit of q under the semigroup Γ_s^o and Γ^o is dense in Γ_s^o.

Now we can define a distance between orbits of Γ_s^o, $\overline{[q_1]}$ and $\overline{[q_2]}$ as:

$$d_{\mathfrak{L}_n^o}(\overline{[q_1]}, \overline{[q_2]}) = \inf_{\gamma_1, \gamma_2 \in \Gamma_s^o, O \in SO(n)} d_{l_n^o}((q_1 \circ \gamma_1)\sqrt{\dot{\gamma}_1}, O(q_2 \circ \gamma_2)\sqrt{\dot{\gamma}_2}) \ .$$

Since Γ^o is dense in Γ_s^o, for any $\epsilon > 0$, there exists a $\gamma^* \in \Gamma^o$ and an $O^* \in SO(n)$ such that:

$$|d_{\mathfrak{L}_n^o}(\overline{[q_1]}, \overline{[q_2]}) - d_{l_n^o}(q_1, O^*(q_2 \circ \gamma^*)\sqrt{\dot{\gamma}^*})| < \epsilon \ .$$

Therefore, our goal is to find this pair $(O^*, \gamma^*) \in SO(n) \times \Gamma^o$ that solves for,

$$\inf_{\gamma \in \Gamma^o, O \in SO(n)} d_{l_n^o}(q_1, O(q_2 \circ \gamma)\sqrt{\dot{\gamma}}) = \inf_{\gamma \in \Gamma^o, O \in SO(n)} \cos^{-1} \langle q_1, O(q_2 \circ \gamma)\sqrt{\dot{\gamma}} \rangle_{\mathbb{L}^2}. \tag{13.2.2}$$

Even though this will not be an exact calculation of the shape distance, this approximation will serve as distance for comparing shapes of curves in practical situations.

13.2.3 Closed Curves in $\mathbb{R}^n$

The preshape space of closed curves, l_n^c, is a submanifold of l_n^o and the Riemannian metric inherited from the embedding space is

$$\langle v_1, v_2 \rangle_{l_n^c} = \langle v_1, v_2 \rangle_{\mathbb{L}^2} = \int_{\mathbb{S}^1} v_1(t)^T v_2(t) dt.$$

The resulting shape space is

$$\mathfrak{L}_n^c = l_n^c / (SO(n) \times \Gamma^c) = \{[q] | q \in l_n^c\},$$

where $[q] = \{O(q \circ \gamma)\sqrt{\dot{\gamma}} | (O, \gamma(t)) \in SO(n) \times \Gamma^c, q \in l_n^c\}$ is the orbit. As with open curves, $\mathfrak{L}_n^c$ is not known to be a Riemannian manifold, so it is approximated by a manifold

$$\mathfrak{L}_n^c \approx \{\overline{[q]} | q \in l_n^c\}.$$

Using the same idea as with open curves, a semigroup Γ^c_s, which is closure under composition and in which Γ^c is dense, is imposed and $\overline{[q]}$ is the orbit of q under the semigroup Γ^c_s.

The distance between orbits of Γ^c_s, $\overline{[q_1]}$ and $\overline{[q_2]}$, is

$$d_{\mathfrak{L}^c_n}(\overline{[q_1]}, \overline{[q_2]}) = \inf_{\gamma_1,\gamma_2 \in \Gamma^c_s, O \in SO(n)} d_{l^c_n}((q_1 \circ \gamma_1)\sqrt{\dot{\gamma}_1}, O(q_2 \circ \gamma_2)\sqrt{\dot{\gamma}_2}) \ .$$

and for any $\epsilon > 0$, there exists a $\gamma^* \in \Gamma^c$ and an $O^* \in SO(n)$ such that:

$$|d_{\mathfrak{L}^c_n}(\overline{[q_1]}, \overline{[q_2]}) - d_{l^c_n}(q_1, O^*(q_2 \circ \gamma^*)\sqrt{\dot{\gamma}^*})| < \epsilon \ .$$

Unlike the case of open curves, there is no known analytical expression of distance on l^c_n. Since l^c_n is a submanifold of l^o_n, the approximation

$$\begin{aligned} &\arg \min_{\gamma_1,\gamma_2 \in \Gamma^c_s, O \in SO(n)} d_{l^c_n}((q_1 \circ \gamma_1)\sqrt{\dot{\gamma}_1}, O(q_2 \circ \gamma_2)\sqrt{\dot{\gamma}_2}) \\ &\approx \arg \min_{\gamma_1,\gamma_2 \in \Gamma^c_s, O \in SO(n)} d_{l^o_n}((q_1 \circ \gamma_1)\sqrt{\dot{\gamma}_1}, O(q_2 \circ \gamma_2)\sqrt{\dot{\gamma}_2}) \end{aligned}$$

is used and our goal is to find $(\gamma^*, O^*) \in SO(n) \times \Gamma^c$ that solves for,

$$\begin{aligned} &\inf_{\gamma \in \Gamma^c, O \in SO(n)} d_{l^o_n}(q_1, O(q_2 \circ \gamma)\sqrt{\dot{\gamma}}) \\ = &\inf_{\gamma \in \Gamma^c, O \in SO(n)} \cos^{-1} \langle q_1(t), O(q_2 \circ \gamma(t))\sqrt{\dot{\gamma}(t)} \rangle_{\mathbb{L}^2} dt. \end{aligned} \tag{13.2.3}$$

As with open curves, this will not be an exact calculation of the shape distance, but this approximation will serve as distance for comparing shapes of curves in practical situations.

13.3 The Coordinate Relaxation Method

The discussion in Sections 13.2.2 and 13.2.3 characterizes the reparameterization problem from the Riemannian manifold point of view but does not suggest an algorithm. Sebastian et al. [SKK03] define an *edit distance* to characterize differences between shapes and develop an algorithm for closed curves with computational complexity $O(N^2 \log N)$. It requires the cost function to be invariant to rotation which is clearly not the case for the cost functions discussed above.

Srivastava et al. [SKJJ11] developed a method for finding (γ^*, O^*) for open and closed curves based on the idea of alternately optimizing on $SO(n)$ and Γ^c, i.e., a generalized Coordinate Relaxation method. The simpler open curve problem and algorithm are discussed first followed by the adaptation to closed curves.

13.3.1 The Basic Ingredients

For open curves Srivastava et al.[SKJJ11] use the cost function

$$H^o(O,\gamma(t)) = \int_0^1 \|q_1(t) - O(q_2 \circ \gamma(t))\sqrt{\dot{\gamma}(t)}\|_2^2 dt, \tag{13.3.1}$$

that has the same extreme points as the cost function used in (13.2.2). This is easily seen from

$$\begin{aligned}\int_0^1 \|q_1(t) - O(q_2 \circ \gamma(t))\sqrt{\dot{\gamma}(t)}\|_2^2 dt &= \langle q_1, q_1\rangle_{\mathbb{L}^2} + \langle q_2, q_2\rangle_{\mathbb{L}^2} - 2\langle q_1, O(q_2 \circ \gamma)\sqrt{\dot{\gamma}}\rangle_{\mathbb{L}^2} \\ &= 2 - 2\cos(\cos^{-1}(\langle q_1, O(q_2 \circ \gamma)\sqrt{\dot{\gamma}}\rangle_{\mathbb{L}^2})) \\ &= 2 - 2\cos(d_{l_n^o}(q_1, O(q_2 \circ \gamma)\sqrt{\dot{\gamma}})).\end{aligned}$$

They propose a variant of the general Coordinate Relaxation method approach given in Algorithm 8.

Algorithm 8 Coordinate Relaxation Algorithm for $H^o(O,\gamma)$

```
Input: Initial $\Gamma_0$;
1: $k = 0$;
2: Find $O_{k+1} = \arg\min_O H^o(O, \gamma_k)$;
3: Find $\gamma_{k+1} = \arg\min_\gamma H^o(O_{k+1}, \gamma)$;
4: If termination criterion is satisfied, stop, otherwise, $k = k + 1$ and go to step 2.
```

The minimizer O_{k+1} of $H^o(O,\gamma_k)$ is $O_{k+1} = UV^T$, where USV^T is the singular value decomposition (SVD) of $A = \int_{\mathbb{S}^1} q_1(t)\tilde{q}_2(t)^T dt$ and $\tilde{q}_2(t) = (q_2 \circ \gamma_k(t))\sqrt{\dot{\gamma}_k(t)}$. The SVD of a generic dense matrix $A \in \mathbb{R}^{n\times n}$ is well-understood and can be computed reliably and efficiently using well-known numerical linear algebra techniques for n up to several hundred, i.e., much larger than typically required for typical shape analysis problems. This is common to both open and closed curve problems. To find the minimizer γ_{k+1} of $H^o(O_k,\gamma)$ for open curves, Srivastava et. al. [SKJJ11] use dynamic programming (DP).

DP can be used to solve approximately optimization problems of the form,

$$\min_{\gamma\in\Gamma^o} \int_0^1 |f(t) - g(\gamma(t))|^2 dt,$$

where f and g are given sufficiently smooth functions. $H^o(O_{k+1},\gamma)$ is of this form and satisfies the additional necessary condition for applying DP that the cost function is additive in t.

The approximation arises for this problem because DP works on a grid in $[0,1] \times [0,1]$ rather than the continuous space Γ^o. Srivastava et al. use $G_N \times G_N$ where $G_N = \{0, 1/N, 2/N, \ldots, (N-1)/N, 1\}$. On $G_N \times G_N$, DP uses a partial cost function

$$E(s,t;\gamma) = \int_s^t |f(\tau) - g(\gamma(\tau))|^2 d\tau.$$

and determines a piecewise linear path defined by connecting points moving to the right and up, i.e., $(0,0) = (i_0, j_0),\ (i_1, j_1), (i_2, j_2) \ldots (i_m, j_m) = (1,1)$ where $(i_r, j_r) \in G_N \times G_N$ that minimizes the cost

$$\sum_{r=0}^{m-1} E(i_r, i_{r+1}; L(i_r, j_r; i_{r+1}, j_{r+1})),$$

where $L(i_r, j_r; i_{r+1}, j_{r+1})$ is a linear function passing though (i_r, j_r) and (i_{r+1}, j_{r+1}).

DP uses induction to construct a minimal path. Suppose $S \subseteq G_N \times G_N$ is such that for any $(p,q) \in S$ the global minimizing path $\gamma^*_{(p,q)}$ from $(0,0)$ to (p,q) and the associated cost function value $W(p,q)$ are known. Let $U_{i,j} \subseteq S$ denote the set $\{(p,q) | 0 \leq p < i, 0 \leq q < j\}$ where $i, j, p, q \in G_N$. The basic DP step adds (i,j) to S by computing $\gamma^*_{(i,j)}$, the global minimizing path on $G_N \times G_N$ from $(0,0)$ to (i,j), and the associated cost function value $W(i,j)$. This is done by considering each $(p,q) \in U_{i,j}$, adding the edge between (p,q) and (i,j) to the path $\gamma^*_{(p,q)}$ and determining its cost. Formally, determining $W(i,j)$ and $\gamma^*_{(i,j)}$ is solving

$$\min_{(k,l) \in U_{i,j}} E(k, i; L(k,l;i,j)) + W(k,l), \text{ with } W(0,0) = 0,$$

Eventually, $S = G_N \times G_N$ and a path with minimal cost on $S \subset \Gamma^o$ is given by $\gamma^*_{(1,1)}$.

The complexity of DP as described above is $O(N^4)$ and too high for practical problems. To reduce the complexity, the set $U_{i,j}$ is constrained to

$$\mathcal{N}_{i,j} = \{(k,l) | \max(i-h, 0) \leq k < i, \max(j-h, 0) \leq l < j\} \subset U_{i,j}, \tag{13.3.2}$$

for some h. The set $\mathcal{N}_{i,j}$ can be further reduced by removing some repeated slopes, e.g., $(i-2, j-2)$ is deleted because $(i-1, j-1)$ exists. Using the set $\mathcal{N}_{i,j}$ rather than $U_{i,j}$ reduces the complexity of DP to $O(N^2)$. However, since the number of slopes considered when adding (i,j) to S is constrained, the minimizer may change and may no longer be a global minimizer on $G_N \times G_N$.

The quality of $\gamma^*_{(1,1)}$ compared to a global minimizer, $\tilde{\gamma}^*_{(1,1)}$ on Γ^o is not known analytically nor is the potential further degradation in quality compared to $\tilde{\gamma}^*_{(1,1)}$ that results in replacing $U_{i,j}$ with

$\mathcal{N}_{i,j}$. Additionally, the path found by DP is piecewise linear and therefore not a diffeomorphism but it is a practical approximation of one.

The cost function defined on $SO(n) \times \Gamma^c$ for closed curves is

$$H^c(O, \gamma) = \int_{\mathbb{S}^1} \|q_1(t) - O(q_2 \circ \gamma(t))\sqrt{\dot{\gamma}(t)}\|_2^2 dt. \tag{13.3.3}$$

A DP-based Coordinate Relaxation algorithm cannot be applied to $H^c(O, \gamma)$ directly since DP requires a grid of a domain that is the cross product of two intervals, e.g., $[0, 1] \times [0, 1]$ rather than $S^1 \times S^1$. Srivastava et al. solve this by applying the open curve DP-based algorithm to a set of open curves, $\{\tilde{\beta}^{(i)},\ 1 \leq i \leq w\}$ derived from the closed curve β using w break points, t_i, $1 \leq i \leq w$, i.e.,

$$\tilde{\beta}^{(i)}(t) = \begin{cases} \beta(t + t_i), & \text{if } 0 \leq t \leq 1 - t_i; \\ \beta(t - (1 - t_i)), & \text{if } 1 - t_i < t \leq 1. \end{cases}$$

The open curve DP algorithm using $H^o(O, \gamma)$ is applied to each open curve $\tilde{\beta}^{(i)}$ to determine $\gamma^{(i)}$. A $\gamma^{(i)}$ with minimal cost is chosen as the closed curve reparameterization. Since DP is run w times, the complexity for this closed curve algorithm is $O(wN^2)$ and w is usually proportional to N, e.g., every second or third point is used as a break point, yielding $O(N^3)$ complexity. A key consideration for closed curve reparameterization is therefore computational complexity versus quality of γ.

13.3.2 Coordinate Relaxation Difficulties

The use of DP on a grid to solve approximately the optimization problem implies that γ is represented by a sequence of scalars such that the i-th scalar is γ at $(i - 1)/N$. The curves β_1 and β_2 are also represented discretely by a sequence of points in $\mathbb{R}^n$ and values at points other than the discrete set are recovered using an interpolatory parameterized polynomial, e.g., an interpolatory spline of degree 1, 2 or 3. The theoretical descriptions of the optimization algorithms for open and closed curves assume that the operations of rotation and reparameterization preserve the shape of the curves and it is important to maintain this invariant in the context of the discrete representations of γ, β_1 and β_2.

Algorithm 9 and Algorithm 10 are two discrete representation versions of the Coordinate Relaxation algorithm applied to closed curves based on the cost function (13.3.3). The open curve discrete versions are easily derived from either. The differences between Algorithm 9 and Algorithm

10 are specifically designed to highlight some crucial implementation decisions and the problems that arise in both implementations. These problems are all overcome by the new Riemannian algorithms we propose in Section 13.4.

Note that cost function (13.3.3) applies the reparameterization, γ to β_2. Also note that in Step 10 of Algorithm 9 interpolation is used when evaluating the reparameterized curve $\beta_2 \circ \gamma$. This implies that the vector of discrete points in $\mathbb{R}^n$ used to represent β_2 is updated by each reparameterization. If, equivalently from an optimization point of view, γ is associated with β_1 then its vector changes. Therefore, when multiple iterations of Coordinate Relaxation are performed, a problem arises. Since the points upon which the interpolatory parameterized polynomial is based change, the parameterized polynomial changes and therefore the shape of the curve changes with each reparameterization. Algorithm 10 overcomes this difficulty by representing β_2 as a continuous function determined by the interpolatory parameterized polynomial (Step 1) and maintaining it throughout the algorithm.

Algorithm 10, however, has a problem that is not seen in Algorithm 9. In Step 11 the expression

$$\bar{\beta}_2^{(\min,k)} = O^{(\min,k)} O_*^{(k)} (\beta_2 \circ \gamma_*^{(k)}) \circ ((\gamma^{(\min,k)} + b_{\min}^{(k)}/N) \mod 1). \tag{13.3.4}$$

is implicitly used when $H^{(\min,k)} = H^c(O_*^{(k)}, \gamma_*^{(k)})$ is computed and is then explicitly used to compute $\bar{\beta}_2^{(\min,k)}$. The curves $\bar{\beta}_2^{(\min,k)}$ and $\beta_2^{(k+1)}$ are, in theory, the same. However, on the next iteration, $k+1$, the curve $\beta_2^{(k+1)}$ is explicitly computed using the composition

$$\beta_2^{(k+1)} = O^{(\min,k)} O_*^{(k)} \beta_2 \circ (\gamma_*^{(k)} \circ ((\gamma^{(\min,k)} + b_{\min}^{(k)}/N) \mod 1)). \tag{13.3.5}$$

Note that associativity has been applied in the composition of functions. This is required given that the interpolatory parameterized polynomial representing β_2 is maintained for all iterations. The change of order does not matter theoretically in the continuous form but the curves are different in the discrete case. If the cost function value $H^{(\min,k)}$ was computed using the order of composition in (13.3.5) it may yield a different value than the cost function value used during iteration k to update β_2 mentioned above. In fact, the cost function value associated with the form (13.3.5) implicit in iteration $k+1$ may be larger than the cost function value actually computed in iteration k using (13.3.4). Therefore, we may compute a $\beta_2^{(k+1)}$ that does not decrease the cost function value in practice.

The experiments in Srivastava et. al. [SKJJ11] simplify the optimization considerably by using only a single iteration of the Coordinate Relaxation algorithm. Algorithm 9 and 10 are then identical and avoid both of these problems. If a more accurate optimization is demanded therefore requiring more iterations, as done in Section 13.6, problems ensue. Note that these problems are not the result of using DP to approximate the optimization problem. Rather, they arise from the Coordinate Relaxation approach. The new Riemannian algorithm discussed in Section 13.4 avoids these difficulties.

