# Riemannian Optimization Algorithms and Their Applications to Numerical Linear Algebra

Hiroyuki Sato

Department of Applied Mathematics and Physics,
Graduate School of Informatics, Kyoto University,
606-8501 Kyoto, Japan

September, 2013

# Preface

Optimization is minimization or maximization of a given real-valued function with or without some constraints on its independent variables. For optimization problems with continuous variables, a possible way to solve such an optimization problem is to generate a sequence which approaches a solution in the search space. In the case of linear programming, one thinks of interior point methods instead of the simplex method. From a viewpoint of numerical computation, it is preferable that the search space is endowed with coordinate systems. This suggests that continuous optimization problems should be dealt with on manifolds. One of the simplest examples of manifolds is the Euclidean space, which is a vector space endowed with an inner product and can be covered with only one coordinate system. A number of researches on optimization methods on the Euclidean space have been done. Methods for solving unconstrained optimization problems on the Euclidean space, which include the steepest descent, Newton's, and the conjugate gradient methods, exploit natural geometrical objects such as straight lines, the Euclidean gradient of the objective function, and so on. These methods are called unconstrained optimization methods.

However, if unconstrained optimization methods are applied to problems with constraints, they may fail to solve the problems by generating sequences subject to the constraints, since the constraints are not taken into account in the procedure of such methods for generating sequences. It is to be noted that the Euclidean space is the simplest example of a manifold. In particular, constraints resulting from problems in numerical linear algebra naturally define submanifolds of the Euclidean space, which are endowed with respective induced metrics. In order to generalize optimization methods so as to be effective on manifolds, the manifolds should be taken as Riemannian manifolds, since geometric objects such as straight line and the gradient are easily extended on Riemannian manifolds. Beyond Euclidean optimization, this thesis studies several topics of Riemannian optimization, where the adjective "Riemannian" stems from Riemannian geometry, which is the differential geometry of smooth manifolds with Riemannian metric. If an optimization problem is subject to some constraints in the Euclidean space and if the search space of the problem forms a Riemannian manifold, it is natural to employ the Riemannian version of unconstrained optimization methods, which are generalizations of Euclidean unconstrained methods. Furthermore, once a Riemannian optimization method has been developed for a general Riemannian optimization problem, the method can be applied not only to problems

on submanifolds of the Euclidean space, but also to those on more abstract Riemannian manifolds which are not necessarily embedded to the Euclidean space.

In this thesis, as a part of theoretical researches on general Riemannian optimization methods, the existing conjugate gradient method is improved and the global convergence analysis for the algorithm proposed in the improved method is provided. On the other hand, from an application point of view, Riemannian optimization methods are set up for the real and complex singular value decomposition problems to propose new algorithms with high accuracy. The performance of computers has been improved dramatically to support science and technology. However, it is also important to develop and improve various algorithms theoretically. The author hopes that the thesis makes contribution to the field of Riemannian optimization and will be helpful for further development of the research.

Hiroyuki Sato
September 2013

# Acknowledgment

I wish to express my deepest gratitude to Professor Toshihiro Iwai for plenty of insightful and valuable advice on the studies and for continuous and encouraging support throughout my graduate program. He carefully read my manuscripts including this thesis over and over again to significantly brush them up. I cannot thank him enough and I shall never forget his kindness. I would like to express my sincere appreciation to Assistant Professor Yoshiyuki Y. Yamaguchi for critical reading of the thesis and for a lot of fruitful discussions. He always treated me warmly and made me have more incentive.

I am also deeply grateful to Professor Yoshimasa Nakamura for giving me opportunities to make presentations of my work in his laboratory and providing me with valuable comments. He kindly took care of me during academic conferences. I would like to express my sincere thanks to Professor Naoshi Nishimura and Associate Professor Nobuo Yamashita for valuable comments on the thesis and for useful discussions.

It was very pleasant for me to study and discuss various topics with the members in Dynamical Systems Theory Laboratory, especially with Dr. Shun Ogawa. I would like to tender my acknowledgments to them.

# Contents


**1 Introduction** **1**

1.1 Overview of Euclidean and Riemannian optimization . . . . . 1

1.1.1 Euclidean optimization . . . . . 1

1.1.2 Some motivating examples for Riemannian optimization . . . . . 2

1.1.3 Optimization algorithms on the Euclidean space and on Riemannian manifolds . . . . . 9

1.2 Overview of the topics in the thesis . . . . . 9

1.2.1 Riemannian conjugate gradient method . . . . . 10

1.2.2 Singular value decomposition of a real matrix and the corresponding optimization problem on a real product manifold . . . . . 10