Algorithm 9 Coordinate Relaxation Algorithm 1 for $H^c(O, \gamma)$

```
Input: Two closed curves $\beta_1 = \{v_1, v_2, \dots, v_N, v_1\}$ and $\beta_2^{(0)} = \{u_1^{(0)}, u_2^{(0)}, \dots, u_N^{(0)}, u_1^{(0)}\}$ where
    $u_i^{(0)}, v_i \in \mathbb{R}^2$; a set of break points $\{b_1, b_2, \dots, b_w\}$;
1: $k = 0$;
2: for $i = 1, 2, \dots, w$ do
3:     Shift $\beta_2^{(k)}$ and get $\tilde{\beta}_2^{(i,k)} = \{u_{b_i}^{(k)}, u_{b_i+1}^{(k)}, \dots, u_N^{(k)}, u_1^{(k)}, \dots, u_{b_i}^{(k)}\}$;
4:     Compute the rotation $O^{(i,k)}$ based on $\beta_1$ and $\tilde{\beta}_2^{(i,k)}$;
5:     Set $\bar{\beta}_2 = O^{(i,k)} \tilde{\beta}_2^{(i,k)}$;
6:     Compute $\gamma^{(i,k)}$ for $\beta_1$ and $\bar{\beta}_2$ by DP;
7:     Compute cost function $H^{(i,k)}$
8: end for
9: Find $H^{(\min,k)} = \min_{1 \leq i \leq w}\{H^{(i,k)}\}$ and get the corresponding $O^{(\min,k)}$, $\gamma^{(\min,k)}$ and $\bar{\beta}_2^{(\min,k)}$;
10: Interpolate points $\bar{\beta}_2^{(\min,k)}$ to get a function, e.g., spline cubic function and get $\beta_2^{(k+1)}$ by e-
    valuating the function at $\gamma^{(\min,k)}$; (This is the implementation of $\beta_2^{(k+1)} = \bar{\beta}_2^{(\min,k)} \circ \gamma^{(\min,k)}$);

11: If a stopping criterion is satisfied, then stop, otherwise $k = k + 1$ and goto step 2;
```

13.4 A Riemannian Optimization Method

In order to make use of Riemannian optimization theory and algorithms in the fundamental elastic shape analysis task of efficiently and effectively computing the distance between two curves, we must define an appropriate cost function on a Riemannian manifold, the Riemannian gradient of the cost function, the retraction operation on the manifold, and an appropriate vector transport. Several Riemannian optimization algorithms are applicable to the distance computation and a representative set is investigated and compared to the DP-based approach of Srivastava et al. for closed curves in this and the following section. Specifically, the algorithms RTR-SR1, LRTR-SR1,

Algorithm 10 Coordinate Relaxation Algorithm 2 for $H^c(O, \gamma)$

```
Input: Two closed curves $\beta_1 = \{v_1, v_2, \ldots, v_N, v_1\}$ and $\beta_2 = \{u_1, u_2, \ldots, u_N, u_1\}$ where $u_i, v_i \in \mathbb{R}^2$;
    a set of break points $\{b_1, b_2, \ldots, b_w\}$; initial $\gamma_*^{(0)} = \{0, 1/n, \ldots, 1\}$; $O_*^{(0)} = I_n$;
1:  Compute interpolation function $F_{\beta_2}$ for $\beta_2$, e.g., a spline cubic function;
2:  $k = 0$;
3:  Compute $\beta_2^{(k)}$ by evaluating $F_{\beta_2}$ at $\gamma_*^{(k)}$ and left multiplying by $O_*^{(k)}$;
4:  for $i = 1, 2, \ldots, w$ do
5:      Shift $\beta_2^{(k)}$ and get $\tilde{\beta}_2^{(i,k)} = \{u_{b_i}^{(k)}, u_{b_i+1}^{(k)}, \ldots, u_N^{(k)}, u_1^{(k)}, \ldots, u_{b_i}^{(k)}\}$;
6:      Compute the rotation $O^{(i,k)}$ based on $\beta_1$ and $\tilde{\beta}_2^{(i,k)}$;
7:      Set $\bar{\beta}_2 = O^{(i,k)} \tilde{\beta}_2^{(i,k)}$;
8:      Compute $\gamma^{(i,k)}$ for $\beta_1$ and $\bar{\beta}_2$ by DP;
9:      Compute cost function $H^{(i,k)}$;
10: end for
11: Find $H^{(\min,k)} = \min_{1 \leq i \leq w}\{H^{(i,k)}\}$ and get the corresponding $O^{(\min,k)}$, $\gamma^{(\min,k)}$, $\bar{\beta}_2^{(\min,k)}$ and the
    shift $b_{\min}^{(k)}$;
12: Set $O_*^{(k+1)} = O^{(\min,k)} O_*^{(k)}$;
13: Interpolate points $\gamma_*^{(k)}$ to get a function, e.g., spline function and evaluate the function at
    $(\gamma^{(\min,k)} + i_{\min}/N) \mod 1 = \begin{pmatrix} (\gamma^{(\min,k)}(0) + b_{\min}^{(k)}/N) \mod 1 \\ (\gamma^{(\min,k)}(1/N) + b_{\min}^{(k)}/N) \mod 1 \\ \vdots \\ (\gamma^{(\min,k)}(1) + b_{\min}^{(k)}/N) \mod 1 \end{pmatrix}$
    to get $\gamma_*^{(k+1)}$; (This is the implementation of $\gamma_*^{(k+1)} = \gamma_*^{(k)} \circ \gamma^{(\min,k)}$);
14: If a stopping criterion is satisfied, then stop, otherwise $k = k + 1$ and goto step 3;
```

RBFGS, LRBFGS and RSD are applied to the distance problem and it is shown that a Riemannian approach is more efficient computationally and produces a superior distance computation than the DP-based approach.

13.4.1 Cost Function

Using the Riemannian approach we can handle the closed curve distance problem directly, i.e., breaking the curve into several open curves and taking the minimal solution is avoided. The first step in defining the cost function and associated Riemannian manifold requires reconsidering the representation of Γ^c for closed curves

$$\Gamma^c = \{\gamma : \mathbb{S}^1 \to \mathbb{S}^1 | \gamma \text{ is a diffeomorphism.}\}.$$

and its group action. We use $\tilde{\Gamma} \times \mathbb{R}$ which is a covering space of Γ^c with the covering mapping from $(\gamma(t), m))$ to $\gamma(t) + m \bmod 2\pi$, where

$$\tilde{\Gamma} = \{\gamma : [0, 2\pi] \to [0, 2\pi] | \gamma \text{ is diffeomorphism. }\}.$$

The Γ^c group action on q

$$(q, \gamma) = q \circ \gamma \sqrt{\dot{\gamma}}, \;\; \gamma \in \Gamma^c$$

is replaced with the $\tilde{\Gamma} \times \mathbb{R}$ group action on q defined by

$$(q, (\gamma, m)) = (q(\gamma + m \bmod 2\pi)) \sqrt{\dot{\gamma}}, \;\; (\gamma, m) \in \tilde{\Gamma} \times \mathbb{R}.$$

Note that the addition of $\mathbb{R}$ to the group definition removes the need for break points since the offset has been added as a decision variable.

The cost function on the Riemannian manifold $SO(n) \times \mathbb{R} \times \tilde{\Gamma}$ is

$$H(O, m, \gamma) = \int_0^{2\pi} \|q_1(t) - O(q_2(\gamma(t) + m \mod 2\pi)) \sqrt{\dot{\gamma}(t)}\|_2^2 dt$$

Using the Riemannian manifold structure of $\tilde{\Gamma}$ complicates the basic objects required for a Riemannian optimization algorithm. The tangent space of $\gamma \in \tilde{\Gamma}$ is $\mathrm{T}_\gamma \tilde{\Gamma} = \{v : [0, 2\pi] \to \mathbb{R} | v(0) = v(2\pi) = 0, v \text{ is smooth}\}$. Note that $\tilde{\Gamma}$ is an open subset of $\mathbb{L}^2([0, 2\pi], \mathbb{R})$, it is natural to endowed the metric from $\mathbb{L}^2([0, 2\pi], \mathbb{R})$. Therefore, the exponential mapping is given by $\mathrm{Exp}_\gamma v = \gamma + v$. When the exponential mapping is used as the retraction in a Riemannian optimization algorithm, the update tangent vector η_i in the update $\gamma_{i+1} = \gamma_i + \eta_i$ must be carefully chosen to guarantee that γ_{i+1} is a diffeomorphism. This is, in general, not easy to guarantee for an arbitrary γ_i on the manifold and we know of no retractions to replace the exponential map that can guarantee a diffeomorphism in a computationally efficient manner. Fortunately, we can approximate $\tilde{\Gamma}$ with another manifold that provides the required computational efficiency while capturing the structure of $\tilde{\Gamma}$ and Γ^c effectively.

Note that any $\gamma \in \tilde{\Gamma}$ and its derivative $\dot{\gamma}$ satisfy the constraints $\gamma(0) = 0, \gamma(2\pi) = 2\pi$ and $\dot{\gamma}(s) > 0$ for all $s \in [0, 2\pi]$. These are equivalent to $\gamma(0) = 0$, $\int_0^{2\pi} \dot{\gamma}(t)dt = 2\pi$ and $\dot{\gamma}(s) > 0$ for all $s \in [0, 2\pi]$. The positivity constraint on the derivative can be guaranteed by replacing $\dot{\gamma}$ with an even power function. For example setting $\dot{\gamma} = l^2$ where $l : [0, 2\pi] \to \mathbb{R}$ yields all of the γ that satisfy the constraints. All three constraints are condensed into the constraints $\int_0^{2\pi} l^2(t)dt = 2\pi$ and $l^2(s) \neq 0$ for all $s \in [0, 2\pi]$. (The method of keeping l^2 away from 0 is discussed in Section 13.5.2.) Therefore, l is an element of the 2-norm sphere and $\gamma(t)$ can be recovered by $\int_0^t l^2(s)ds$. It follows that $\sqrt{\dot{\gamma}} = |l|$ and the cost function becomes

$$L(O, m, l) = \int_0^{2\pi} \|Oq_1(t) - q_2(\int_0^t l^2(s)ds + m \mod 2\pi)|l(t)|\|_2^2 dt.$$

While this approach simplifies the constraints considerably, the resulting cost function is only partly smooth due to the present of $|l(t)|$ and does not satisfy the C^2 smoothness assumption required for the Riemannian quasi-Newton algorithms and other superlinearly convergent Riemannian optimization algorithms.

An alternative is to let $l^4 = \dot{\gamma}$. Therefore,

$$l \in \mathcal{L} = \{l : [0, 2\pi] \to \mathbb{R} | \int_0^{2\pi} l^4(t)dt = 2\pi\}.$$

Therefore, l is an element of 4-norm sphere and the cost function becomes

$$\int_0^{2\pi} \|q_1(t) - Oq_2(\int_0^t l^4(s)ds + m \mod 2\pi)l^2(t)\|_2^2 dt$$

which is defined on $SO(n) \times \mathbb{R} \times \mathcal{L}$. Exploiting the invariance of the norm under isometry, we use the equivalent cost function

$$L(O, m, l) = \int_0^{2\pi} \|Oq_1(t) - q_2(\int_0^t l^4(s)ds + m \mod 2\pi)l^2(t)\|_2^2 dt.$$

The reason we put O on $q_1(t)$ will be discussed in Section 13.5.1.

13.4.2 The Riemannian Manifold

The Riemannian manifold used to define the constraints for the optimization problem associated with the efficient algorithm to compute the distance function for elastic shape analysis is $SO(n) \times \mathbb{R} \times \mathcal{L}$. The Riemannian gradient of the cost function, the retraction operation on the manifold, and an appropriate vector transport, as discussed in Chapter 9, can be constructed by considering each on the components of the product.

$SO(n)$ is a well-known Riemannian manifold the structure of which is discussed in the literature [AMS08] and the associated implementation issues are considered in Section 10.5.

$\mathcal{L}$ is an infinite dimension Riemannian manifold. The tangent space of $\mathcal{L}$ at any point is therefore an infinite dimensional linear space with elements that are functions defined on $[0, 2\pi]$. The following lemma characterizes, $\mathrm{T}\,\mathcal{L}$, the tangent bundle of $\mathcal{L}$ and an element of a tangent space.

Lemma 13.4.1. *The tangent space $T_l\mathcal{L}$ of $l \in \mathcal{L}$ is*

$$\mathrm{T}_l\,\mathcal{L} = \{v : [0, 2\pi] \to \mathbb{R} | \langle l^3, v\rangle_{\mathbb{L}^2} = 0\},$$

and therefore the projection onto the tangent space is

$$P_l(v) = v - l^3 \frac{\langle v, l^3\rangle_{\mathbb{L}^2}}{\langle l^3, l^3\rangle_{\mathbb{L}^2}}$$

Proof. By definition of a tangent vector, $\forall v \in \mathrm{T}_l\,\mathcal{L}$, we can find a smooth curve $C(s)$, $s \in [0, 1]$ on $\mathcal{L}$ such that $\frac{dC}{ds}(0) = v$ and $C(0) = l$. Let $F_c(t, s)$ denote $C(s)$, and notice that for a fixed s, $C(s)$ is in $\mathcal{L}$. Since $\mathcal{L}$ is the 4-norm sphere, we have

$$\int_0^{2\pi} F_c^4(t, s)dt = 2\pi.$$

Taking a derivative with respect to s, we obtain

$$\int_0^{2\pi} 4F_c^3(t, s)\frac{\partial F_c(t, s)}{\partial s}dt = 0.$$

When $s = 0$, we have

$$0 = \int_0^{2\pi} F_c^3(t, 0)\frac{\partial F_c(t, 0)}{\partial s}dt = \int_0^{2\pi} C^3(0)\frac{dC}{ds}(0)dt = \int_0^{2\pi} l^3 v dt$$
$$\therefore \langle l^3, v\rangle_{\mathbb{L}^2} = 0$$

Since $\mathcal{L} \subset \mathbb{L}^2([0, 2\pi], \mathbb{R})$ and $\mathrm{T}_l\,\mathcal{L}$ has only one constraint, $l^3/\sqrt{\langle l^3, l^3\rangle_{\mathbb{L}^2}}$ is an orthonormal basis of the normal space of $\mathcal{L}$ at l and we obtain the projection given in the Lemma. □

Let the metric of $\mathcal{L}$ be endowed from the embedding space $\mathbb{L}^2([0, 2\pi], \mathbb{R})$. Analytical forms of the exponential mapping and parallel vector transport on $\mathcal{L}$ are unknown. However, an efficient retraction and isometric vector transport can be constructed. They are characterized in the following Lemma.

Lemma 13.4.2. *The function* $R_l(v) : \mathcal{L} \times \mathrm{T}_l\,\mathcal{L} \to \mathcal{L}$

$$R_l(v) = (2\pi)^{1/4} \frac{l+v}{\|l+v\|_{\mathbb{L}^4}}$$

defines a retraction on $\mathcal{L}$ *and the associated differentiated retraction is*

$$\mathcal{T}_{R_v} u = (2\pi)^{1/4} \frac{u}{\|l+v\|_{\mathbb{L}^4}} - (2\pi)^{1/4} \frac{(l+v)\langle (l+v)^3, u\rangle_{\mathbb{L}^2}}{\|l+v\|_{\mathbb{L}^4}^5},$$

where $\|\cdot\|_{\mathbb{L}^4} = (\int_0^{2\pi} (\cdot)^4 dt)^{1/4}$ *and* $u, v \in \mathrm{T}_l\,\mathcal{L}$. *An isometric vector transport is given by*

$$\mathcal{T}_{S_v} u = u - \frac{2(\tilde{l}_1 + \tilde{l}_2)\langle \tilde{l}_2, u\rangle_{\mathbb{L}^2}}{\|\tilde{l}_1 + \tilde{l}_2\|_{\mathbb{L}^2}^2}.$$

where $\tilde{l}_1 = l^3 / \langle l^3, l^3\rangle_{\mathbb{L}^2}$, $\tilde{l}_2 = l_2^3 / \langle l_2^3, l_2^3\rangle_{\mathbb{L}^2}$, $l_2 = R_l(v)$.

Proof. Define

$$\Phi : \mathcal{L} \times \mathbb{R}_+ \to \mathbb{L}^2([0, 2\pi], \mathbb{R}) - \{0\} : (l(t), r) \mapsto l(t) r,$$

and

$$\pi_1 : \mathcal{L} \times \mathbb{R}_+ \to \mathcal{L} : (l(t), r) \mapsto l(t),$$

where $\mathbb{R}_+$ denotes the set of positive number. By [AMS08, Proposition 4.1.2], we have $\pi_1(\Phi^{-1}(l+v))$ is a retraction, i.e.,

$$R_l(v) = (2\pi)^{1/4} \frac{l+v}{\|l+v\|_{\mathbb{L}^4}}$$

By the definition of differentiated retraction,

$$\begin{aligned} \mathcal{T}_{R_v} u = \frac{dR(v+tu)}{dt}|_{t=0} &= (2\pi)^{1/4} \frac{d}{dt} \left(\frac{l+v+tu}{\|l+v+tu\|_{\mathbb{L}^4}} \right) |_{t=0} \\ &= (2\pi)^{1/4} \frac{u}{\|l+v\|_{\mathbb{L}^4}} - (2\pi)^{1/4} \frac{(l+v)\langle (l+v)^3, u\rangle_{\mathbb{L}^2}}{\|l+v\|_{\mathbb{L}^4}^5}. \end{aligned}$$

It is easily verified that the proposed isometric vector transport satisfies the three conditions of the Definition 1.2.3 of vector transports and smoothness. The isometry is a consequence of the isometry property of Householder reflectors. □

For the cost function of interest an analytical form of the Riemannian gradient can be derived. It is given in the following Lemma.

Lemma 13.4.3. *The Riemannian gradient of the cost function*

$$L(O,m,l) = \int_0^{2\pi} \|Oq_1(t) - q_2(\int_0^t l^4(s)ds + m \mod 2\pi)l^2(t)\|_2^2 dt, \ \ (O,m,l) \in SO(n) \times \mathbb{R} \times \mathcal{L}$$

is

$$\begin{aligned}&\nabla L(O,m,l) = \\ &(P_O(-2\int_0^{2\pi} q_2(\int_0^t l^4(s)ds + m \mod 2\pi)l^2(t)q_1(t)^T dt), \\ &-2\int_0^{2\pi} \langle Oq_1(t) - l^2(t)q_2(\int_0^t l^4(s)ds + m \mod 2\pi), l^2(t)q_2'(\int_0^t l^4(s)ds + m \mod 2\pi)\rangle_2 dt, \\ &P_l(2y(t)l^3(t) - 2x(t)))\end{aligned}$$

where $P_O U = (U - OU^T O)/2$ is the projection to $\mathrm{T}_O\, SO(n)$,

$$x(t) = \langle Oq_1(t) - l^2(t)q_2(\int_0^t l^4(s)ds + m \mod 2\pi), 2l(t)q_2(\int_0^t l^4(s)ds + m \mod 2\pi)\rangle_2,$$

and $y(t)$ is any primitive of

$$y'(t) = \langle Oq_1(t) - l^2(t)q_2(\int_0^t l^4(s)ds + m \mod 2\pi), 4l^2(t)q_2'(\int_0^t l^4(s)ds + m \mod 2\pi)\rangle_2.$$

Proof. Consider the cost function defined on the embedding manifold

$$\bar{L}(O,m,l) : \mathbb{R}^{n\times n} \times \mathbb{R} \times \mathbb{L}^2 \to \mathbb{R} : (O,m,l) \mapsto \int_0^{2\pi} \|Oq_1(t) - q_2(\int_0^t l^4(s)ds + m \mod 2\pi)l^2(t)\|_2^2 dt.$$

For simplicity of notation, let $\tilde{q}_2(l,m)$ denote $q_2(\int_0^t l^4(s)ds + m \mod 2\pi)$ and $\tilde{q}_2'(l,m)$ denote $q_2'(\int_0^t l^4(s)ds + m \mod 2\pi)$. The gradient for the variable O is

$$\operatorname{grad}_O \bar{L}(O,m,l) = -2\int_0^{2\pi} \tilde{q}_2(l,m)l^2(t)q_1(t)^T dt \in \mathbb{R}^{n\times n}.$$

The gradient for the variable m is

$$\operatorname{grad}_m \bar{L}(O,m,l) = -2\int_0^{2\pi} \langle Oq_1(t) - l^2(t)\tilde{q}_2(l,m), l^2(t)\tilde{q}_2'(l,m)\rangle_2 dt.$$

The gradient for the variable l is not easy to compute directly. First, consider the directional derivative along $v \in \mathrm{T}_l\, \mathcal{L}$.

$$\mathrm{D}_l\, \bar{L}(O,m,l)[v] = -2\int_0^{2\pi} \langle Oq_1(t) - \tilde{q}_2(l,m)l^2(t), 2l(t)v(t)\tilde{q}_2(l,m) + 4l^2(t)\tilde{q}_2'(l,m)\int_0^t l^3(s)v(s)ds\rangle_2 dt.$$

Simplifying, we have

$$\begin{aligned}\mathrm{D}_l\,\bar{L}(O,m,l)[v] = &-2\int_0^{2\pi} \langle Oq_1(t) - \tilde{q}_2(l,m)l^2(t), 2l(t)\tilde{q}_2(l,m)\rangle_2 v(t)dt\\ &- 2\int_0^{2\pi} \langle Oq_1(t) - \tilde{q}_2(l,m)l^2(t), 4l^2(t)\tilde{q}_2'(l,m)\rangle_2 \int_0^t l^3(s)v(s)dsdt\end{aligned}$$

If

$$\begin{aligned}x(t) &= \langle Oq_1(t) - l^2(t)\tilde{q}_2(l,m), 2l(t)\tilde{q}_2(l,m)\rangle_2\\ y'(t) &= \langle Oq_1(t) - l^2(t)\tilde{q}_2(l,m), 4l^2(t)\tilde{q}_2'(l,m)\rangle_2.\end{aligned}$$

then

$$\begin{aligned}&\mathrm{D}_l\,\bar{L}(O,m,l)[v]\\ &= -2\int_0^{2\pi} x(t)v(t)dt - 2\int_0^{2\pi} y'(t)\int_0^t l^3(s)v(s)dsdt\\ &= -2\int_0^{2\pi} x(t)v(t)dt - 2(y(t)\int_0^t l^3(s)v(s)ds|_0^{2\pi} - \int_0^{2\pi} y(t)l^3(t)v(t)dt) \text{ (integration by part)}\\ &= -2\int_0^{2\pi} x(t)v(t)dt + 2\int_0^{2\pi} y(t)l^3(t)v(t)dt \text{ (by } v \in \mathrm{T}_l\,\mathcal{L})\\ &= \int_0^{2\pi} (2y(t)l^3(t) - 2x(t))v(t)dt\\ &= \langle 2y(t)l^3(t) - 2x(t), v(t)\rangle_{\mathbb{L}^2}.\end{aligned}$$

Since the gradient is the vector that satisfies

$$D_l\bar{L}(O,m,l)[v] = \langle \nabla_l\bar{L}(O,m,l), v(t)\rangle_{\mathbb{L}^2},$$

we obtain

$$\mathrm{grad}_l\,\bar{L}(O,m,l) = 2y(t)l^3(t) - 2x(t).$$

Therefore, the gradient of $\bar{L}(O,m,l)$ is

$$\begin{aligned}&\mathrm{grad}\,\bar{L}(O,m,l)\\ =&(-2\int_0^{2\pi} q_2(\int_0^t l^4(s)ds + m \bmod 2\pi)l^2(t)q_1(t)^T dt,\\ &-2\int_0^{2\pi} \langle Oq_1(t) - l^2(t)q_2(\int_0^t l^4(s)ds + m \bmod 2\pi), l^2(t)q_2'(\int_0^t l^4(s)ds + m \bmod 2\pi)\rangle_2 dt,\\ &2y(t)l^3(t) - 2x(t))\end{aligned}$$

Finally, the Riemannian gradient is given by projecting each component of $\bar{L}(O, m, l)$ to its associated manifold. □

13.5 Implementation Comments

13.5.1 Representation and Cost Function

We assume all of the curves are continuous In practice, all of the curves are represented by a set of points and therefore, the q-function of a curve $\beta(t)$ is also represented by points that are on some smooth function. Since O is an isometry in the cost functions, we can apply it to either q_1 or q_2. We apply it to q_1. Therefore representing q_1 does not require the use of an interpolatory function and a vector of points is sufficient.

In all of the cost functions considered, q_2 is composed with some function and, therefore, representing q_2 as a set of points is not sufficient. A suitable function must be used. Since the convergence analysis of Riemannian quasi-Newton optimization algorithms requires a C^2 cost function, an interpolatory cubic spline of the set of points on q_2 is used but a spline of degree 1, i.e., piecewise linear, or degree 2 are also practical.

It should be noted however that there is nothing in the formulation that requires an interpolatory approximation. The discrete points in the representation could be control points for a continuous approximating parameterized curve, e.g., a parameterized B-spline.

Finally, all integrals required by the algorithms are approximated by the Composite Trapezoidal Rule.

13.5.2 Diffeomorphism Considerations

As discussed earlier and seen from (13.2.3) and (13.3.3), γ^* may not be a diffeomorphism due to a horizontal and/or vertical region and is therefore in the closure of Γ^c. In order to guarantee the symmetry of the distance function

$$d_{l_n^c}(q_1, O(q_2 \circ \gamma)\sqrt{\dot{\gamma}}) = d_{l_n^c}(O(q_1 \circ \gamma^{-1}, q_2)\sqrt{\dot{\gamma^{-1}}}),$$

we must have the symmetry of the cost function

$$\begin{aligned} H^c(O, \gamma) &= 2 - 2\int_{\mathbb{S}^1} \langle q_1(t), O(q_2 \circ \gamma(t))\sqrt{\dot{\gamma}(t)}\rangle_{\mathbb{L}^2} dt && (13.5.1) \\ &= 2 - 2\int_{\mathbb{S}^1} \langle O(q_1 \circ \gamma^{-1}(t))\sqrt{\dot{\gamma^{-1}}(t)}, q_2(t)\rangle_{\mathbb{L}^2} dt. && (13.5.2) \end{aligned}$$

If $(O, \gamma) \in SO(n) \times \Gamma^c$, then the symmetries are guaranteed by isometry of $SO(n)$ and Γ^c. However, if γ is not a diffeomorphism, then there are some problems. For γ containing a flat region, $\gamma(t) = a, \forall t \in [b, c]$, the cost function (13.5.1) is well defined. However, (13.5.2) is not due to the non-existent of $\gamma^{-1}(a)$. One way to guarantee symmetry is to define $\gamma^{-1}(a)$ to be b or c. In fact, for the purpose of computing the value of the cost function, $\gamma^{-1}(a)$ can be defined as any finite number since the jump discontinuity of γ^{-1} at a does not change the integral. For γ containing a vertical region, $\gamma(a) = [b, c]$, it is not a function. Similar to previous idea, we can redefine $\gamma(a)$ to be any finite point and γ^{-1} to satisfy $\gamma^{-1}(t) = a, \forall t \in [b, c]$ and symmetry is satisfied.

Theoretically, therefore, when γ is not a diffeomorphism evaluation and symmetry of the cost function can be handled. In practice, however, numerically evaluating the cost function $H^c(O, \gamma)$ requires a quadrature rule that depends on every point in the discrete set. If γ has a vertical or near vertical segment then $\dot{\gamma}$ is infinite or very large and numerical overflow may occur. γ containing a flat region does not cause numerical problems when evaluating the cost function. In some versions, e.g., Algorithm 10 an interpolatory spline is used to represent γ. If a spline of degree 1, i.e., piecewise linear, is used there is no numerical problem. However, a higher degree spline requires care must be taken to guarantee that it is nondecreasing. This is not an issue during the iteration of the new Riemannian algorithm.

These theoretical and practical issues can be avoided for both the Coordinate Relaxation DP-based Algorithm 10 and the new Riemannian algorithm. In Section 13.3.1, the DP algorithm constrains the set of slope choices $\mathcal{N}_{i,j}$ to remain sufficiently far from 0 or ∞ and thereby avoids horizontal and vertical regions in γ^*. In the Riemannian algorithm, the third component l is defined by $l^2 = \sqrt{\dot{\gamma}}$. To avoid horizontal and vertical regions in γ, a penalty term is added and the cost function becomes

$$\int_0^{2\pi} \|Oq_1(t) - q_2(\int_0^t l^4(s)ds + m \bmod 2\pi) l^2(t)\|_2^2 dt + \omega((\int_0^{2\pi} (l^2(t) - 1)^2 dt + \int_0^{2\pi} (l^{-2}(t) - 1)^2 dt)), \tag{13.5.3}$$

where ω is a constant that makes the extra term relative small. When some region of γ is close to horizontal or vertical, the extra term increases and γ^* does not have such a region. There is no explicit lower or upper bound for the slopes of γ unlike the approach above for the DP-based algorithm.

13.5.3 Escaping Local Minima

The Riemannian optimization methods convergence results relate to a local minimum. There are many approaches to escape from local minima when working in Euclidean space. Two standard ones are the MCMC simulated annealing algorithm and the use of multiple runs with different initial conditions. For Riemannian optimizations, we can use similar ideas.

We have tested a Riemannian gradient-based MCMC simulated annealing algorithm using a Metropolis-Hastings acceptance test. For sufficiently small "temperature", the algorithm changes to one of Riemannian quasi-Newton algorithms. The basic idea of this algorithm is to search the domain sufficiently and find a satisfactory minimum. Unfortunately, the dimension of domain $SO(n) \times \mathbb{R} \times \mathcal{L}$ is infinite and the dimension of the finite approximation used is large enough so that a sufficiently thorough search was often found to be unacceptably expensive.

A simpler and, in practice effective, choice in this setting is to run the Riemannian quasi-Newton algorithms with multiple initial conditions. Let $(O_0, m_0, l_0) \in SO(n) \times \mathbb{R} \times \mathcal{L}$ denote the initial iterate. The initial rotation O_0 is given by the method used in Algorithm 8. The initial l_0 is given by choosing a small number of points on the curve, N_s, and running DP with small h, where h is defined in (13.3.2). The motivation is to make use of the global minimization property of DP on a coarse grid and then improve the quality of the solution by Riemannian quasi-Newton algorithms.

The value m_0 can be chosen uniformly or randomly on $[0, 2\pi]$. However, we automatically choose a set of m_0's as well as N_s for Riemannian quasi-Newton algorithms by exploring the structure of the curves. For example, let the curves in Figure 13.1 be two parts of two closed curves. If the rest of the curves are ignored, there are two minima that correspond to the peak of curve 1 matching the first peak or second peak of curve 2. The two minima can be obtained by using only two m_0's. Suppose the starting point of curve 1 is the point marked with a cross on the graph. The starting point on curve 2 can be any point in the green parts of curve 2. For these two initial conditions, Riemannian algorithms are able to search for the best matching point. Using this idea, if the total change of the angle for some interval along the curve is greater than a specified threshold T_m, an m_0 is added at the end of the interval. In order to avoid noise, it is required that the difference between consecutive m_0 points is greater than or equal to some positive value z. Each of the m_0's produced generates a distinct initial condition for the Riemannian optimization.

In practice, when one curve changes direction frequently and the other curve is relatively simple in shape, choosing which curve is used as the basis for the generation of the set of of m_0's depends on the context of the distance computation. Two curves with significantly different shape are expected to have a large distance. If the application requires an accurate approximation of the large distance then the curve with the more complicated shape should be used to generate the m_0's. If, however, large distances need not be approximated accurately, e.g., when distances are used to determine that the shapes are not in the same class, then the simpler curve should be used to generate the m_0's and the computational complexity of the optimization is reduced. This is quite different from DP which often requires a large number of break points to get a satisfactory result in either case above.

N_s is taken as $\min(2\pi/(m_0^{(i+1)} - m_0^{(i)}))$ where $m_0^{(i)}$ is the i-th initial condition of m. It is used for all runs of the Riemannian optimization algorithm. In practice, N_s can be chosen by another set of m_0's that result from the procedure above applied with a different threshold $T_{\hat{m}}$ rather than the threshold T_m used to generate the initial conditions for the parameter m.

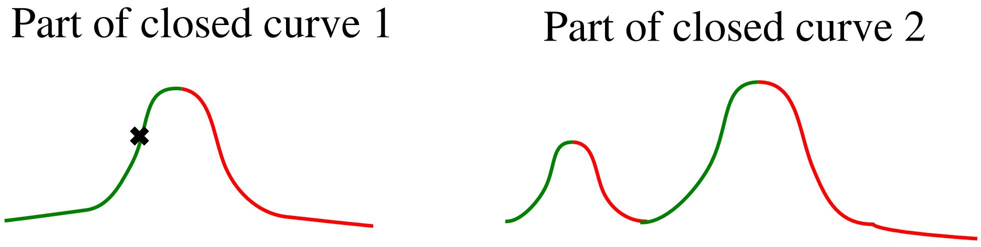


Figure 13.1: Choosing initial m for Riemannian quasi-Newton algorithms

13.6 Experiments

13.6.1 Overview of Experiments

The performance of the Riemannian optimization approach and Coordinate Relaxation methods to computing the elastic distance metric for curves in $\mathbb{R}^2$ is assessed in three stages. First, the problems Algorithms 9 and 10 have with multiple coordinate relaxation iterations, as discussed in Section 13.3.2 are illustrated. Second, the performances of the Riemannian optimization algorithms, RBFGS, LRBFGS, RTR-SR1, LRTR-SR1 and RSD, are compared to identify the preferred Riemannian method. Third, the preferred Riemannian method is compared systematically with the

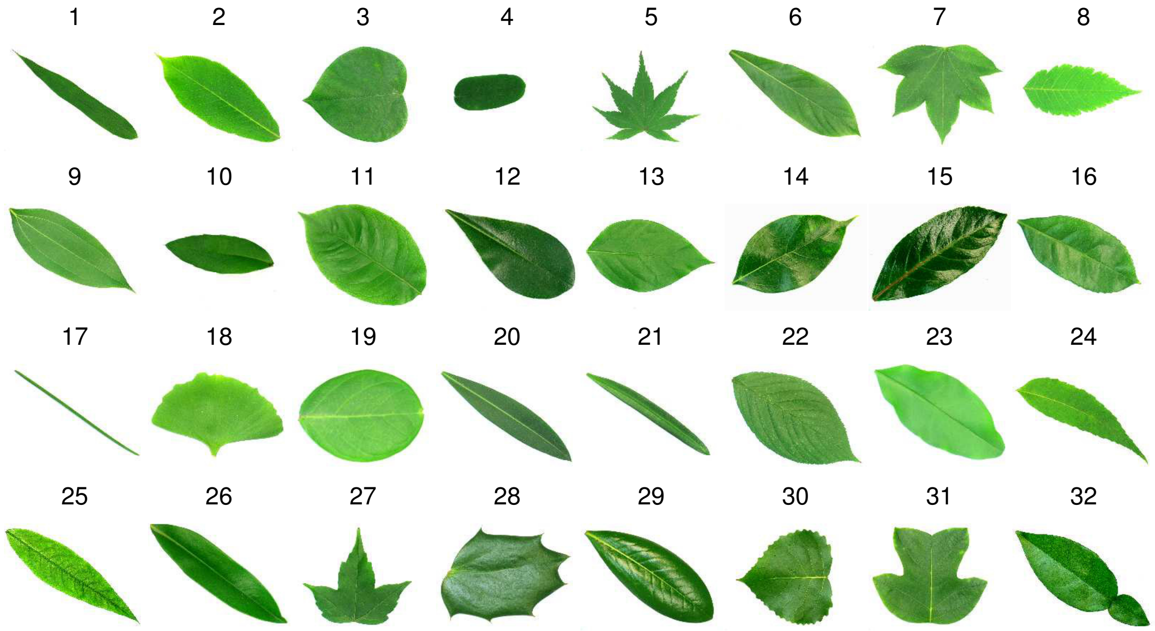


Figure 13.2: Samples of leaves from Flavia leaf dataset. One sample per species is illustrated.