1.2.3 Complex singular value decomposition and Riemannian optimization 11

1.3 Outline of the thesis . . . . . 12

**2 A Brief Review of Optimization Methods and Basic Facts on the Real Stiefel Manifold** **14**

2.1 A Review of unconstrained optimization methods on the Euclidean space . 14

2.1.1 General framework . . . . . 14

2.1.2 Exact line search and the Armijo and the Wolfe conditions . . . . . 16

2.1.3 Steepest descent method . . . . . 18

2.1.4 Newton's method . . . . . 19

2.1.5 Conjugate gradient method . . . . . 20

2.2 Riemannian optimization methods . . . . . 23

2.2.1 Line search and retraction . . . . . 23

2.2.2 Vector transport . . . . . 25

2.2.3 Steepest descent method . . . . . 26

2.2.4 Newton's method . . . . . 27

2.2.5 Conjugate gradient method . . . . . 28

2.3 Riemannian geometry of the real Stiefel manifold . . . . . 29

2.3.1 Tangent spaces and the induced metric . . . . . 30

2.3.2 Geodesics . . . . . 32

2.3.3 Retractions . . . . . 33

2.3.4 Vector transport . . . . . 34

**3 A New, Globally Convergent Riemannian Conjugate Gradient Method 35**
3.1 Introduction . . . . . 35
3.2 Setup for Riemannian optimization . . . . . 36
3.2.1 Vector transport and scaled vector transport . . . . . 36
3.2.2 Strong Wolfe conditions . . . . . 38
3.3 A new conjugate gradient method on a Riemannian manifold . . . . . 39
3.4 Convergence analysis of the new algorithm . . . . . 41
3.4.1 Zoutendijk's theorem . . . . . 41
3.4.2 Global convergence . . . . . 42
3.5 Numerical experiments . . . . . 45
3.5.1 A sphere endowed with a peculiar metric . . . . . 45
3.5.2 The sphere endowed with the orthographic retraction . . . . . 50
3.6 Summary . . . . . 52
3.7 Appendix: Examples in which the condition (3.4.2) holds . . . . . 53

**4 A Riemannian Optimization Approach to the Matrix Singular Value Decomposition 58**
4.1 Introduction . . . . . 58
4.2 The singular value decomposition and a Riemannian optimization problem 59
4.3 The Riemannian geometry of $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ . . . . . 63
4.3.1 Tangent spaces . . . . . 63
4.3.2 Geodesics . . . . . 64
4.3.3 Retractions . . . . . 65
4.3.4 The gradient and the Hessian of the objective function . . . . . 66
4.4 Optimization algorithms on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ . . . . . 68
4.4.1 The steepest descent method on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ . . . . . 68
4.4.2 The conjugate gradient method on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ . . . . . 70
4.4.3 Newton's method on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ . . . . . 75
4.5 New algorithms for the singular value decomposition based on optimization methods on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ . . . . . 78
4.5.1 An algorithm for computing the largest singular value and associated left and right singular vectors . . . . . 78
4.5.2 An algorithm for computing the $p$ largest singular values and associated left and right singular vectors . . . . . 78
4.5.3 Accuracy of numerical solutions . . . . . 81
4.6 Degenerate optimal solutions . . . . . 82
4.7 Summary . . . . . 90

**5 A Complex Singular Value Decomposition Algorithm Based on the Riemannian Newton Method** **91**
5.1 Introduction . . . . . 91
5.2 Complex singular value decomposition and the corresponding Riemannian optimization problem . . . . . 92
5.2.1 Setting up . . . . . 92
5.2.2 Realization of $\mathrm{St}(p, n, \mathbb{C})$ as the intersection of the real Stiefel manifold and the quasi-symplectic set . . . . . 96
5.3 Riemannian geometry of $\mathrm{Stp}(p, m) \times \mathrm{Stp}(p, n)$ . . . . . 98
5.3.1 Tangent spaces and the orthogonal projection . . . . . 98
5.3.2 Retraction . . . . . 100
5.3.3 The gradient and the Hessian . . . . . 101
5.4 Newton's method and a new complex singular value decomposition algorithm 101
5.4.1 Newton's method for Problem 5.2.3 . . . . . 101
5.4.2 Newton's method for Problem 5.2.1 . . . . . 102
5.4.3 Complex singular value decomposition algorithm based on the Riemannian Newton method . . . . . 106
5.5 Summary . . . . . 107

**6 Concluding Remarks** **109**

# Chapter 1

# Introduction

## 1.1 Overview of Euclidean and Riemannian optimization

### 1.1.1 Euclidean optimization

A problem of minimizing or maximizing the value of a given real-valued function with or without some constraint conditions is called an optimization problem. The function to be minimized or maximized in the problem is called the objective function. Optimization problems appear not only in science and engineering but also in economics and many other fields. Optimization problems are classified into several classes according to the characteristics of their properties and a number of methods for solutions have been developed [All07, Diw08, GGT04].

In some optimization problems, the variables of the objective function are restricted to be discrete values such as integers. Problems of this type are called discrete optimization problems [NW88, Sch03]. The traveling salesman problem [LLKS85] and the knapsack problem [KPP04] are typical examples of discrete optimization problems. In contrast with this, if the variables are continuous ones ranging over a domain, the problems are referred to as continuous optimization problems [Ber99, FH10, LY08, NW06, Rus06, Sny05], which are mainly focused on in this thesis. The continuous variables are usually real numbers. Continuous optimization problems are conventionally described on the Euclidean space $\mathbb{R}^n$ and formulated as follows:

**Problem 1.1.1.**

$$\begin{aligned} \text{minimize} \quad & f(x), && && (1.1.1)\\ \text{subject to} \quad & g_i(x) = 0, && i \in \mathcal{E} && (1.1.2)\\ & h_j(x) \leq 0, && j \in \mathcal{I}, && (1.1.3)\\ & x \in \mathbb{R}^n, && && (1.1.4) \end{aligned}$$

where the objective function $f$ and constraint functions $g_i$, $h_j$ are real-valued functions on $\mathbb{R}^n$ and often assumed to be smooth, and where $\mathcal{E}$ and $\mathcal{I}$ are sets of indices for equality and inequality constraints, respectively. Problem 1.1.1 with either the set $\mathcal{E}$ or $\mathcal{I}$ being non-empty should be called a constrained optimization problem on $\mathbb{R}^n$. To the contrary, if both $\mathcal{E}$ and $\mathcal{I}$ are empty, that is, (1.1.2) and (1.1.3) are non-existent, then Problem 1.1.1 becomes an unconstrained optimization problem on $\mathbb{R}^n$ [DS83, Fle13].