DP-based algorithm used by Srivastava et al. in their experiments [SKJJ11], i.e. the Coordinate Relaxation method using only one iteration that we denote CR1.

Two public datasets are used in the experiments: the Flavia leaf dataset [WBX+07] and the MPEG-7 dataset [Uni]. The Flavia leaf dataset contains images of 1907 leaves from 32 species. Figure 13.2 shows an example leaf from each species. MPEG-7 contains 1400 images in 70 clusters each of which contains 20 shapes. Figure 13.3 shows an example shape from each cluster. The boundary curves of the shapes are extracted using the BWBOUNDARIES function in Matlab and piecewise linear interpolation is used to resample the curve at 100 points in $\mathbb{R}^2$, i.e., a 2×100 matrix.

13.6.2 Examples of Coordinate Relaxation Difficulties

For the experiments using Coordinate Relaxation based on DP, the mesh size, h defined in (13.3.2), is 6 and every 4-th point is chosen as a break point. The shapes from MPEG-7 dataset shown in Figure 13.4 are used to illustrate the problems of Algorithm 9 and Algorithm 10.

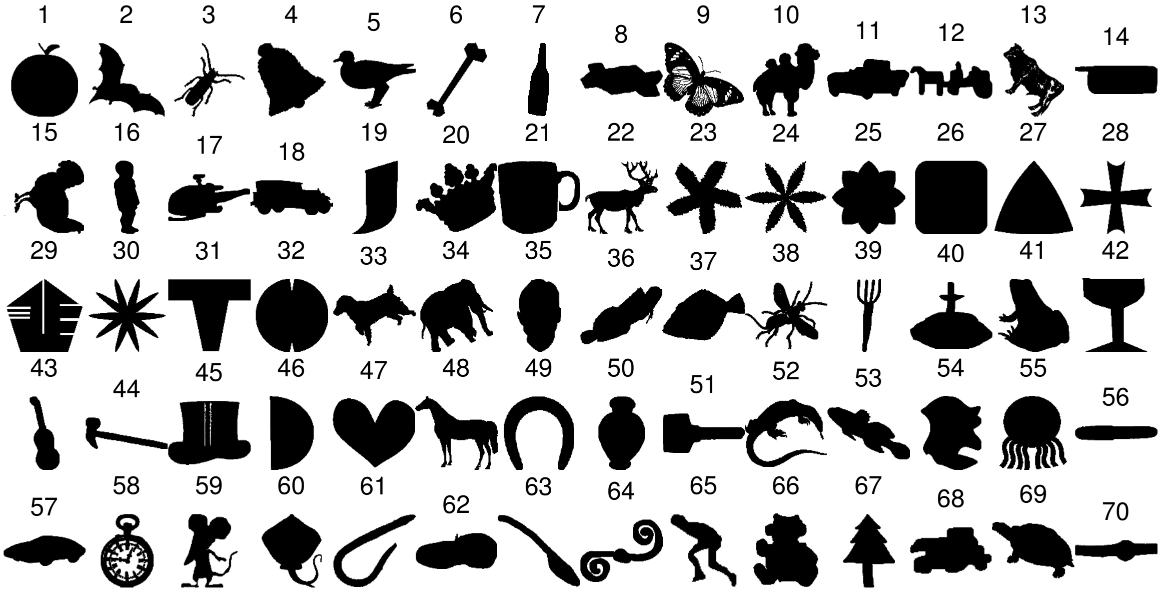


Figure 13.3: Samples of curves from MPEG-7 dataset. One sample per cluster is illustrated.

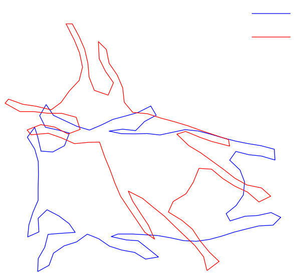


Figure 13.4: Two shapes from the MPEG-7 dataset.

The variation in the shape of curve β_2 in Algorithm 9 was identified as a serious problem. Figure 13.5 shows the shape of β_2 initially and in the first 4 iterations of the algorithm along with value of the cost function. The change to the shape is clear with many of its details disappearing gradually. Most significantly, the cost function is increasing and the algorithm is not reliable for optimization.

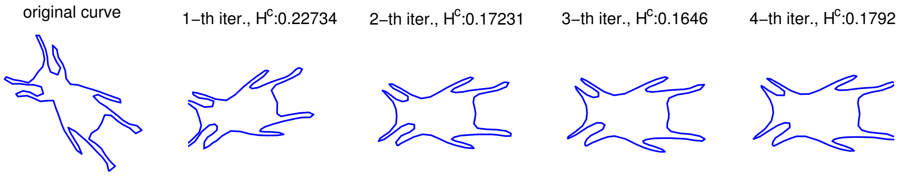


Figure 13.5: The variance of curve β_2 during the iteration in Algorithm 9.

The potential conflict in Algorithm 10 between the value of the cost function,H^c, evaluated during iteration k for $\bar{\beta}_2^{(\min,k)}$ computed using (13.3.4) and the value of H^c for the theoretically identical curve $\beta^{(k+1)}$ computed using (13.3.5) is also observed for the illustrative pair of shapes. Table 13.1 shows the variations of cost function. The value of H^c in the second row for iteration k should be the same as the value in the first row for iteration $k+1$. Clearly, the values are significantly different. Note also the values in the second row, which are the ones used by the algorithm in optimization decisions, are not decreasing. They, in fact, increase in subsequent iterations and the algorithm is unreliable.

Table 13.1: The variations of the cost function values in Algorithm 10

iteration (k)	1	2	3	4
H^c for $\beta^{(k)}$ of (13.3.5)	1.684911	0.267587	0.338316	0.285586
H^c for $\bar{\beta}_2^{(\min,k)}$ of (13.3.4)	0.227343	0.202650	0.196659	0.205583

The two difficulties illustrated by this pair of shapes were observed at some point in the iterations for every pair tested. This does not necessarily result in a significant error in the distance, but it does result in the inability to trust that additional iterations improve distance accuracy.

The difficulties can be avoided by only performing a single iteration of Coordinate Relaxation. This was done by Srivastava et al. [SKJJ11] and as a result they did not observe the problems. However, the accuracy of the distance computed using a single iteration is thereby limited by the quality of the choice of the initial reparameterization and rotation. Of the two, the initial reparameterization is the main difficulty since the optimal rotation for any particular reparameterization is given by the SVD.

Using a standard initial reparameterization such as the identity map does not reliably give an accurate distance. Figure 13.6 shows the optimization results for a single iteration of Coordinate Relaxation (CR1), using the identity map and the SVD, and for the Riemannian algorithm LRBFGS iterating until the cost function value is invariant to three digits. The final cost function is a factor of 2 smaller for LRBFGS and the superior quality of the final rotation and reparameterization from LRBFGS is clearly illustrated.

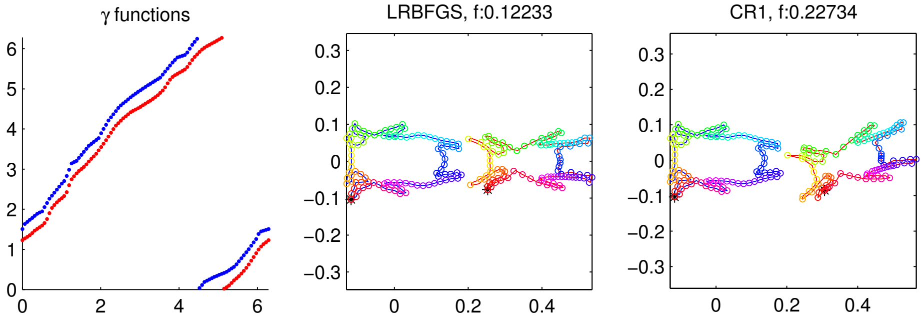


Figure 13.6: Results for LRBFGS and CR1.

13.6.3 The Preferred Riemannian Quasi-Newton Algorithm

The two public datasets were also used to compare the performances of several Riemannian optimization algorithms in minimizing the cost function (13.5.3). For these experiments, the stopping criterion for the Riemannian algorithms requires the relative change of the cost function in two successive iterates to be less than 10^{-3}, and the minimum number of iterations is 10. The number of points used to get the third component of the initial condition for the Riemannian algorithms, N_s, is set as described in Section 13.5.3. The weight, ω, in the cost function (13.5.3) is 1/8 initially and decreases on each iteration by $\omega \leftarrow 0.8\omega$. For the Flavia dataset, the values $T_m = 3\pi/4$, $T_{\tilde{m}} = \pi$ and $z = 4$ are used when setting initial m_0 and N_s. Since the shapes in the MPEG-7 dataset are more complex, the values $T_{\tilde{m}} = \pi/2$ and $z = 2$ are used. For both datasets if the value recommended automatically for N_s is less than 40 it is reset to 40.

All codes are written in C, compiled with gcc and run on the Florida State University HPC system using Quad-Core 2356 2.3 GHz Opterons [Cen]. The output time is the average CPU time of 10 runs with identical parameters. (The times observed had very low variance.)

To find the preferred Riemmanian method, all of the methods were run on several sets of randomly chosen pairs of shapes from the two datasets. Table 13.2 reports, t_{ave}, the average time to compute the distance between two shapes and, L_{ave}, the average cost function value for one of these sets from the Flavia and MPEG-7 shapes. The trends in other sets were similar.

The RBFGS and RTR-SR1 methods produce the smallest final cost function values, but this comes at the cost of computational times that are approximately 2 to 3 times those of the other methods. This is easily explained by noting that the computational complexity per step for these two methods is $O(N^2)$ due to the use of a dense matrix vector product. Note this also implies $O(N^2)$ space complexity. Both are less complex than the CR1 method with computational complexity per step of $O(N^3)$ and $O(N^2)$ space complexity.

The RSD method has low computational times due to its relatively low $O(N)$ computational complexity per step, but it does not result in a competitive final cost function value due to the simplicity of the approach.

The limited memory methods of LRBFGS and LRTR-SR1, do not require an $N \times N$ dense matrix vector product or $N \times N$ dense matrix storage. The final cost function values they achieve are not as small as those from RBFGS and RTR-SR1 but they are close, e.g., within 5% for LRBFGS.

Table 13.2: Comparison of Riemannian Methods for representative sets from the Flavia and MPEG-7 datasets: average time per pair (t_{ave}) in seconds and average cost function per pair (L_{ave}).

		RBFGS	LRBFGS	RTR-SR1	LRTR-SR1	RSD
Flavia dataset	L_{ave}	0.1783	0.1841	0.1779	0.1941	0.2068
	t_{ave}	0.3012	0.1136	0.3387	0.1398	0.1211
MPEG-7 dataset	L_{ave}	0.3485	0.3656	0.3421	0.4030	0.4506
	t_{ave}	0.9259	0.3024	1.0483	0.3897	0.3331

The computational complexity per step for LRBFGS is $O(Nkm_s)$ where k is the number of iterations and m_s is the number of initial conditions. Given two curves and a fixed stopping criterion, the m_s and k do not vary that much as N increases. Therefore, practically, the complexity of LRBFGS is $O(N)$. The effect of the substantial difference in the complexities of LRBFGS and CR1 is illustrated in Figure 13.7 for a representative pair of leaf shapes from the 27th species of the Flavia dataset. Boundary curves were extracted with different number of points N to test the relationship between N and time costs for LRBFGS and CR1. The break points of CR1 are chosen to be every 4 points and the time cost is the average of 10 runs with identical parameters.

In summary, all of the Riemannian algorithms were competitive with CR1 in terms of complexity and LRBFGS with its acceptable optimization of the cost function and its low computational and storage complexity is chosen as the preferred Riemannian algorithm for use further comparisons to CR1 from the point of view of quality of shape distance computations.

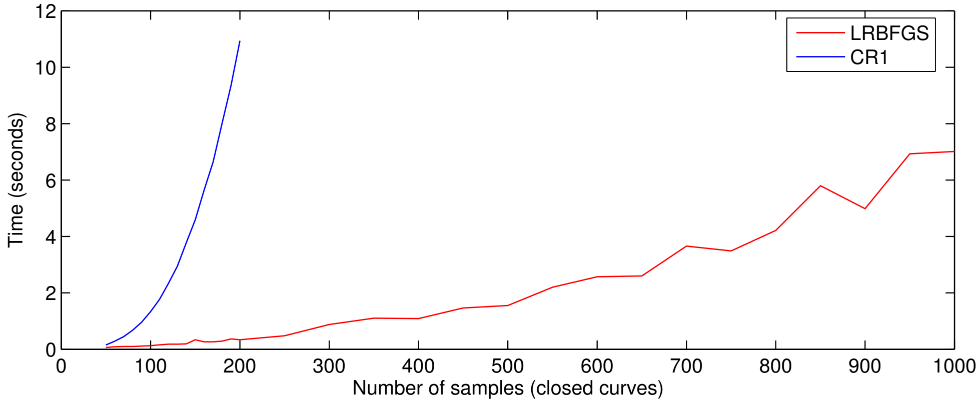


Figure 13.7: Comparison of complexities of CR1 and LRBFGS.

13.6.4 Performance Comparison for Flavia and MPEG-7 Datasets

In order to compare the cost and efficacy of the preferred Riemannian algorithm, LRBFGS, to those of the current state-of-the-art, CR1, all pairwise distances in the Flavia and MPEG-7 data sets were computed ($1,819,278$ and $980,700$ pairs respectively) using the testing environment described in Section 13.6.3. For CR1, the effect of the number of break points was considered by running each pair with a break point every 2, 4, 8 and 16 points given a fixed initial point, i.e., the sets are nested.

In addition to comparing the computation times and cost function values for the two algorithms, the quality of the distance computations was assessed using the one-nearest-neighbor (1NN) metric of cluster (species) preservation for the MPEG-7 (Flavia) shapes. The 1NN metric, μ, computes the percentage of points whose nearest neighbor are in the same cluster, i.e.,

$$\mu = \frac{1}{n}\sum_{i=1}^{n} C(i), \quad C(i) = \begin{cases} 1 & \text{if point } i \text{ and its nearest neighbor are in the same cluster} \\ 0 & \text{otherwise} \end{cases}.$$

A very significant improvement in the final value of the cost function achieved by LRBFGS compared to the value achieved by CR1 is observed. For the Flavia dataset, LRBFGS reduces the cost function more than CR1 in 93.65%, 94.84%, 96.28% and 97.85% of the pairs when choosing break points every 2, 4, 8 and 16 points for CR1 respectively. For the MPEG-7 dataset, LRBFGS minimizes the cost function better than CR1 in 88.39%, 90.66%, 93.35% and 96.02% of the pairs when choosing break points every 2, 4, 8 and 16 points for CR1 respectively.

The distribution of the ratio of the cost function value of CR1 to that of LRBFGS is shown in the histograms in Figure 13.8. Ratios where LRBFGS was more than 4 times better are not include for presentation purposes. The maximum ratios for the Flavia data set (and the number of ratios exceeding 4) were 13.3 (215), 17.4 (525), 30.9 (1393) and 40.9 (4312) when choosing break points every 2, 4, 8 and 16 points for CR1 respectively. The maximum ratios for the MPEG-7 data set (and the number of ratios exceeding 4) were 11750 (2), 11750 (7), 11750 (27) and 12329 (232) when choosing break points every 2, 4, 8 and 16 points for CR1 respectively. The amazingly large ratios beyond 4 occur for pairs of shapes that are fairly close in shape where LRBFGS achieves a very small cost function value. Not only is it clear from this data that, in general, LRBFGS reduces the cost function more than CR1, but also in the cases when CR1 produces a smaller cost function value it is usually very close to the value produced by LRBFGS.

Of course, if the improvement in the reduction of the cost function requires a very large increase in computation time then the argument in favor of LRBFGS and the other Riemannian methods weakens. The histograms of computation times for CR1 and LRBFGS for the MPEG-7 and Flavia datasets in Figure 13.9 show that LRBFGS has a significant advantage in computation time. The largest computation time for LRBFGS is smaller than all computation times of CR1 using $N/2$ and $N/4$ break points for both Flavia and MPEG-7 datasets and also when using $N/8$ break points for the Flavia dataset. When using $N/8$ break points for the MPEG-7 dataset only 0.03% of the pairs have CR1 computation times smaller than the largest LRBFGS computation time. When using $N/16$ break points 0.06% and 13% of the pairs have CR1 computation times smaller than the largest LRBFGS computation time for the Flavia and MPEG-7 datasets respectively. Therefore, the Riemannian approach to the cost function and its optimization using LRBFGS yields superior optimization in significantly less time than CR1 for the vast majority of the pairs computed.

A more careful examination of the the times indicates that the Riemannian approach has another advantage. The computation time for a pair of shapes using CR1 is essentially proportional to the number of break points used. There is very little variation between computation times when using the same number of break points as is seen in the CR1 spikes in Figure 13.9.

Figure 13.9 also shows that the, much smaller, computation times for LRBFGS have significant variation. Recall that the Riemannian methods automatically select the position and number of initial conditions used to compute the distance for a pair of shapes. The computation time per initial condition (PIC) for LRBFGS varies only slightly as is shown also in Figure 13.10, and the computation time for a pair of shapes is essentially proportional to the number of initial conditions used. Since the number of initial conditions is a simple measure of the complexity of one or both of the shapes in the pair, the Riemannian methods have the additional advantage of only requiring a computation time that reflects the difficulty of the problem.

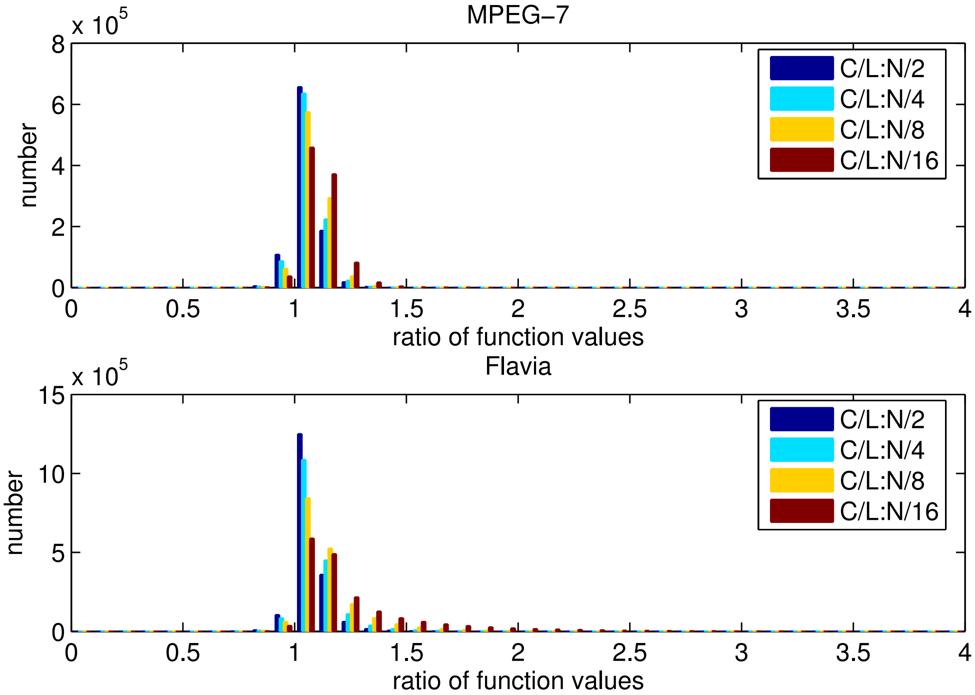


Figure 13.8: Histograms of ratios of the CR1 cost function value to the LRBFGS cost function value (C/L) for MPEG-7 and Flavia datasets. $N/i, i = 2, 4, 8, 16$ denote the number of break points in CR1. Bins are $(0, 0.1), \ldots, (0.9, 1.0), \ldots, (3.9, 4.0)$.

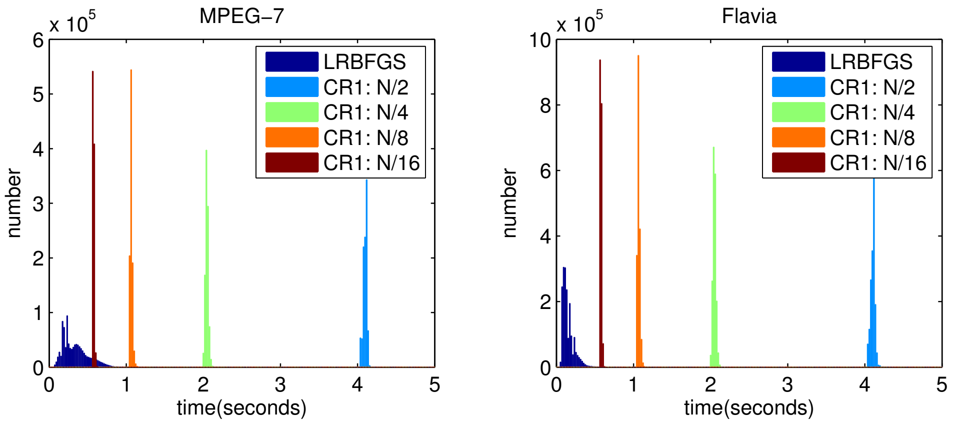


Figure 13.9: Histograms of computation times of LRBFGS and CR1 for MPEG-7 and Flavia datasets.

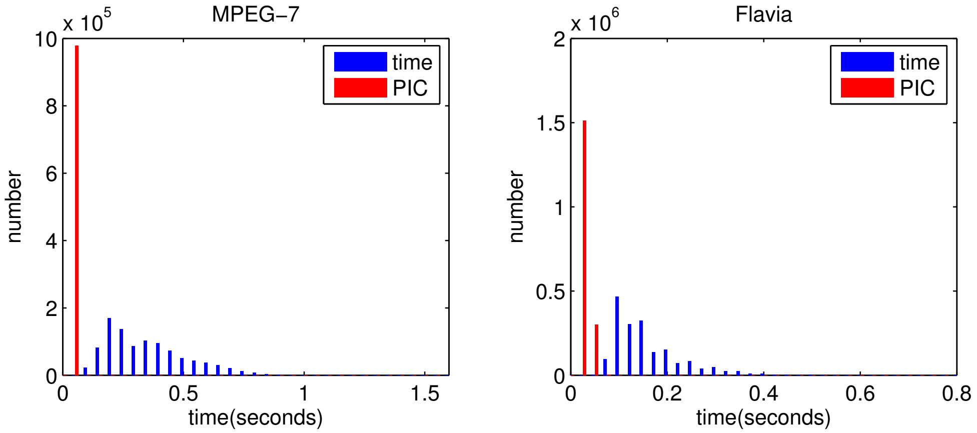


Figure 13.10: Histograms of computation times of LRBFGS and computation times per initial condition (PIC) of LRBFGS for MPEG-7 and Flavia datasets.

Table 13.3 shows the average time cost and 1NN for both datasets. The trends are as expected given the examples in Figures 13.2, 13.3. For the MPEG-7 dataset, the shapes in different clusters are very distinct compared to the significantly greater similarity of shapes in certain pairs of species in the Flavia dataset, e.g., species 1 and 21. Therefore, the μ values are expected to be higher for MPEG-7 distances since the distinctions are easier to make while lower μ values are expected for Flavia distances. For CR1 it is expected that μ values would increase as the number of break points increases. All of these trends are observed in the μ data.

The comparison of μ achieved by LRBFGS to those of CR1 shows a clear advantage to L-

RBFGS. LRBFGS achieves a value of μ higher than CR1 using the densest set of break points. Not surprisingly, given the distributions of computation times discussed earlier, the average time for LRBFGS is significantly smaller than the average time for even the sparsest set of CR1 break points ($N/16$).

Table 13.3: The average computation time and 1NN of LRBFGS and CR1 with break points chosen to be every 2, 4, 8 and 16 points.

		LRBFGS	CR1			
			$N/16$	$N/8$	$N/4$	$N/2$
Flavia leaf dataset	average time(seconds)	0.1596	0.5754	1.0683	2.0541	4.1077
	1NN of 32 species	87.41%	80.02%	82.01%	84.58%	87.31%
MPEG-7 dataset	average time(seconds)	0.3509	0.5744	1.0665	2.0507	4.1010
	1NN of 70 clusters	97.21%	95.21%	97.00%	97.21%	97.00%

13.7 Conclusion

We have explored the computation of the elastic distance metric for open and closed curves in $\mathbb{R}^2$ and reviewed CR1, the DP-based algorithm of Srivastava et al., [SKJJ11], the approximations upon which it is based, its computational complexity, its difficulties, and its performance in terms of time and cost function reduction. As an alternative to CR1, we have derived a Riemannian approach to computing the elastic distance metric and developed an efficient implementation using various Riemannian optimization algorithms.

Empirical comparisons of the Riemannian approach using LRBFGS and CR1 and shapes from the MPEG-7 and Flavia datasets were performed. The results demonstrate that the Riemannian approach produces more useful distance estimates, as measured by the 1NN metric for clustering, in significantly less time and that the computational time required adapts to the complexity of the shapes being compared.

The efficiency and efficacy of the Riemannian approach to computing the elastic distance metric promises to improve considerably shape analysis computations that are based upon distance computations, e.g., the Karcher mean of a set of shapes, geodesic paths between shapes, and inferences on shapes. These improvements will be demonstrated in future work.

CHAPTER 14

SECANT-BASED NONLINEAR DIMENSION REDUCTION

14.1 Introduction

The problem of dimension reduction occurs in many forms. In its most straightforward form, sample data is given in a metric space of unknown dimension or a Euclidean space of high dimension and the problem is to determine a representation for the data in a sufficiently low dimensional Euclidean space while approximately preserving some specified property, e.g., pairwise distances, [LV07a]. The techniques are commonly used in manifold learning (see [ZZ04] and its references). It is used often also in model reduction of dynamical systems where the data represents one or more solution trajectory in the state space (see for example [Li00, Rew03, FST88, MT89, JKT90, BK05, TAJP08]).

The work in this chapter is based on the work of Broomhead and Kirby [BK05]. They consider dynamical systems that are defined on high-dimensional spaces, but which have low-dimensional attractors. The low-dimensional attractor may be approximated by a subspace or a submanifold $\mathcal{M}$. The idea of Broomhead and Kirby is based on Whitney's Theorem [Whi36] and uses a cost function that is optimized in the sense that the projection is easy to invert. Our work is to compare the performance of different optimization methods.

This chapter is organized as follows. The problem is stated in Section 14.2. Section 14.3 provides some details of the computation of the two cost functions considered in the experiments. Experimental results are presented in Section 14.4 and conclusions are discussed in Section 14.5.

14.2 Problem Statement

Suppose $\mathcal{M}$ is a manifold with dimension m embedded in $\mathbb{R}^n$. Let U denote a tall thin $n \times p$ matrix with full rank, $\mathrm{col}(U)$ denote the column space of U and π_U denote the orthogonal projection onto $\mathrm{col}(U)$,

$$\pi_U = U(U^T U)^{-1} U^T.$$

Since π_U is a projection onto a space, it is related to the Grassmann manifold. Based on the implementation of Section 10.6, we consider that an orthonormal matrix U such that $[U]$ represents an element of the Grassmann manifold. Noting that the projection is invariant to the choice of representations in $[U]$, we define the projection

$$\pi_{[U]} = UU^T.$$

The Whitney Embedding Theorem [Whi36] says that when $p \geq 2m + 1$, there is a large (open dense) set of projections $\pi_{[U]}$ such that $\pi_{[U]}|_{\mathcal{M}}$, the projection restricted to $\mathcal{M}$, is invertible. Specifically, let $\mathrm{col}(U)^{\perp}$ denote the perpendicular space of $\mathrm{col}(U)$. There is a function $g : \mathrm{dom}(g) \subseteq \mathrm{col}(U) \to \mathrm{col}(U)^{\perp}$ such that

$$\mathcal{M} = \{(x, g(x))|x \in \mathrm{dom}(g)\},$$

and $\pi_{[U]}|_{\mathcal{M}} : \mathcal{M} = (x, g(x)) \mapsto x$, called an embedding projection of $\mathcal{M}$ into the p-dimensional linear space $\mathrm{col}(U)$, is invertible. Note a invertible linear function is smooth. Therefore, $\pi_{[U]}|_{\mathcal{M}}$ is a diffeomorphism.

Remark 14.2.1. *$p \geq 2m+1$ guarantees that there is an open dense set of embedding projections for $\mathcal{M}$, but for a given $\mathcal{M}$, it is quite possible that there is an open dense set of embedding projections for smaller p. For example, let $\mathcal{M}$ be the unit circle in* $\mathrm{span}\{e_1, e_2\} \subset \mathbb{R}^n$. *We have $m = 1$ (the circle is a one-dimensional manifold) and $2m+1 = 3$, but it is clear that there is an open dense set of $n \times 2$ matrices U such that π_U is an embedding of $\mathcal{M}$. It is also clear that the "best"* $\mathrm{col}(U)$ *is* $\mathrm{span}\{e_1, e_2\}$, *in which case* $g : \mathrm{span}\{e_1, e_2\} \to \mathrm{span}\{e_3\}$ *is the zero function with* $\mathrm{dom}(g)$ *the unit circle in* $\mathrm{span}\{e_1, e_2\}$.

The Whitney Embedding Theorem provides conditions for the existence of the projections. Broomhead and Kirby [BK05] further search for projections that are easy to invert. The function $\pi_{[U]}^{-1} : \mathrm{dom}(g) \to \mathcal{M}$ is Lipschitz due to diffeomorphism, i.e., there exists a constant $k_{\pi_{[U]}}$ such that

$$k_{\pi_{[U]}} \|\tilde{x} - \tilde{y}\|_2 \leq \|\pi_{[U]}\tilde{x} - \pi_{[U]}\tilde{y}\|_2, \tag{14.2.1}$$

for all $\tilde{x}, \tilde{y} \in \mathcal{M}$. $k_{\pi_{[U]}}$ is taken as a measure of the conditioning of the inverse of the projection. The larger $k_{\pi_{[U]}}$ is, the more slowly varying the inverse is, and the easier it is to fit the function g. Based on (14.2.1), $k_{\pi_{[U]}}$ satisfies

$$k_{\pi_{[U]}} \leq \left\| \frac{\pi_{[U]}(\tilde{x} - \tilde{y})}{\|\tilde{x} - \tilde{y}\|_2} \right\|_2 .$$

Therefore, $k_{\pi_{[U]}}$ can be defined as

$$k_{\pi_{[U]}} = \inf_{\tilde{x},\tilde{y}\in\mathcal{M},\tilde{x}\neq\tilde{y}} \left\| \frac{\pi_{[U]}(\tilde{x}-\tilde{y})}{\|\tilde{x}-\tilde{y}\|_2} \right\|_2 . \tag{14.2.2}$$

Let Σ denote the set of unit secants of the manifold $\mathcal{M}$,

$$\Sigma = \left\{ \frac{\tilde{x}-\tilde{y}}{\|\tilde{x}-\tilde{y}\|_2} : \forall \tilde{x}, \tilde{y} \in \mathcal{M}, \tilde{x} \neq \tilde{y} \right\},$$

and $\bar{\Sigma}$ denote the closure of Σ. Using $\phi([U])$ to denote the square of (14.2.2), gives the cost function

$$\phi([U]) = \min_{k\in\bar{\Sigma}} \|\pi_{[U]}k\|_2^2. \tag{14.2.3}$$

$\phi([U])$ is a partly smooth function defined on the Grassmann manifold. Broomhead and Kirby [BK05] did not work on this cost function directly. Instead, they proposed a smooth cost function

$$F([U]) = \frac{1}{|\Sigma|} \sum_{k\in\Sigma} \frac{1}{\|\pi_{[U]}k\|_2}, \tag{14.2.4}$$

where $|\Sigma|$ denotes the number of elements in Σ. In (14.2.4), small projections of unit secant norms are heavily penalized while in (14.2.3) the smallest projection is forced to be large. Maximizing $\phi([U])$ and minimizing $F([U])$ have similar goals, however, whether and when they have same optima are still open questions.

The horizontal lift of the gradient of $\phi([U])$ at U is

$$(\operatorname{grad} \phi([U]))_{\uparrow_U} = (I - UU^T)(2k_* k_*^T U),$$

where $k_* = \arg\min_{k\in\bar{\Sigma}}(\|U^T k\|)$. If $\min_{k\in\bar{\Sigma}}(\|U^T k\|)$ has more than one solution for a given U, then $\phi([U])$ is not differentiable at $[U]$. The horizontal lift of the gradient of $F([U])$ at U is

$$(\operatorname{grad} F([U]))_{\uparrow_U} = -\frac{1}{|\Sigma|} \sum_{k\in\Sigma} \frac{kk^T U}{\|UU^T k\|_2^3}$$

and the action of Hessian of $F([U])$ on a vector ξ in the horizontal space $\mathcal{H}_U$ is

$$(\operatorname{Hess} F([U]))_{\uparrow_U}[\xi_{\uparrow_U}] = -\frac{I - UU^T}{|\Sigma|} \sum_{k\in\Sigma} \left(\frac{kk^T\xi - \xi U^T kk^T U}{\|U^T k\|_2^3} - \frac{3\,\mathrm{trace}(k^T U \xi^T k) kk^T U}{\|U^T k\|_2^5} \right).$$

14.3 Properties of the Cost Functions and Discretization

Given a manifold $\mathcal{M}$ and U, the projection $\pi_{[U]}|_{\mathcal{M}} : \mathcal{M} \to \text{dom}(g)$ may not be an invertible function. If it is not, then there exist $\tilde{x}$, $\tilde{y}$ and $\tilde{x} \neq \tilde{y}$ such that $\pi_{[U]}\tilde{x} = \pi_{[U]}\tilde{y}$. Therefore, $\|\pi_{[U]}\tilde{k}\|_2 = 0$, where $\tilde{k} = (\tilde{x} - \tilde{y})/\|\tilde{x} - \tilde{y}\|_2$. By the definitions of $\phi([U])$ and $F([U])$, we have that

$$\phi([U]) = 0 \text{ and } F([U]) = \infty,$$

where ∞ represents a positive number divided by 0. Let $\mathcal{Z}$ denote the set of $[U]$ that defines a non-invertible projection $\pi_{[U]}|_{\mathcal{M}}$, i.e.,

$$\mathcal{Z} = \{[U] \in \text{Gr}(p, n) | \pi_{[U]}|_{\mathcal{M}} : \mathcal{M} \to \text{dom}(g) \text{ is not invertible.}\}$$

If the initial $[U_0]$ is in the set $\mathcal{Z}$, then $F([U])$ cannot be used as a cost function. Furthermore, $\mathcal{Z}$ may contain a dense open set $\mathcal{N}$. When the initial $[U_0]$ is in this open set $\mathcal{N}$, the gradient $\text{grad}\, \phi([U_0])$ is a zero tangent vector. All of the gradient methods cannot escape from this point $[U_0]$. Therefore, $\phi([U])$ cannot be used as a cost function. Overall, a bad choice of initial $[U_0]$ may cause serious problems for both cost functions.