Optimization on the Euclidean space is called Euclidean optimization. In this thesis, as far as Euclidean optimization techniques are discussed, $f$, $g_i$, and $h_j$ in Problem 1.1.1 are assumed to be smooth unless otherwise noted. Continuous optimization problems are typically solved by iterative algorithms which generate sequences of points converging to respective solutions. Methods for solving unconstrained optimization problems are generally simpler than those for solving constrained ones. Unconstrained optimization methods include the steepest descent method, Newton's method and the quasi-Newton method [Bro70, Fle70, Gol70, Sha70], the conjugate gradient method [HS52, FR64], and so on. Since these methods exploit information of the objective function without attention to the constraints, generated sequences are not subject to the constraints in general, so that such methods fail to solve constrained optimization problems. In order to solve constrained optimization problems, the augmented Lagrangian method [Hes69, Pow73], the interior point method [FGW02], the sequential quadratic programming method [PM76], and so on, are employed. In addition, constrained optimization problems are classified into more concrete categories, such as linear programming [Kar84], quadratic programming [GHN01], second-order cone programming [AG03], semidefinite programming [VB96], and so on, in which optimization methods have been individually developed.

### 1.1.2 Some motivating examples for Riemannian optimization

There are typical optimization problems which should be solved beyond the traditional frameworks on the Euclidean space. For a given $n \times n$ real symmetric matrix $A$, we consider the following practical optimization problem:

**Problem 1.1.2.**

$$\text{minimize} \quad \frac{x^T A x}{x^T x}, \tag{1.1.5}$$

$$\text{subject to} \quad x \in \mathbb{R}^n,\ x \neq 0, \tag{1.1.6}$$

where the superscript $T$ denotes the transposition of a vector (this notation will be also used for the transposition of a matrix later). The objective function of the problem is called the Rayleigh quotient. According to the Courant-Fischer min-max theorem [GVL12], an optimal solution to Problem 1.1.2 is an eigenvector associated with the smallest eigenvalue of $A$. Problem 1.1.2 is an unconstrained optimization problem on $\mathbb{R}^n_*$, where $\mathbb{R}^n_*$ is $\mathbb{R}^n$ with the origin removed. Against expectations, almost all sequences generated by Newton's

method for the problem do not even converge to a local optimal solution in general. Indeed, according to [AMS08], Newton's equation for the search direction $\xi_k \in \mathbb{R}^n$ at the $k$-th iterate $x_k \in \mathbb{R}^n$ (see Chapter 2) is written out as

$$\frac{2}{x_k^T x_k}\left(I_n - 2\frac{x_k x_k^T}{x_k^T x_k}\right)\left(A - \frac{x_k^T A x_k}{x_k^T x_k} I_n\right)\left(I_n - 2\frac{x_k x_k^T}{x_k^T x_k}\right)\xi_k = -\frac{2}{x_k^T x_k}\left(A - \frac{x_k^T A x_k}{x_k^T x_k} I_n\right) x_k, \tag{1.1.7}$$

where $I_n$ is the identity matrix of size $n$. The equation (1.1.7) has a unique solution $\xi_k = x_k$ if and only if the Rayleigh quotient $x_k^T A x_k / x_k^T x_k$ is not an eigenvalue of $A$. For $\xi_k = x_k$, the resulting next point is $x_{k+1} = x_k + \xi_k = 2x_k$. Therefore, almost all sequences generated by Newton's method for this problem cannot converge to any stationary point $x_*$ except for the case that the origin 0, the initial point $x_0$, and the target point $x_*$ are on the same straight line and $x_0^T A x_0 / x_0^T x_0$ is an eigenvalue of $A$. Thus, we cannot obtain an optimal solution by Newton's method in general.

Another set up for Problem 1.1.2 is possible on account of the scale invariance of the objective function (1.1.5). Using the scale invariance, one can translate the problem into an equivalent problem of the form:

**Problem 1.1.3.**

$$\begin{aligned} \text{minimize} \quad & x^T A x, && (1.1.8)\\ \text{subject to} \quad & x^T x = 1, && (1.1.9)\\ & x \in \mathbb{R}^n. && (1.1.10) \end{aligned}$$

Problem 1.1.3 is a constrained optimization problem on $\mathbb{R}^n$, to which the augmented Lagrangian method can be applied. The augmented Lagrangian function $L_A(x, \lambda; \mu)$ for Problem 1.1.3 is defined as

$$L_A(x, \lambda; \mu) := x^T A x - \lambda(x^T x - 1) + \mu(x^T x - 1)^2. \tag{1.1.11}$$

The presence of the last squared term in the right-hand side of (1.1.11) makes the augmented Lagrangian method different from the standard Lagrangian method. The $L_A$ is viewed as a combination of the standard Lagrangian function and the quadratic penalty function. In the augmented Lagrangian method, not only $x$ but also $\lambda$ and $\mu$ are to be updated at every iterate in order to make the sequence reach optimal ones. The updating of $\lambda$ and $\mu$ is somewhat artificial and also critical for the convergence property of the algorithm.