Since the number of the points on manifold $\mathcal{M}$ is infinite, the number of elements in Σ is also infinite and it is impossible to compute $\phi([U])$ and $F([U])$ exactly in practice. A practical method must discretize the manifold and choose a finite set of points making the number of elements in Σ finite. Let $\tilde{\mathcal{M}}$ denote the discretized manifold. Let $\tilde{\Sigma}$ denote the finite set and let $\tilde{\phi}([U])$, $\tilde{F}([U])$ denote the cost functions for $\tilde{\Sigma}$.

There are some consequences of discretization. First, accuracy depends upon the discretization. Second and more important, $\mathcal{Z}$ is replaced with the finite set

$$\tilde{\mathcal{Z}} = \{[U] \in \text{Gr}(p, n) | \pi_{[U]} : \tilde{\mathcal{M}} \to \text{dom}(g) \text{ is not invertible.}\}$$

Since the number of elements in $\tilde{Z}$ is finite, it is unlikely for $[U_0]$ to be in $\tilde{\mathcal{Z}}$. $\tilde{\phi}([U])$ and $\tilde{F}([U])$ are therefore well-defined for most of initial conditions. The new cost functions $\tilde{\phi}([U])$ and $\tilde{F}([U])$ may have many local optima over $\mathcal{N}$ rather than the flat profiles of $\phi([U])$ and $F([U])$ at the values of 0 and ∞ respectively. Escaping from the local optima becomes an algorithmic problem. In the implementation used for the experiments in the next section, the manifold is discretized and multiple initial conditions are used.

14.4 Experiments

In [BK05], Broomhead and Kirby use RCG(Algorithm 11) based on the Polak-Ribiére formula with parallel translation and exponential mapping to optimize the smooth cost function $\tilde{F}([U])$. Details of an RCG implementation that do not depend on a retraction and a vector transport are discussed in Chapter 9. For the cost function $\tilde{F}([U])$, the behavior of RCG is compared to the behaviors of RBFGS, LRBFGS, RTR-SR1, RTR-Newton and LRTR-SR1. For the partly smooth cost function $\tilde{\phi}([U])$ the behaviors RBFGS and RGS are presented. Aspects of the implementations specific to the two cost functions are given below.

Algorithm 11 Conjugate gradient for minimizing $F(Y)$ on the Grassmann manifold

```
Given $Y_0$ such that $Y_0^T Y_0 = I$, compute $G_0 = \operatorname{grad} F(Y_0)$ and set $H_0 = -G_0$;
**for** $k = 0, 1, \dots$ **do**
    Minimize $F(Y_k(t))$ over $t$ and find $t_{\min}$ where
        $Y(t) = YV \cos(\Sigma t) V^T + U \sin(\Sigma t) V^T$
    and $U \Sigma V^T$ is the compact singular value decomposition of $H_k$.
    Set $t_k = t_{\min}$ and $Y_{k+1} = Y_k(t_k)$.
    Compute $G_{k+1} = \operatorname{grad} F(Y_{k+1})$.
    Apply parallel translation for tangent vectors $H_k$ and $G_k$ and transport them to the horizontal space of $Y_{k+1}$:
        $P_{\gamma_k}^{1 \leftarrow 0} H_k = (-Y_k V \sin \Sigma t_k + U \cos \Sigma t_k) \Sigma V^T,$
        $P_{\gamma_k}^{1 \leftarrow 0} G_k = G_k - (Y_k V \sin \Sigma t_k + U(I - \cos \Sigma t_k)) U^T G_k,$
    where $\gamma_k$ is a geodesic from $Y_k$ to $Y_{k+1}$ such that $\gamma_k(0) = Y_k$ and $\gamma_k(1) = Y_{k+1}$.
    Compute the new search direction
        $H_{k+1} = -G_{k+1} + \sigma_k P_{\gamma_k}^{1 \leftarrow 0} H_k, \text{ where } \sigma_k = \frac{\langle G_{k+1} - P_{\gamma_k}^{1 \leftarrow 0} G_k, G_{k+1} \rangle}{\langle G_k, G_k \rangle}$
    and $\langle \Delta_1, \Delta_2 \rangle = \operatorname{trace} \Delta_1^T \Delta_2$.
    Reset $H_{k+1} = -G_{k+1}$ if $k + 1 \mod p(n-p) = 0$.
**end for**
```

For the Riemannian methods other than RCG, the retraction is (10.6.11) and the vector transport is (9.5.2) or (4.4.9) depending on whether the locking condition is imposed or not. The algorithmic parameter settings of RBFGS, LRBFGS, RTR-SR1 and LRTR-SR1 are the same as

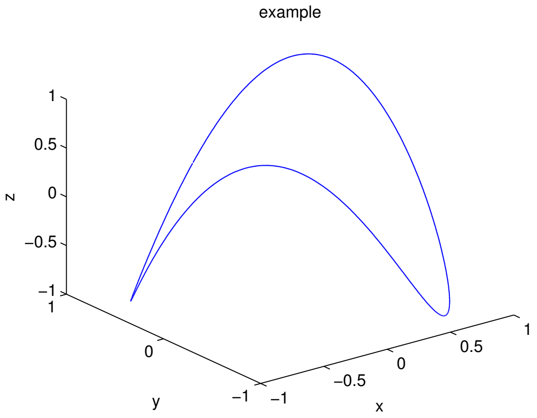


Figure 14.1: The example $P = G(\mathbb{S}^1)$.

those in Section 11.3. The exact line search algorithm of RCG is based on the function FMINUNC in Matlab. The comparison is performed in Matlab 7.0.0 on a 32 bit Windows platform with a 2.4 GHz CPU (T8300).

We consider the example that first appeared in [BK00] and is also used in [BK05]. The manifold $\mathcal{M}$ is the range of a map of circle

$$G : \mathbb{S}^1 \to \mathbb{R}^3, \theta \mapsto (\sin\theta, \cos\theta, \sin 2\theta),$$

and as shown in Figure 14.1. The experiments consider projecting this manifold from $\mathbb{R}^3$ to $\mathbb{R}^2$. Thus, n is 3 and p is 2. $\mathcal{W}$ is defined by 50 points uniformly separated in $[0, 2\pi]$ and the manifold is discretized as $P = G(\mathbb{S}^1)$ by $\tilde{\mathcal{M}} = G(\mathcal{W})$. The initial iterate is chosen by orthonormalizing the columns of a matrix with elements drawn from a normal random distribution. Since the dimension of the Grassmann manifold $\mathrm{Gr}(2, 3)$ is only 2 and the parameter m in LRBFGS an LRTR-SR1 are usually chosen to be less than the dimension of the manifold, we choose the m to be 1.

Given a $[U] \in \mathrm{Gr}(2, 3)$, the functions $\hat{\phi}_{[U]}(\eta) = \tilde{\phi}(\mathrm{Exp}_{[U]}(\eta))$ and $\hat{F}_{[U]}(\eta) = \tilde{F}(\mathrm{Exp}_{[U]}(\eta))$ are defined on the tangent space $\mathrm{T}_{[U]}\,\mathrm{Gr}(2, 3)$ which is a 2-dimensional flat space. Figure 14.2 shows the graphs of $\hat{\phi}_{[U_*]}(\eta)$ and $\hat{F}_{[U_*]}(\eta)$ where $[U_*]$ is the desired optimum (see Figure 14.3 and 14.5). The tangent space of two figures in the left column is $[-\pi, \pi] \times [-\pi, \pi]$ and tangent space of the two figures in the right column is $[-\pi/2, \pi/2] \times [-\pi/2, \pi/2]$. For all $U \in \mathrm{Gr}(2, 3)$, $\eta \in \mathrm{T}_{[U]}\,\mathrm{Gr}(2, 3)$ and $\|\eta\|_2 = \pi/2$, we know $\mathrm{Exp}_{[U]}(2\eta) = U$ and thus $\mathrm{Exp}_{[U]}(\eta) = \mathrm{Exp}_{[U]}(-\eta)$. We only need to observe the region where $\|\eta\|_2 \leq \pi/2$.

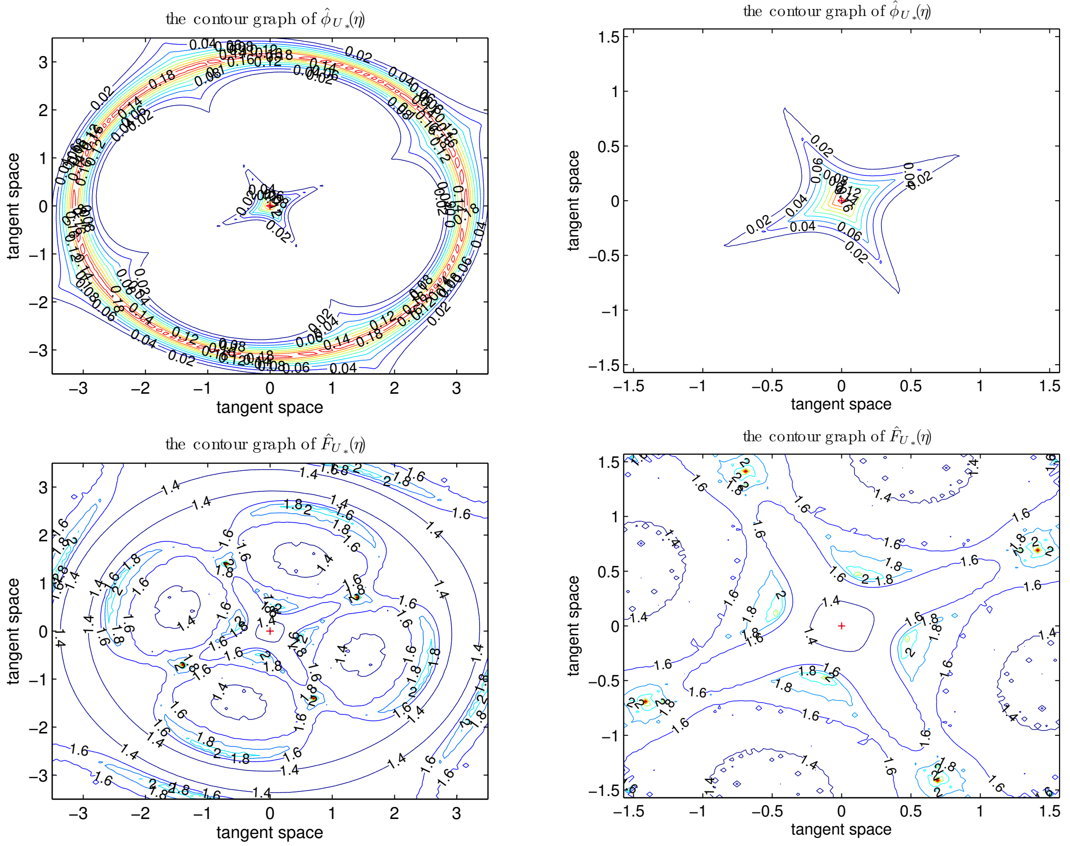
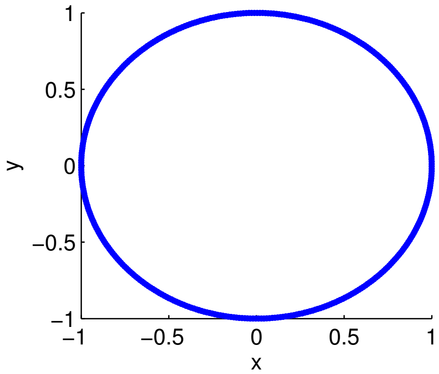


Figure 14.2: The top two figures show the contours of $\hat{\phi}_{[U_*]}(\eta)$ and the bottom two figures are the contours of $\hat{F}_{[U_*]}(\eta)$.



$$[U_*] = \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \right]$$

Figure 14.3: The left figure is the projection of $[U_*]$ onto 2-dimensional space $\pi_{[U_*]}P$ and the matrix on the right is the desired optimizer. The optimal projection $\pi_{[U_*]}$ projects P onto the X-Y plane [BK00].

From the figure of $\hat{\phi}_{[U_*]}(\eta)$ in Figure 14.2, we can see the origin is a maximizer which verifies the way we choose $[U_*]$. What is more, within the region $\|\eta\|_2 \leq \pi/2$, the origin is also the only significant maximizer, i.e., other local maxima with much smaller cost function values are not visible. Around the maximizer, there is a large flat region that corresponds to the set $\mathcal{Z}$ on which the function $\pi_{[U]}$ is non-invertible. Let $\mathrm{Exp}_{[U_*]}^{-1} \mathcal{Z}$ denote this flat region. Ideally, $\hat{\phi}_{[U_*]}(\eta)$ should be 0 in $\mathrm{Exp}_{[U_*]}^{-1} \mathcal{Z}$. Due to the discretization, it is not 0 and there are many local optima in $\mathrm{Exp}_{[U_*]}^{-1} \mathcal{Z}$ (see discussion in Section 14.3). From the figure of $\hat{F}_{[U_*]}(\eta)$, the origin is a minimizer. Based on the discussion in Section 14.3, $\hat{F}_{[U_*]}(\eta)$ should be ∞ in $\mathrm{Exp}_{[U_*]}^{-1} \mathcal{Z}$ or at least in practice much larger than the value of the cost function at the origin. However, the figure clearly shows that this is not the case for the discretized problem and this cost function. Instead of using 50 points to represent the manifold, 100, 200 and 400 points are also used and similar contours are observed.

To test the efficiency of the optimization algorithms, we use an initial point $[U_0] \notin \mathcal{Z}$ to make all algorithms tested converge to the desired optimizer $[U_*]$. The method of choosing the set of initial conditions to improve the likelihood of converging to $[U_*]$ or at least to an invertible local extrema is still an open question for both cost functions.

Tables 14.1 and 14.2 show the performance of the algorithms. We can see for the partly smooth function $\tilde{\phi}([U])$, RBFGS is the fastest one and for the smooth cost function $\tilde{F}([U])$, LRBFGS is the fastest.

Table 14.1: Comparison of RTR-Newton, Riemannian quasi-Newton algorithms and RCG for smooth cost function $\tilde{F}([U])$. $\tilde{F}([U_*]) = 1.355322$.

	RTR-Newton	RBFGS	LRBFGS	RTR-SR1	LRTR-SR1	RCG
$iter$	6	15	11	22	32	6
nf	6	23	19	22	32	37
ng	6	19	15	22	32	37
nH	12	28	0	47	0	0
nV	0	46	34	21	79	36
nR	5	22	18	21	31	31
gf_f	8.34_{-9}	1.44_{-7}	2.58_{-7}	6.38_{-11}	1.06_{-11}	6.02_{-9}
gf_f/gf_0	2.66_{-8}	4.60_{-7}	8.25_{-7}	2.04_{-10}	3.38_{-11}	1.92_{-8}
t	1.41	1.53	1.19	1.72	2.55	4.19

Table 14.2: Comparison of RBFGS and RGS for partly smooth cost function $\tilde{\phi}([U])$. $\tilde{\phi}([U_*]) = 2.006318\text{e-}001$

	RBFGS	RGS
$iter$	20	33
nf	54	330
ng	54	297
nH	38	0
nV	96	264
nR	53	625
gf_f	8.01_{-1}	8.01_{-1}
gf_f/gf_0	3.32	3.32
t	1.70	9.75

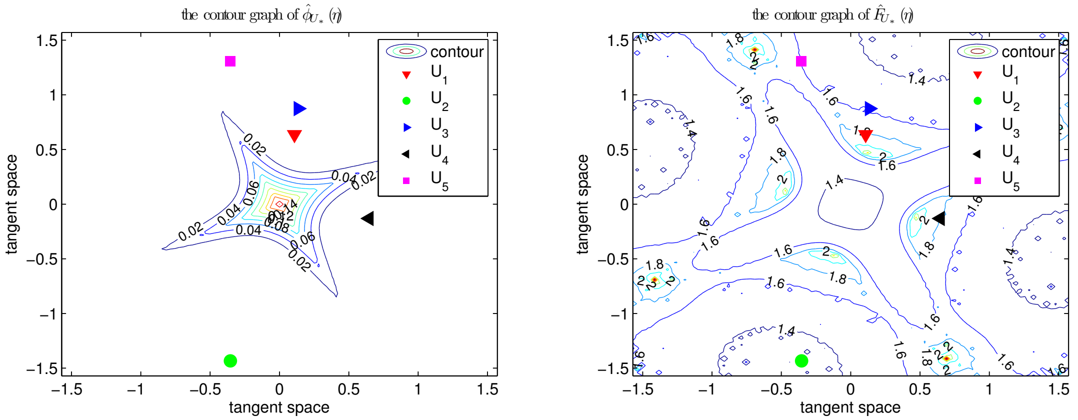


Figure 14.4: The initial points $U_i \notin \mathcal{Z}, i = 1, \ldots 5$ in the contour graphs of $\hat{\phi}_{[U_*]}(\eta)$ and $\hat{F}_{[U_*]}(\eta)$.

RCG, used by Broomhead and Kirby [BK05], is the slowest algorithm, despite its small number of iterations, due to the relatively large number of function evaluations and gradient evaluations required by its exact line search. RCG using the modified Polak-Ribiére formula (see [NW06, (5.45)]) and an inexact line search algorithm based on the Wolfe conditions was investigated as a potentially more efficient algorithm. While the cost per iteration decreased, the computational time increased significantly since the number of iterations increased to 52. RTR-Newton converges quadratically and requires fewer function and gradient evaluations. However, its computational time is larger than LRBFGS and comparable to that of RBFGS due to the expense of computing the action of the Hessian on a tangent vector.

To show the influence of the choice of initial condition to the combination of cost functions and optimization algorithms, 20 initial conditions $[U_i] \in \mathcal{Z}, i = 1, 2, \ldots, 20$ are used. The first 5, $[U_i] \in \mathcal{Z}, i = 1, 2, \ldots, 5$, are used as typical examples in Figures 14.4, 14.5 and Table 14.3. Figure 14.4 shown the 5 initial conditions $[U_i] \in \mathcal{Z}, i = 1, 2, \ldots, 5$ lifted into $\mathrm{T}_{[U_*]}\,\mathrm{Gr}(2, 3)$. Table 14.3 presents the performance results and Figure 14.5 presents the 2-dimensional projections of the computed optimizers for the 5 initial conditions. Table 14.4 presents the types of optima obtained by the combinations of the cost functions and the algorithms for the 20 initial conditions.

The partly smooth cost function $\phi([U])$ is Lipschitz continuous. As we discussed in Chapter 7, on such cost functions, RBFGS is typically faster than RGS due to its smaller number of function and gradient evaluations. This does not mean RGS is useless. RGS is more resistant than RBFGS

Table 14.3: The cost function values at the optimizers from 5 initial conditions for the different methods. The 2-dimensional projections of the optimizers are shown in Figure 14.5 at the indices given.

RTR-Newton	$F([U_i^*])$	1.3555	1.3573	1.3555	1.3555	1.3535
	index	(1)	(2)	(3)	(4)	(5)
RBFGS	$F([U_i^*])$	1.3537	1.3535	1.3535	1.3535	1.3555
	index	(6)	(7)	(8)	(9)	(10)
LRBFGS	$F([U_i^*])$	1.3535	1.3535	1.3555	1.3535	1.3535
	index	(11)	(12)	(13)	(14)	(15)
RTR-SR1	$F([U_i^*])$	1.3555	1.3535	1.3537	1.3573	1.3555
	index	(16)	(17)	(18)	(19)	(20)
LRTR-SR1	$F([U_i^*])$	1.3535	1.3535	1.3537	1.3573	1.3555
	index	(21)	(22)	(23)	(24)	(25)
RCG	$F([U_i^*])$	1.3555	1.3573	1.3555	1.3555	1.3535
	index	(26)	(27)	(28)	(29)	(30)
RBFGS	$\phi([U_i^*])$	2.8641_{-4}	1.4792_{-3}	7.3108_{-4}	3.1011_{-4}	5.7567_{-4}
	index	(31)	(32)	(33)	(34)	(35)
RGS	$\phi([U_i^*])$	2.0063_{-1}	1.2965_{-2}	2.0063_{-1}	2.0063_{-1}	1.8031_{-2}
	index	(36)	(37)	(38)	(39)	(40)

Table 14.4: The results of algorithms using 20 initial points in $\mathcal{Z}$. NNIP denotes the number of non-invertible projections. NIP denotes the number of invertible projections, NDP denotes the number of times the desired global optimizer was found and AT denotes the average computational time of the 20 initial conditions(second).

Function	Algorithm	NNIP	NIP(NDP)	AT
$\tilde{F}$	RTR-Newton	18	2(2)	8.27
$\tilde{F}$	RBFGS	18	2(2)	2.08
$\tilde{F}$	LRBFGS	19	1(1)	2.64
$\tilde{F}$	RTR-SR1	18	2(2)	2.59
$\tilde{F}$	LRTR-SR1	18	2(2)	3.31
$\tilde{F}$	RCG	16	4(4)	1.17_1
$\tilde{\phi}$	RBFGS	15	5(5)	1.37
$\tilde{\phi}$	RGS	0	20(16)	1.41_1

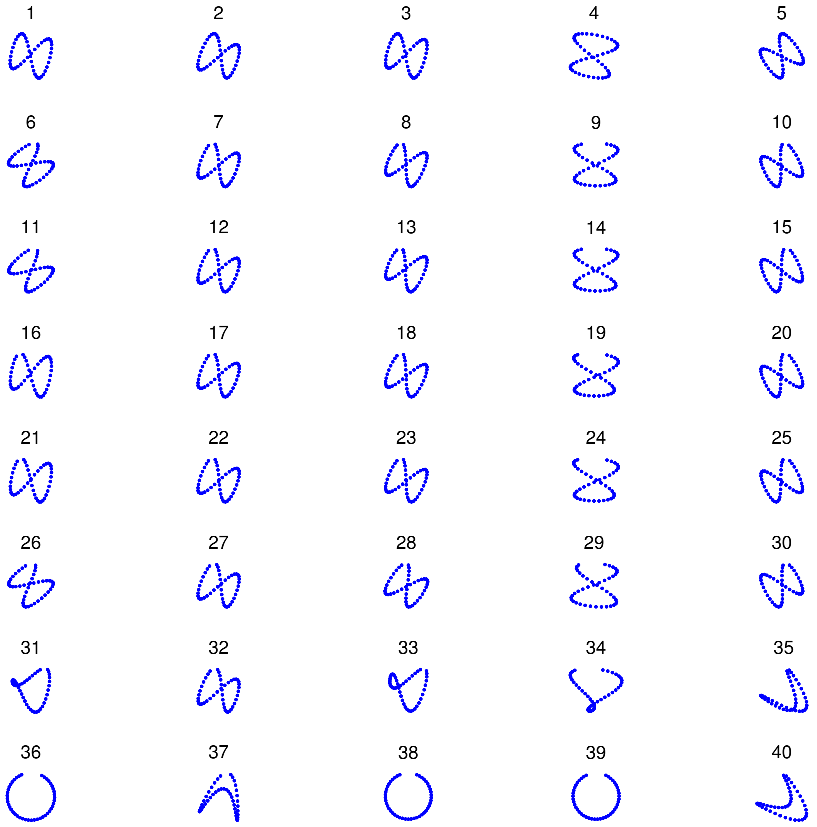


Figure 14.5: 2-dimensional projections of optimizers from 5 initial conditions and 8 algorithms for both smooth and partly smooth cost functions.

to getting stuck in the neighborhood of a local maxima. Using the same set of initial conditions, RGS is able to avoid the flat region of $\phi([U])$ and converge to the global maximizer, $[U_*]$, for 16 of the 20 initial conditions. (Since all of the sampled gradients in the flat region are fairly random in direction and RGS tends to behave more like a random walk algorithm than a line search descent method.) For the other 4 initial conditions, RGS converges to points that are only local maxima but that have the desired property of invertibility.

For the same 20 initial conditions to optimize $\tilde{\phi}([U])$, RBFGS does not perform as well. It finds 15 local maximizers that are different from one another and different from all of the maximizers found by RGS. The values of cost function $\tilde{\phi}([U])$ at the local maximizers $[U_i^*]$ of RBFGS are relatively small compared to the value at $[U_*]$ and therefore are located in the flat region $\mathcal{Z}$ which was to be avoided. Additionally, the projected curves of the $[U_i^*]$ of RBFGS are self-intercepting and the $\pi_{[U_i^*]}|_{\mathcal{M}}$ are not invertible making these local maxima not desirable solutions.

The 20 initial conditions were used also in an attempt to optimize the discretized smooth cost function $\tilde{F}([U])$. As seen from Table 14.4, all of the algorithms have difficulty to escape from local minima for the smooth cost function $\tilde{F}([U])$. The sequence generated by the algorithms with most of the initial conditions converge to similar minimizers of $\tilde{F}([U])$ and the projected curve is self-intercepting(see Figure 14.5 for 5 examples). Therefore, $\pi_{[U_i^*]}|_{\mathcal{M}}$ is not invertible and these minimizers are clearly not desirable. Furthermore, the values of $\tilde{F}([U])$ at some of these minimizers are approximately 1.353, which is even smaller than the function value, 1.355, at the desired invertible minimizer. The fact that $\tilde{F}([U])$ at the undesired minimizers is smaller than the desired one is also true when 100, 200, and 400 points are used in the discretized manifold $\tilde{\mathcal{M}}$. This indicates that discretization of $F([U])$ produces an approximation $\tilde{F}([U])$ whose global minimizer does not match the corresponding global minimizer $F([U])$. So minimizing $\tilde{F}([U])$ does not approximately solve the problem of minimizing $F([U])$. However, the value of $\tilde{\phi}([U])$ at the desired maximizer is much larger than the value at the other local maximizers (see Table 14.3). This indicates that unlike for the cost function $\tilde{F}([U])$, finding a global maximizer is sufficient to find the desired one for the cost function $\tilde{\phi}([U])$.

As seen from Table 14.4, the most efficient algorithm is RBFGS for the smooth cost function $\tilde{F}([U])$. RTR-Newton needs a relatively large number of iterations before detecting a neighborhood of a local minimizer. Therefore, RTR-Newton needs much large computational time compared to the problem with an initial point close to the desire minimizer. The RCG, used by Broomhead and Kirby [BK05], is still the slowest method for $\tilde{F}([U])$. For the partly smooth cost function $\tilde{\phi}([U])$, even though RGS needs more computational time than RBFGS, it shows its robustness of finding invertible projection.

14.5 Conclusion

We compared the performance of algorithms for $\tilde{F}(U)$ and $\tilde{\phi}(U)$ and also compared the two cost functions $\tilde{F}(U)$ and $\tilde{\phi}(U)$ for the discretized manifold $\tilde{\mathcal{M}}$. $\tilde{F}(U)$, the discretization of a smooth approximation to the original partly smooth cost function $\phi([U])$ was shown to be unsatisfactory for reliably solving the example problem. The discretization $\tilde{\phi}([U])$ does not have these difficulties. For $\tilde{\phi}([U])$, RBFGS has advantage of converging fast but does not reliably produce an invertible projection even at a local maximizer with an acceptably large cost function value. RGS exhibited the desirable ability of escaping from the flat region $\mathcal{Z}$ to move toward an acceptable invertible projection. The resulting reliability of RGS as a potential global optimization algorithm for this problem must be weighed carefully against the speed of any future modifications of RBFGS that mitigate is problems with local maxima.

CHAPTER 15

CONCLUSIONS AND FURTHER RESEARCH

In this dissertation, we generalized the Euclidean quasi-Newton methods which are used for optimizing sufficiently smooth functions to the Riemannian setting with a combination of line search and trust region strategies. In addition, two optimization algorithms, the gradient sampling algorithm and a modified version of BFGS, which are used for optimizing partly smooth functions, were generalized to the Riemannian setting. Experiments and applications were used to illustrate the value of Riemannian quasi-Newton methods.

The major contributions of this dissertation are:

1. **Generalizing the Broyden family of methods to the Riemannian setting and combining it with line search strategy;**
   This is a significant extension of the core theory and implementation for Riemannian optimization. Previous work has concentrated on generalizing the BFGS member of the Broyden family. There are two main Riemannian versions and both have strong constraints on the retraction and vector transport allowed. The version of Qi [Qi11] requires the use of the exponential mapping and parallel translation and the version of Ring and Wirth [RW12] requires the often computationally expensive differentiated retraction. Our work weakens significantly the requirements on a retraction and a vector transport and proved the RBroyden family algorithm to be well-defined.

2. **Systematic and complete analysis of the convergence properties of the line search based RBroyden family methods;**
   The convergence analyses are generalized from [DM77], [GT82], and [BNY87]. Global and linear convergence analyses are given for a twice continuously differentiable and retraction-convex cost function. Superlinear convergence is proven when the Hessian of the cost function satisfies the generalized Hölder continuity condition at the minimizer (Assumption 6.2.2).

3. **Generalizing limited-memory BFGS algorithm to the Riemannian setting;**
   The LRBFGS method in this dissertation is the first general Riemannian version of the limited-memory BFGS algorithm. Imposing the limited-memory constraint, may decrease the convergence rate, however, LRBFGS is useful in the sense that it not only reduces the complexity of storage, but also reduces the computational time for each iteration and the computational time of the vector transport in particular.

4. **Generalizing the SR1 update to the Riemannian setting and combining it with trust region strategy;**
   Previous work for the Riemannian setting based on the trust region strategy is given by Baker [Bak08]. To obtain a fast local convergent rate, his algorithm needs the action of the Hessian which may be unavailable or too expensive computationally. This dissertation avoids the requirement of the action of the Hessian by generalizing the SR1 update to the Riemannian setting while maintaining a satisfactory convergence rate. Unlike the restricted Broyden family update, SR1 update does not guarantee the positive definiteness of the Hessian approximation. However, the result of producing a better Hessian compared with restricted Broyden family update and combining it with the trust region strategy produces an effective and efficient method. Even though the known convergence rate of RTR-SR1 is slower than the RBroyden family, there are some important benefits of RTR-SR1. For example, the SR1 update is a cheaper update and its combination with trust region strategy completely avoids the requirement of the information of the differentiated retraction.

5. **Systematically analyzing the convergence properties of RTR-SR1;**
   The global convergence analysis of a Riemannian trust region approach given in [Bak08] does not require the information of the second order term in the local model. Therefore, our work focused on the local convergence analysis and it is based on the work of [CGT91], [KBS93] and [BKS96]. Comparing to the work in the Euclidean setting, we weaken the accuracy required when solving the local model. By assuming the Lipschitz continuity of the Hessian of the cost function around the minimizer, $d+1$-superlinear convergence is obtained, where d is the dimension of the manifold.

6. **Generalizing the limited-memory SR1 trust region method to the Riemannian setting;**
   Similar to LRBFGS, LRTR-SR1 is also the first general Riemannian version of a limited-memory SR1 algorithm. It also has the benefits of reducing the storage requirements and computational time per iteration.

7. **Generalizing some important concepts and theorems to the Riemannian setting based on the framework of retraction and vector transport;**
   The main Euclidean concepts and theorems generalized are:

   - Concepts:
     - Lipschitz continuity of the Hessian of a function (see Assumptions 3.3.3 and 3.3.4);
     - Convexity of a function (Definition 4.3.1);
     - Lipschitz continuity of the gradient of a function (Definition 5.2.1);
   - Theorems:

- Equivalence property of the Broyden family of methods (Theorem 4.7.1);
- Dennis Moré conditions for root solving and optimization (Theorem 5.2.2, Corollary 5.2.1 and Theorem 5.2.4).

8. **Generalizing two algorithms, the gradient sampling algorithm and modified version of RBFGS, for optimizing partly smooth functions to the Riemannian setting without convergence analysis;**
   The generalizations of the gradient sampling algorithm and modified version of BFGS are based on previous Euclidean work [BLO05] and [LO13] respectively.

9. **Efficient implementation design for four types of manifolds;**
   A frequently encountered situation is that an element in a manifold can be represented by a n-dimensional vector and this happens commonly in four situations. The implementations of metric, adjoint, linear operator and vector transport for these kinds of manifolds are discussed in detail in Chapter 9.

10. **Providing detailed efficient implementations for four particular manifolds, the sphere, the Stiefel manifold, the orthogonal group and the Grassmann manifold;**
    Implementations of different kinds of retractions, non-isometric and isometric vector transports and cotangent vectors required by Ring and Wirth's RBFGS are given in detail. The efficiency of the implementations is also discussed.

11. **Providing empirical assessments and comparisons of the performance of proposed Riemannian algorithms and existing Riemannian algorithms.**
    The preferred pair of retraction and vector transport for the RBroyden family and RTR-SR1 are identified. RBFGS ($\phi = 0$) is the best in the restricted Broyden family algorithms for the problems tested. The systematic selection of the parameter ϕ to improve performance remains an open problem. The value of limited-memory versions of RBFGS and RTR-SR1 is shown in both moderately sized and large scale problems. RBFGS shows an advantage of computational time for Lipschitz continuous partly smooth function and RGS shows an advantage of robustness for non-Lipschitz continuous partly smooth function.

12. **Applying Riemannian quasi-Newton algorithms to applications and illustrating their effectiveness and efficiency;**

    - The joint diagonalization problem for ICA
      RTR-SR1 and LRBFGS are the two relatively fastest algorithms for this problem when N is sufficiently large. RTR-Newton requires the action of the Hessian which is expensive in this problem for large N and, therefore, Riemannian quasi-Newton algorithms show advantages in computational time for large N.

- Synchronization of rotation problem
  RBFGS and LRBFGS are the two relatively fastest algorithms for this problem. Similar to the joint diagonalization problem, the action of the Hessian is also expensive in this application. RTR-Newton does not converge relatively fast.
- Rotation and reparameterization problem of curves in elastic shape analysis
  The closed form of the Hessian of the cost function in this application is unknown and, therefore, RTR-Newton cannot be applied. Riemannian quasi-Newton algorithms can be applied for this problem, even though the dimension of the domain of the cost function is infinite. The Riemannian quasi-Newton algorithms are shown to outperform RSD. LRBFGS is chosen as the representative Riemannian quasi-Newton algorithm and is significantly faster the existing method, DP, in most of the problems tested . Additionally, the quality of the solution is assessed by 1NN and LRBFGS has competitive or superior performance to DP with dense breaking points.

13. **Applying RGS and a modified version of RBFGS for a problem in secant-based nonlinear dimension reduction and illustrating their advantages;**
    Two cost functions are proposed in [BK05]. The partly smooth cost function $\phi([U])$ is shown in this dissertation to be a better cost function. The modified version of RBFGS converges quickly for initial conditions sufficiently close to the global minimizer, while RGS is more robust in the sense of escaping from the flat region of the domain and obtaining a desirable invertible projection.

14. **The Matlab and C packages have been developed .**

There are several avenues of future research in both algorithms and their applications. For algorithms, we will consider modifications to the RBroyden family algorithms to extend the convergence properties to a wider range of ϕ (especially for $\phi < 0$), to nonconvex objective functions, and without requiring information about differentiated retraction. Additionally, the convergence analysis of RGS and the modified version of RBFGS for optimizing partly smooth functions will be developed. All algorithms in this dissertation find a local minimizer of an objective function efficiently and effectively without the guarantee of finding a global minimizer. We will investigate the global optimization properties on Riemannian manifolds. Finally, one of the most important topics for future algorithmic research is the optimization of problems with constraints that are a non-manifold subset of a Riemannian manifold. Such problems arise in many areas, e.g., rank-constrained matrix approximation and reconstruction.

For applications, the growing body of advanced theory, understanding of efficient design, and implementation of efficient computational libraries are increasing the acceptance of Riemannian optimization methods. This is evident in the many applications of RTR variants in the literature. The work in this dissertation provides the promise of successfully applying Riemannian quasi-Newton algorithms to problems in many fields. We will continue to conduct systematic comparisons with existing methods to adapt and improve both the Riemannian methods and our understanding of their behaviors and relationship to application characteristics. In particular, the results will be presented in so far as it is possible from application's point of view. These fields of interest include but are not limited to large scale data mining, image analysis, signal processing, machine learning and shape analysis.