If we introduce the notion of manifold, some of constrained optimization problems can be viewed as unconstrained optimization problems, to which unconstrained optimization techniques such as the steepest descent and Newton's methods, if adapted suitably on manifolds, can be applied to show better convergence properties. If we use the notion of sphere, Problem 1.1.3 is put verbatim in the form,

**Problem 1.1.4.**

$$\text{minimize} \quad x^T A x, \tag{1.1.12}$$
$$\text{subject to} \quad x \in S^{n-1}, \tag{1.1.13}$$

where $S^{n-1} := \left\{x \in \mathbb{R}^n \,|\, x^T x = 1\right\}$ denotes the $(n-1)$-dimensional unit sphere. The objective function is considered to be defined on $S^{n-1}$ and Problem 1.1.4 is regarded as an unconstrained optimization problem on $S^{n-1}$. However, traditional unconstrained optimization techniques on the Euclidean space in the original form cannot be applied to Problem 1.1.4, since the linear structure of the domain of the objective function is lost. It is worth pointing out that the sphere $S^{n-1}$ is an example of manifolds. Further, $S^{n-1}$ can be endowed with the natural induced Riemannian metric from the standard Euclidean inner product $\mathbb{R}^n$ through

$$\langle \xi, \eta \rangle_x = \xi^T \eta, \qquad \xi, \eta \in T_x S^{n-1}, \tag{1.1.14}$$

where we regard tangent vectors in $T_x S^{n-1}$ as vectors in $\mathbb{R}^n$. A manifold endowed with a Riemannian metric is called a Riemannian manifold and optimization on Riemannian manifolds is called Riemannian optimization. Riemannian optimization techniques have been intensively researched and developed in the last two decades, in order to solve optimization problems on Riemannian manifolds such as Problem 1.1.4.

In their paper entitled "The geometry of algorithms with orthogonality constraints," Edelman, Arias, and Smith set up several algorithms for unconstrained problems on the Stiefel and the Grassmann manifolds [EAS98]. In their book entitled "Optimization Algorithms on Matrix Manifolds," Absil, Mahony, and Sepulchre developed algorithms for optimization problems on a general Riemannian manifold and discussed the convergence properties of the algorithms [AMS08]. A detailed exposition of these results will be provided together with a review of optimization methods on the Euclidean space in Chapter 2.

Like the Rayleigh quotient problem on the sphere, which is related to the smallest eigenvalue and the associated eigenvector of a symmetric matrix, some of Riemannian optimization problems can be associated to problems in numerical linear algebra. For example, Problem 1.1.4 is generalized to a problem on the real Stiefel manifold $\mathrm{St}(p, n) := \left\{Y \in \mathbb{R}^{n \times p} \,|\, Y^T Y = I_p\right\}$ with the integer $p$ not greater than $n$ as follows:

**Problem 1.1.5.**

$$\text{minimize} \quad \mathrm{tr}(Y^T A Y N), \tag{1.1.15}$$
$$\text{subject to} \quad Y \in \mathrm{St}(p, n), \tag{1.1.16}$$

where $N$ is a constant diagonal matrix of the form $N = \mathrm{diag}(\mu_1, \ldots, \mu_p)$ with $0 < \mu_1 < \cdots < \mu_p$. It can be shown that the $j$-th column of an optimal solution $Y_*$ to this problem is

a normalized eigenvector associated with the $j$-th smallest eigenvalue of $A$. The diagonal matrix $N$ in the problem is necessary to ensure that the columns of the $Y_*$ are sorted in ascending order of the corresponding eigenvectors from the left.

If eigenvectors themselves are not of interest but only the linear subspace spanned by $p$ eigenvectors associated with the $p$ smallest eigenvalues is of importance, then the matrix $N$ in Problem 1.1.5 should be removed. The resulting function $\mathrm{tr}(Y^TAY)$ has $O(p)$ invariance, where $O(p)$ is the orthogonal group, so that the search space should be reduced to the quotient manifold $\mathrm{St}(p,n)/O(p)$, which is called the Grassmann manifold denoted by $\mathrm{Grass}(p,n)$. The resulting problem on the Grassmann manifold $\mathrm{Grass}(p,n)$ is expressed as follows:

**Problem 1.1.6.**

$$\begin{aligned} \text{minimize} \quad & \mathrm{tr}(Y^TAY), && (1.1.17)\\ \text{subject to} \quad & [Y] \in \mathrm{Grass}(p,n) := O(n)/\mathrm{St}(p,n), && (1.1.18) \end{aligned}$$

where $[Y] \in \mathrm{Grass}(p,n)$ denotes the equivalence class represented by $Y \in \mathrm{St}(p,n)$. An optimal solution $[Y_*]$ to the problem corresponds to the space spanned by eigenvectors associated with the $p$ smallest eigenvalues. An important point to note here is that the Grassmann manifold $\mathrm{Grass}(p,n) = O(n)/\mathrm{St}(p,n)$ in this form is not embedded to the Euclidean space, while the sphere $S^{n-1}$ and $\mathrm{St}(p,n)$ are naturally embedded to $\mathbb{R}^n$ and $\mathbb{R}^{n\times p}$, respectively. The Grassmann manifold $\mathrm{Grass}(p,n)$ can be, however, viewed as the set of all orthogonal projection matrices of rank $p$ [HM94, HHT07];

$$\begin{aligned} \mathrm{Grass}(p,n) &\simeq \left\{X \in \mathbb{R}^{n\times n} | X^T = X,\ X^2 = X,\ \mathrm{rank}(X) = p\right\} && (1.1.19)\\ &= \left\{X = YY^T | Y \in \mathrm{St}(p,n)\right\}. && (1.1.20) \end{aligned}$$