BIBLIOGRAPHY

[ABG07] P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on Riemannian manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, 2007.

[ADM02] R. L. Adler, J.-P. Dedieu, and J. Y. Margulies. Newton's method on Riemannian manifolds and a geometric model for the human spine. *IMA Journal of Numerical Analysis*, 22(3):359–390, 2002.

[AG06] P.-A. Absil and K. A. Gallivan. Joint diagonalization on the oblique manifold for independent component analysis. *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings.*, 5:V945–V948, 2006.

[AMS08] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds.* Princeton University Press, Princeton, NJ, 2008.

[ATV13] B. Afsari, R. Tron, and R. Vidal. On the convergence of gradient descent for finding the Riemannian center of mass. *SIAM Journal on Control and Optimization*, 51(3):2230–2260, 2013. arXiv:1201.0925v1.

[BA10] P. B. Borckmans and P.-A. Absil. Oriented bounding box computation using particle swarm optimization. *ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning*, pages 345–350, 2010.

[BA11] N. Boumal and P.-A. Absil. RTRMC: A Riemannian trust-region method for low-rank matrix completion. *Advances in Neural Information Processing Systems 24 (NIPS)*, pages 406–414, 2011.

[BAG08] C. G. Baker, P.-A. Absil, and K. A. Gallivan. An implicit trust-region method on Riemannian manifolds. *IMA Journal of Numerical Analysis*, 28(4):665–689, 2008.

[Bak08] C. G. Baker. *Riemannian manifold trust-region methods with applications to eigenproblems.* PhD thesis, Florida State University, 2008.

[Ber82] D. P. Bertsekas. Projected Newton methods for optimization problems with sample constraints*. *SIAM Journal on Control and Optimization*, 20(2):221–246, 1982.

[Ber03] D. P. Bertsekas. *Nonlinear programming.* Athena Scientific, second edition, 2003.

[BI13] D. A. Bini and B. Iannazzo. Computing the Karcher mean of symmetric positive definite matrices. *Linear Algebra and its Applications*, 438(4):1700–1710, February 2013. doi:10.1016/j.laa.2011.08.052.

[BK00] D. S. Broomhead and M. Kirby. A new approach to dimensionality reduction: theory and algorithms. *SIAM Journal on Applied Mathematics*, 60(6):2114–2142, 2000.

[BK05] D. S. Broomhead and M. J. Kirby. Dimensionality reduction using secant-based projection methods : the induced dynamics in projected systems. *Nonlinear Dynamics*, 41(1-3):47–67, 2005.

[BKS96] R. H. Byrd, H. F. Khalfan, and R. B. Schnabel. Analysis of a symmetric rank-one trust region method. *SIAM Journal on Optimization*, 6(4):1025–1039, 1996.

[BLN92] R. H. Byrd, D. C. Liu, and J. Nocedal. On the behavior of Broyden's class of quasi-Newton methods. *SIAM Journal on Optimization*, 2(4):533–557, 1992.

[BLO05] J. V. Burke, A. S. Lewis, and M. L. Overton. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM Journal on Optimization*, 15(3):751–779, January 2005. doi:10.1137/030601296.

[BM06] I. Brace and J. H. Manton. An improved BFGS-on-manifold algorithm for computing weighted low rank approximations. *Proceedings of 17th international Symposium on Mathematical Theory of Networks and Systems*, pages 1735–1738, 2006.

[BNS94] R. H. Byrd, J. Nocedal, and R. B. Schnabel. Representations of quasi-Newton matrices and their use in limited memory methods. *Mathematical Programming*, 63(1-3):129–156, 1994.

[BNY87] R. H. Byrd, J. Nocedal, and Y.-X. Yuan. Global convergence of a class of quasi-Newton methods on convex problems. *SIAM Journal on Numerical Analysis*, 24(5):1171–1190, 1987.

[Bor12] R. Borsdorf. *Structured matrix nearness problems: theory and algorithms*. PhD thesis, The University of Manchester, 2012.

[BSAB12] N. Boumal, A. Singer, P.-A. Absil, and V. D. Blondel. Cramer-Rao bounds for synchronization of rotations, 2012. arXiv:1211.1621v1.

[Cen] Florida State University Research Computing Center. FSU high performance computing system.

[CGT91] A. R. Conn, N. I. M. Gould, and P. L. Toint. Convergence of quasi-Newton matrices generated by the symmetric rank one update. *Mathematical Programming*, 50(1-3):177–195, March 1991. doi:10.1007/BF01594934.

[CGT00] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-region methods*. MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[Cha06] I. Chavel. *Riemannian geometry: a modern introduction.* Cambridge Studies in Advanced Mathematics, second edition, 2006.

[Cla90] F. H. Clarke. *Optimization and nonsmooth snalysis.* Classics in Applied Mathematics of SIAM, 1990.

[Dav75] W. C. Davidon. Optimally conditioned optimization algorithms without line searches. *Mathematical Programming*, 9(1):1–30, 1975.

[dC92] M. P. do Carmo. *Riemannian geometry.* Mathematics: Theory & Applications, 1992.

[DE99] L. Dieci and T. Eirola. On smooth decompositions of matrices. *SIAM Journal on Matrix Analysis and Applications*, 20(3):800–819, January 1999. doi:10.1137/S0895479897330182.

[DK70] C. Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

[DKM12] W. Dai, E. Kerman, and O. Milenkovic. A geometric approach to low-rank matrix completion. *IEEE Transactions on Information Theory*, 58(1):237–247, 2012. arXiv:1006.2086v1.

[DM74] J. E. Dennis and J. J. Moré. A characterization of superlinear convergence and its application to quasi-Newton methods *. *Mathematics of Computation*, 28(126):549–560, 1974.

[DM77] J. E. Dennis and J. J. Moré. Quasi-Newton methods, motivation and theory. *SIAM Review*, 19(1):46–89, 1977.

[DS83] J. E. Dennis and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations.* Springer, New Jersey, 1983.

[EAS98] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, January 1998. doi:10.1137/S0895479895290954.

[FST88] C. Foias, G. R. Sell, and R. Temam. Inertial manifolds for nonlinear evolutionary equations. *Journal of Differential Equations*, 73(2):309–353, 1988.

[Gab82] D Gabay. Minimizing a differentiable function over a differential manifold. *Journal of Optimization Theory and Applications*, 37(2):177–219, 1982.

[GQA12] K. A. Gallivan, C. Qi, and P.-A. Absil. A Riemannian Dennis-More condition. In Michael W. Berry, Kyle A. Gallivan, Efstratios Gallopoulos, Ananth Grama, Bernard

Philippe, Yousef Saad, and Faisal Saied, editors, *High-Performance Scientific Computing*, pages 281–293. Springer London, 2012. doi:10.1007/978-1-4471-2437-5_14.

[GT82] A. Griewank and P. L. Toint. Local convergence analysis for partitioned quasi-Newton updates. *Numerische Mathematik*, 29:429–448, 1982.

[GV96] G. H. Golub and C. F. Van Loan. *Matrix computations.* Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, third edition, 1996.

[Haa04] M. Haarala. *Large-scale nonsmooth optimization: variable metric bundle method with limited memory.* PhD thesis, University of Jyvaskyla, 2004.

[HAG13] W. Huang, P.-A. Absil, and K. A. Gallivan. A Riemannian symmetric rank-one trust-region method. *Tech. report UCL-INMA-2013.03-v1*, 2013.

[HKO01] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*, volume 7. John Wiley & Sons, Inc., 2001.

[HM94] U. Helmke and J. B. Moore. *Optimization and dynamical systems.* Spfinger-Verlag, June 1994. doi:10.1109/JPROC.1996.503147.

[IAVD11] M. Ishteva, P.-A. Absil, S. Van Huffel, and L. De Lathauwer. Best low multilinear rank approximation of higher-order tensors, based on the Riemannian trust-region scheme. *SIAM Journal on Matrix Analysis and Applications*, 32(1):115–135, 2011.

[JBAS10] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010.

[JKT90] M. S. Jolly, I. G. Kevrekidis, and E. S. Titi. Approximate inertial manifolds for the Kuramoto-Sivashinsky equation: analysis and computation. *Physica D: Nonlinear Phenomena*, 44(1C2):38–60, 1990.

[Kar84] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

[KBS93] H. F. Khalfan, R. H. Byrd, and R. B. Schnabel. A theoretical and experimental study of the symmetric rank-one update. *SIAM Journal on Optimization*, 3(1):1–24, February 1993. doi:10.1137/0803001.

[Kel99] C. T. Kelley. *Iterative methods for optimization.* Society for Industrial and Applied Mathematics, 1999.

[Kiw85] K. C. Kiwiel. *Methods of descent for nondifferentiable optimization.* Springer Berlin Heidelberg, 1985.

[KS12] M. Kleinsteuber and H. Shen. Blind source separation with compressively sensed linear mixtures. *IEEE Signal Processing Letters*, 19(2):107–110, 2012. arXiv:1110.2593v1.

[Lau04] A. J. Laub. *Matrix analysis for scientists and engineers.* SIAM, Philadelphia, PA, 2004.

[Li00] J.-R. Li. *Model reduction of large linear systems via low rank system gramians.* PhD thesis, Massachusetts Institute of Technology, 2000.

[LO13] A. S. Lewis and M. L. Overton. Nonsmooth optimization via quasi-Newton methods. *Mathematical Programming*, 141(1-2):135–163, February 2013. doi:10.1007/s10107-012-0514-2.

[Lue72] D. G. Luenberger. The gradient projection method along geodesics. *Management Science*, 18(11):620–631, 1972.

[Lue73] D. G. Luenberger. *Introduction to linear and nonlinear programming.* Addison-Wesley, 1973.

[LV07a] J. A. Lee and M. Verleysen. *Nonlinear dimensionality reduction.* Springer, 2007.

[LV07b] C. Liu and S. A. Vander Wiel. Statistical quasi-Newton: a new look at least change. *SIAM Journal on Optimization*, 18(4):1266–1285, 2007.

[LY08] D. G. Luenberger and Y. Ye. *Linear and nonlinear programming.* Springer, third edition, 2008.

[MMBS11] B. Mishra, G. Meyer, F. Bach, and R. Sepulchre. Low-rank optimization with trace norm penalty, 2011. arXiv:1112.2318v2.

[MMS11] B. Mishra, G. Meyer, and R. Sepulchre. Low-rank optimization for distance matrix completion. *In Proceeding of 50th IEEE Conference on Decision and Control and European Control Conference*, pages 4455–4460, December 2011. doi:10.1109/CDC.2011.6160810.

[MT89] M. Marion and R. Temam. Nonlinear Galerkin methods. *SIAM Journal on Numerical Analysis*, 26(5):1139–1157, October 1989. doi:10.1137/0726063.

[NH95] I. Najfeld and T. F. Havel. Derivatives of the matrix exponential and their computation. *Advances in Applied Mathematics*, 16(3):321–375, 1995.

[NW99] J. Nocedal and S. J. Wright. *Numerical optimization.* Springer, 1999.

[NW06] J. Nocedal and S. J. Wright. *Numerical optimization.* Springer, second edition, 2006.

[Pow76] M. J. D. Powell. Some global convergence properties of a variable metric algorithm for minimization without exact line searches. *Nonlinear Programming, SIAM-AMS Proceedings*, 9, 1976.

[Pow86] M. J. D. Powell. How bad are the BFGS and DFP method when the objective function is quadratic? *Mathematical Programming*, 34:34–37, 1986.

[QGA10] C. Qi, K. A. Gallivan, and P.-A. Absil. Riemannian BFGS algorithm with applications. *Recent Advances in Optimization and its Applications in Engineering*, pages 183–192, 2010.

[Qi11] C. Qi. *Numerical optimization methods on Riemannian manifolds.* PhD thesis, Florida State University, 2011.

[QZL05] L. Qiu, Y. Zhang, and C.-K. Li. Unitarily invariant metrics on the Grassmann space. *SIAM Journal on Matrix Analysis and Applications*, 27(2):507–531, 2005.

[Rew03] M. J. Rewienski. *A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems.* PhD thesis, Massachusetts Institute of Technology, 2003.

[Rit79] K. Ritter. Local and superlinear convergence of a class of variable metric methods. *Computing*, 23:287–297, 1979.

[Rit81] K. Ritter. Global and superlinear convergence of a class of variable metric methods. *Mathematical Programming*, 15:178–205, 1981.

[Rob12] D. T. Robinson. *Functional data analysis and partial shape matching in the square root velocity framework.* PhD thesis, Florida State University, 2012.

[RV74] A. W. Roberts and D. E. Varberg. Another proof that convex functions are locally lipschitz. *The American Mathematical Monthly*, 81(9):1014–1016, 1974.

[RW12] W. Ring and B. Wirth. Optimization methods on Riemannian manifolds and their application to shape space. *SIAM Journal on Optimization*, 22(2):596–627, January 2012. doi:10.1137/11082885X.

[SAGQ12] S. E. Selvan, U. Amato, K. A. Gallivan, and C. Qi. Descent algorithms on oblique manifold for source-adaptive ICA contrast. *IEEE Transactions on Neural Networks and Learning Systems*, 23(12):1930–1947, 2012.

[San10] O. Sander. Geodesic finite elements for Cosserat rods. *International Journal for Numerical Methods in Engineering*, 82(13):1645–1670, 2010. doi:10.1002/nme.2814.

[Sch78] R. B. Schnabel. Optimal conditioning in the convex class of rank two updates. *Mathematical Programming*, 15(1):247–260, 1978.

[Ska10] A. Skajaa. *Limited memory BFGS for nonsmooth optimization.* PhD thesis, New York University, 2010.

[SKH13] M. Seibert, M. Kleinsteuber, and K. H\"uper. 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*, pages 395–412, 2013.

[SKJJ11] A. Srivastava, E. Klassen, S. H. Joshi, and I. H. Jermyn. Shape analysis of elastic curves in Euclidean spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1415–1428, September 2011. doi:10.1109/TPAMI.2010.184.

[SKK03] T. B. Sebastian, P. N. Klein, and B. B. Kimia. On aligning curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):116–125, January 2003. doi:10.1109/TPAMI.2003.1159951.

[SL10] B. Savas and L. H. Lim. Quasi-Newton methods on Grassmannians and multilinear approximations of tensors. *SIAM Journal on Scientific Computing*, 32(6):3352–3393, 2010.

[Smi94] S. T. Smith. Optimization techniques on Riemannian manifolds. *Hamiltonian and Gradient Flows, Algorithms and Control*, 3:113–136, 1994.

[Sta81] A. Stachurski. Superlinear convergence of Broyden's bounded theta-class of methods. *Mathematical Programming*, 20(1):196–212, 1981.

[Sto75] J. Stoer. On the convergence rate of imperfect minimization algorithms in Broyden's beta class. *Mathematical Programming*, 9(1):313–335, 1975.

[TAJP08] P. Tabuada, A. D. Ames, A. Julius, and G. J. Pappas. Approximate reduction of dynamic systems. *Systems and Control Letters*, 57(7):538–545, July 2008. doi:10.1016/j.sysconle.2007.12.005.

[TCA09] F. J. Theis, T. P. Cason, and P.-A. Absil. Soft dimension reduction for ICA by joint diagonalization on the Stiefel manifold. *Proceedings of the 8th International Conference on Independent Component Analysis and Signal Separation*, 5441:354–361, 2009.

[TVSC11] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2273–86, November 2011. doi:10.1109/TPAMI.2011.52.

[Uni] Temple University. Shape similarity research project.

[Van12] B. Vandereycken. Low-rank matrix completion by Riemannian optimization—extended version, 2012.

[VV10] B. Vandereycken and S. Vandewalle. A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2553–2579, January 2010. doi:10.1137/090764566.

[WBX$^+$07] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang. A leaf recognition algorithm for plant classification using probabilistic neural network. *2007 IEEE International Symposium on Signal Processing and Information Technology*, pages 11–16, 2007. arXiv:0707.4289v1.

[Whi36] H. Whitney. Differentiable manifolds. *The Annals of Mathematics*, 37(3):645–680, 1936.

[WY12] Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming, Published online*, August 2012. doi:10.1007/s10107-012-0584-1.

[YVGS08] Jin Yu, S. V. N. Vishwanathan, S. Gunter, and N. N. Schraudolph. A quasi-Newton approach to non-smooth convex optimization. *Proceedings of the 25th International Conference on Machine Learning*, pages 1216–1223, 2008.

[ZT88] Y. Zhang and R. P. Tewarson. Quasi-Newton algorithms with updates from the preconvex part of Broyden's family. *IMA Journal of Numerical Analysis*, 8(4):487–509, 1988.

[ZZ04] Z. Zhang and H. Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Scientific Computing*, 26(1):313–338, January 2004. doi:10.1137/S1064827502419154.

[ZZA$^+$00] S. Zhang, X. Zou, J. Ahlquist, I. M. Navon, and J. G. Sela. Use of differentiable and nondifferentiable optimization algorithms for variational data assimilation with discontinuous cost functions. *Monthly Weather Review*, 128(12):4031–4044, 2000.

BIOGRAPHICAL SKETCH

Wen Huang, son of Weixing Huang and Zhu Jiang, was born on October 8th, 1985 in Fu'an, Fujian province of P.R. China. He has a elder-sister called Shu Huang. He finished his Bachelor degree in Information and Computing Science in 2007 at the University of Science and Technology of China and worked at G-bits Network Technology Co., Ltd as a numerical designer from 2007 to 2008. He enrolled in the Ph.D. program of the Department of Mathematics at Florida State University on August 2008 and worked with Prof. Kyle A. Gallivan and Prof. Pierre-Antoine Absil.

Wen's research topics include nonlinear dimension reduction, optimization on Riemannian manifolds and their application to problems such as statistical shape analysis. He developed software, called TreeScaper, for phylogenetic analysis and a toolbox for Riemannian optimization. He made significant contributions to Riemannian quasi-Newton optimization methods.

After his Ph.D., Wen will start his post doctoral research position in the Mathematical Engineering Department of the Catholic University of Louvain.