This view together with Problem 1.1.6 leads us to the following problem:

**Problem 1.1.7.**

$$\begin{aligned} \text{minimize} \quad & \mathrm{tr}(AX), && (1.1.21)\\ \text{subject to} \quad & X \in \mathrm{Grass}(p,n) := \left\{X \in \mathbb{R}^{n\times n} | X^T = X,\ X^2 = X,\ \mathrm{rank}(X) = p\right\}. && (1.1.22) \end{aligned}$$

The joint diagonalization problem is another problem which can be formulated as a Riemannian optimization problem on $\mathrm{St}(p,n)$. It is well known that the mutually commuting symmetric matrices are simultaneously diagonalizable. Let $A_1, A_2, \ldots, A_K$ be $K$ real $n \times n$ symmetric matrices, which does not necessarily mutually commute. For these $K$ matrices, the approximate joint diagonalization problem can be formulated on the Stiefel manifold $\mathrm{St}(p,n)$ as follows [TCA09]:

**Problem 1.1.8.**

$$\text{maximize} \quad \sum_{l=1}^{K} \|\mathrm{diag}(Y^T A_l Y)\|_F^2, \tag{1.1.23}$$

$$\text{subject to} \quad Y \in \mathrm{St}(p,n), \tag{1.1.24}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of the matrix concerned, and $\mathrm{diag}(\cdot)$ denotes the diagonal part of the matrix in the parentheses. The present problem is closely related to the independent component analysis [CA02, Car99, CS93]. Problems 1.1.5, 1.1.6, and 1.1.8 can be also formulated as constrained Euclidean optimization problems. However, the advantage of solving them as Riemannian optimization problems is that we can make full use of the geometrical structures of the manifolds just as the case of Problem 1.1.4.

An important problem other than the eigenvalue problem in numerical linear algebra is the singular value decomposition problem. The singular value decomposition is a very important matrix factorization in frequent use in various fields such as signal and image processing, control theory, and statistics [GVL12,WB12,ZCW12]. While the latter problem is also expected to be formulated as an optimization problem on a Riemannian manifold, such a research had not been developed. One of aims of the present thesis is to study the singular value decomposition problem in the form of a Riemannian optimization problem.

The singular value decomposition can be applied to any matrices, even if they are rectangular ones. While the eigenvalue decomposition of an $n \times n$ real symmetric matrix $A$ takes the form

$$A = P\Lambda P^T, \qquad P \in O(n),\ \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n), \tag{1.1.25}$$

the singular value decomposition of an $m \times n$ real matrix $A$ with $m \geq n$ takes the form

$$A = U\Sigma V^T, \qquad U \in O(m),\ V \in O(n),\ \Sigma = \begin{pmatrix} \Sigma_1 \\ 0 \end{pmatrix}, \tag{1.1.26}$$

where $\Sigma_1 = \mathrm{diag}\,(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$. In [HM94], the problem of a full decomposition of $A$ into the form (1.1.26) is translated into an optimization problem on the product manifold $O(m) \times O(n)$, that is,

**Problem 1.1.9.**

$$\text{maximize} \quad \mathrm{tr}(U^T A V N), \tag{1.1.27}$$

$$\text{subject to} \quad (U,V) \in O(m) \times O(n), \tag{1.1.28}$$

where $N = \begin{pmatrix} N_1 & 0 \end{pmatrix} \in \mathbb{R}^{n \times m}$, $N_1 = \mathrm{diag}(\mu_1, \ldots, \mu_p)$ with $\mu_1 > \cdots > \mu_n > 0$. Since Problem 1.1.5 about a truncated eigenvalue decomposition has a matrix variable $Y \in \mathrm{St}(p,n)$, which corresponds to the matrix $P \in O(n)$ in (1.1.25), the truncated singular value decom-

position problem, which is to find the largest $p$ ($\leq n$) singular values and the corresponding left and right singular vectors, is expected to be formulated as a Riemannian optimization problem on a manifold other than $O(m) \times O(n)$. Indeed, Absil *et al.* [AMS08] suggest to replace the manifold $O(m) \times O(n)$ in Problem 1.1.9 with the manifold $\mathrm{St}(p, m) \times \mathrm{St}(p, n)$ (see Problem 1.2.1). We will prove that this generalized problem is truly equivalent to the truncated singular value decomposition problem and will develop optimization algorithms for it.

The problems mentioned in this subsection are listed in Table 1.1. We here note that maximizing problems are rewritten into minimizing problems by multiplying the objective functions by $-1$ in the table. Table 1.2 shows the progresses in the study of the problems.

Table 1.1: Several Riemannian optimization problems. The matrices $A$ and $A_1, \ldots, A_K$ are the target matrices to be decomposed in the problems. The diagonal matrices $N$ in Problems 1.1.5, 1.1.9, and 1.2.1 are constant matrices. All of the other vector and matrices such as $x$ and $X, Y$ in the objective functions are variables on the manifolds in question.

| Problem No. | Manifold | Objective function | Attainment |
|---|---|---|---|
| 1.1.4 | $S^{n-1}$ | $x^T A x$ | Leftmost eigenvector |
| 1.1.5 | $\mathrm{St}(p, n)$ | $\mathrm{tr}(Y^T A Y N)$ | $p$ leftmost eigenvectors |
| 1.1.6 | $\mathrm{Grass}(p, n)$ as a quotient manifold | $\mathrm{tr}(Y^T A Y)$ | Leftmost $p$-dimensional eigenspace |
| 1.1.7 | $\mathrm{Grass}(p, n)$ as a submanifold | $\mathrm{tr}(AX)$ | Leftmost $p$-dimensional eigenspace |
| 1.1.8 | $\mathrm{St}(p, n)$ | $-\sum_{l=1}^{K} \|\mathrm{diag}(Y^T A_l Y)\|_F^2$ | Approximate joint diagonalizing matrix |
| 1.1.9 | $O(m) \times O(n)$ | $\mathrm{tr}(U^T A V N)$ | Full SVD |
| 1.2.1 (in Sec. 1.2.2) | $\mathrm{St}(p, m) \times \mathrm{St}(p, n)$ | $\mathrm{tr}(U^T A V N)$ | Truncated SVD |

Table 1.2: Progresses in the study of the problems in Table 1.1. We note that the reference [SI13] is the author's paper on which Chapter 4 of this thesis is based.

| Problem No. | Progress | Reference |
|---|---|---|
| 1.1.4 | Well studied. | [AMS08, Bro93] |
| 1.1.5 | Well studied. | [EAS98, HM94] |
| 1.1.6 | Well studied. | [AMS08, EAS98] |
| 1.1.7 | Currently being studied. | [HHT07] |
| 1.1.8 with $p = n$ | Developed to some extent. | [Yer02, WS97] |
| 1.1.8 with $p < n$ | Not well developed. | [TCA09] |
| 1.1.9 | Formulated in Riemannian optimization. However, algorithms to solve it had not been developed enough before [SI13]. | [HM94] |
| 1.2.1 (in Sec. 1.2.2) | Suggested to be studied in Riemannian optimization. However, it had not been studied before [SI13]. | [AMS08, SI13] |

### 1.1.3 Optimization algorithms on the Euclidean space and on Riemannian manifolds

In the Euclidean space, the line search strategies are often used to construct iterative optimization algorithms such as the steepest descent, Newton's, and the conjugate gradient methods. The steepest descent method has the global convergence property, but the speed of convergence is practically very slow. In contrast with this, a sequence generated by Newton's method, if it converges, exhibits a very fast convergence speed. However, Newton's method does not have a global convergence property. On the other hand, the conjugate gradient method has the global convergence property and generate sequences which converge faster than those generated by the steepest descent method (but slower than those generated by Newton's method). The speed of convergence of a sequence generated by the conjugate gradient method is practically fast.

These methods are generalized to those on Riemannian manifolds. The steepest descent and Newton's methods on Riemannian manifolds have been developed and proved to have the same convergence properties as those on the Euclidean space.

To the contrary, for the conjugate gradient method, generalized algorithms on Riemannian manifolds have been developing. In particular, an algorithm which is of practical use and which has a global convergence property without any special assumptions had not been proposed before the author's paper [SIarb]. We shall discuss a proposed algorithm at full length in this thesis.

In the next section, the problems to be tackled in this thesis are introduced in detail.

## 1.2 Overview of the topics in the thesis

The interest of this thesis centers on Riemannian optimization algorithms with applications to numerical linear algebra. The theory of Riemannian optimization has been developed to some extent, but still stays in the stage of development. For example, the conjugate gradient method on the Euclidean space has been generalized to that on Riemannian manifolds. However, the resulting method does not have a global convergence property in general. One of main purposes of this thesis is to improve the existing Riemannian conjugate gradient method and to prove the global convergence property of the proposed method.

The theory should be practically applied. As is discussed in Section 1.1, the theory of Riemannian optimization has indeed a great potential to solve practical problems. Another purpose of the thesis is to develop methods for solving problems in numerical linear algebra on the basis of Riemannian optimization. In particular, the singular value decompositions of real and complex matrices are formulated as Riemannian optimization problems and solved to provide efficient algorithms.

### 1.2.1 Riemannian conjugate gradient method

It is an empirical fact that the convergence speed of the steepest descent method is often very slow (see Subsection 4.4.1, for example). On the other hand, Newton's method generally generates quickly convergent sequences, but its convergence property is local. Among unconstrained optimization methods on the Euclidean space, the conjugate gradient method is known to generate sequences with global convergence property and the generated sequences are shown to converge faster than those generated by the steepest descent method. An extension of the conjugate gradient method to that on a Riemannian manifold is expected to solve Riemannian optimization problems efficiently.

The conjugate gradient method on a Riemannian manifold was proposed by Smith in his paper entitled "Optimization techniques on Riemannian manifolds" [Smi94]. In the conjugate gradient method in [Smi94], he proposed to use the parallel translation of tangent vectors along a geodesic on the manifold in question. However, the parallel translation of a tangent vector cannot be computed even for the Stiefel manifold $\mathrm{St}(p, n)$ in general. In [AMS08], the notion of a vector transport is introduced instead of the parallel translation in order to resolve the computational difficulties. The global convergence property of the Riemannian conjugate gradient method with the vector transport was discussed by Ring and Wirth in [RW12]. In [RW12], they proved that a sequence generated by the algorithm is globally convergent under the assumption that the vector transport in use does not increase the norm of the search direction vector. In fact, a sequence generated by the algorithm may not converge, if the assumption is not satisfied. Some numerical experiments will be performed in Chapter 3 to illustrate the situation where generated sequences are not convergent.

In order to improve the global convergence property of the Riemannian conjugate gradient method, the notion of a scaled vector transport will be introduced in Chapter 3. In the new proposed algorithm, the scaled vector transport is applied only if the vector transport increases the norm of the previous search direction vector. The proposed Riemannian conjugate gradient method with a scaled vector transport is shown to have the global convergence property even if the vector transport in use increases the norm of tangent vectors, since the scaling cancels the effect of the increase of the norm.

### 1.2.2 Singular value decomposition of a real matrix and the corresponding optimization problem on a real product manifold

In [AMS08], it is suggested that the truncated singular value decomposition problem can be set up as a Riemannian optimization problem on $\mathrm{St}(p, m) \times \mathrm{St}(p, n)$ as follows:

**Problem 1.2.1.**

$$\text{maximize} \quad \mathrm{tr}(U^T A V N), \tag{1.2.1}$$

$$\text{subject to} \quad (U, V) \in \mathrm{St}(p, m) \times \mathrm{St}(p, n), \tag{1.2.2}$$

where $N$ is defined to be $N = \mathrm{diag}(\mu_1, \ldots, \mu_p)$ with $\mu_1 > \cdots > \mu_p > 0$. We here note that the order of the values of the diagonal elements of the matrix $N$ differs from $N$ in Problem 1.1.5. In Chapter 4, we will prove that solving Problem 1.2.1 is indeed equivalent to performing the truncated singular value decomposition.

To this end, the geometry of the manifold $\mathrm{St}(p, m) \times \mathrm{St}(p, n)$ is intensively studied to acquire requisites such as the gradient and the Hessian of the objective function of the problem in Chapter 4. Then, a hybrid algorithm which consists of the conjugate gradient and Newton's methods is proposed. In other words, the conjugate gradient method is used in the first part of the algorithm to obtain an approximate optimal solution to the problem, and then Newton's method is applied with the approximate solution as an initial point, which forms the second part of the algorithm. Switching from the conjugate gradient method to Newton's method makes the convergence speed much faster than keeping the conjugate gradient method running, since Newton's method ensures that the generated sequence converges quadratically. If we use the scaled conjugate gradient method which is proposed in Chapter 3, it is ensured that the conjugate gradient part of the hybrid algorithm resolves the difficulty that Newton's method does not generally have a global convergence property.

### 1.2.3 Complex singular value decomposition and Riemannian optimization

The singular value decomposition of an $m \times n$ complex manifold takes the form

$$A = U\Sigma V^H, \qquad U \in U(m),\ V \in U(n),\ \Sigma = \begin{pmatrix} \Sigma_1 \\ \\ 0 \end{pmatrix}, \tag{1.2.3}$$

where the superscript $H$ denotes the Hermitian conjugation of a matrix, $U(n)$ is the unitary group, and where $\Sigma_1 = \mathrm{diag}\,(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$.

The complex singular value decomposition is expected to be reformulated as a Riemannian optimization problem on the product manifold $\mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$, where $\mathrm{St}(p, n, \mathbb{C})$ is the complex Stiefel manifold defined by $\mathrm{St}(p, n, \mathbb{C}) := \left\{ Y \in \mathbb{C}^{n \times p} \,|\, Y^H Y = I_p \right\}$. Since we will deal with the complex Stiefel manifold $\mathrm{St}(p, n, \mathbb{C})$ only in Chapter 5, we simply denote the real Stiefel manifold by $\mathrm{St}(p, n)$ in the other chapters. While the real singular value decomposition problem is equivalent to Problem 1.2.1, the function of the form (1.2.1) with $U \in \mathrm{St}(p, m, \mathbb{C})$ and $V \in \mathrm{St}(p, n, \mathbb{C})$ cannot be an objective function of the optimization problem on $\mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$. This is because the function (1.2.1) is no longer real-valued if $U$ and $V$ are complex matrices. Therefore, it is not straightforward

to establish an optimization problem on $\mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$. In fact, the complex singular value decomposition problem can be reformulated as the following problem:

**Problem 1.2.2.**

$$\begin{aligned} \text{maximize} \quad & \mathrm{Re}(\mathrm{tr}(U^H A V N)), && (1.2.4) \\ \text{subject to} \quad & (U, V) \in \mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C}), && (1.2.5) \end{aligned}$$

where $\mathrm{Re}(\cdot)$ denotes the real part of the quantity in the parentheses. It will be shown in Chapter 5 that the truncated complex singular value decomposition is indeed equivalent to Problem 1.2.2.

In order to put Problem 1.2.2 into a real one, the real manifold $\mathrm{Stp}(p, n)$, which is the intersection of the real Stiefel manifold and the "quasi-symplectic set", is introduced. This can be regarded as an expression of $\mathrm{St}(p, n, \mathbb{C})$ in a real form (see (5.2.33)). Problem 1.2.2 is reformulated as a real optimization problem on $\mathrm{Stp}(p, m) \times \mathrm{Stp}(p, n)$, for which Newton's equation can be derived in a similar manner to the case of Problem 1.2.1. The resulting Newton's method for Problem 1.2.1 is in turn rewritten as an algorithm for Problem 1.2.2, which results in the complex singular value decomposition.

## 1.3 Outline of the thesis

In this thesis, Riemannian optimization is studied from both the theoretical and application viewpoints. Theory and application are mutually-supportive and neither can be lacking as a whole. In this thesis, the Riemannian conjugate gradient method for a problem on a general Riemannian manifold will be theoretically improved to have a global convergence property. This improvement in turn ensures that the conjugate gradient method for the singular value decomposition related problem (Problem 1.2.1) on the product manifold $\mathrm{St}(p, m) \times \mathrm{St}(p, n)$ works well. Consequently, a more efficient algorithm is naturally proposed, which will be generalized to that for the complex case.

The organization of this thesis is as follows:

Chapter 2 starts with a brief exposition of unconstrained optimization techniques in the Euclidean space and proceeds to a generalization of the methods to those on a Riemannian manifold. The last part of this chapter is concerned with a review of the geometry of the real Stiefel manifold, which is necessary to set up the singular value decomposition as a Riemannian optimization problem.

Chapter 3 provides a new Riemannian conjugate gradient method through the introduction of the notion of a scaled vector transport. The strong Wolfe step condition, which is used to find the step size at each iterate, is also discussed on a general Riemannian manifold. A new algorithm with a scaled vector transport will be proposed. The global convergence property of the proposed algorithm is proved theoretically and numerical experiments show the efficiency of the algorithm. In addition, numerical experiments also

show that there are problems which the existing algorithm cannot solve efficiently but the proposed algorithm can do. This chapter is based on [SIarb].

Chapters 4 and 5 are devoted to the (truncated) real and complex singular value decomposition algorithms based on Riemannian optimization, respectively. They are based on [SI13] and [SIara].

In Chapter 4, the singular value decomposition of an $m \times n$ real matrix is dealt with. The decomposition problem is reformulated as a Riemannian optimization problem on the product manifold $\mathrm{St}(p, m) \times \mathrm{St}(p, n)$. The steepest descent, Newton's, and the conjugate gradient methods for the present optimization problem are developed, and the advantage and disadvantage of each algorithm are discussed. Furthermore, the conjugate gradient and Newton's methods are put together to give a hybrid algorithm. However, Newton's equation for the problem is practically difficult to solve unless $p = 1$. Therefore, an algorithm for solving Newton's equation with $p = 1$ is provided and the original problem is divided into $p$ subproblems in Newton's part of the proposed hybrid algorithm. If a sufficiently approximate solution to the problem is available in advance, the conjugate gradient part can be skipped. Numerical experiments are performed to show that Newton's method can improve approximate solutions, which are obtained by MATLAB's svd function for example. At the end of the chapter, degenerate optimal solutions, which appear if the target matrix $A$ has degenerate singular values, are also studied.

In Chapter 5, Newton's method developed in Chapter 4 is extended to the complex case. It is shown that the complex singular value decomposition problem is translated into a Riemannian optimization problem on the product of two complex Stiefel manifolds. For feasibility purpose, the problem is equivalently rewritten as a problem on the product of two real manifolds. The Riemannian geometry of the real product manifold in question is investigated after Chapter 4. Then, Newton's method on the real product manifold is developed and is converted to that on the complex product manifold. Moreover, like the algorithm given in Chapter 4, the proposed algorithm divides into easier subproblems, which can be solved in parallel. The resulting algorithm provides a new complex singular value decomposition algorithm. In a similar manner to that in Chapter 4, numerical experiments are also performed to show that the present algorithm may improve the accuracy of an approximate complex singular value decomposition.

Chapter 6 contains conclusions and discussions on the results in comparison with the existing algorithms with constraints taken into account.

# Chapter 2

# A Brief Review of Optimization Methods and Basic Facts on the Real Stiefel Manifold

In this chapter, we make a brief review of several optimization methods both on the Euclidean space and on Riemannian manifolds. As is explained in Introduction, the truncated singular value decompositions of a real matrix is reformulated as a Riemannian optimization problem on the product of the real Stiefel manifolds. In view of this fact, the geometry of the real Stiefel manifold is reviewed within the scope of Riemannian optimization.

## 2.1 A Review of unconstrained optimization methods on the Euclidean space

### 2.1.1 General framework

We consider the following general unconstrained optimization problem on $\mathbb{R}^n$:

**Problem 2.1.1.**

$$\text{minimize} \quad f(x), \tag{2.1.1}$$
$$\text{subject to} \quad x \in \mathbb{R}^n, \tag{2.1.2}$$

There have been developed several optimization methods such as the steepest descent, Newton's, and the conjugate gradient methods. They have the following common framework called the line search strategy.

**Algorithm 2.1.1** The line search strategy on the Euclidean space $\mathbb{R}^n$

1: Choose an initial point $x_0 \in \mathbb{R}^n$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Compute the search direction $\eta_k \in \mathbb{R}^n$ and the step size $t_k > 0$.
4: Compute the next iterate $x_{k+1} = x_k + t_k \eta_k$.
5: **end for**

The choice of the search direction $\eta_k \in \mathbb{R}^n$ and the step size $t_k > 0$ in Step 3 of Algorithm 2.1.1 characterizes the individual optimization methods.

As will be discussed in the following subsections, in each of the above-mentioned three methods, the search direction $\eta_k$ is distinctively computed by using the Euclidean gradient $f_x(x_k)$ of $f$ at $x_k$. The angle $\theta_k$ between the search direction $\eta_k$ and the negative gradient $-f_x(x_k)$ is defined through

$$\cos\theta_k = -\frac{f_x(x_k)^T \eta_k}{\|f_x(x_k)\|\|\eta_k\|}, \tag{2.1.3}$$

where $\|\cdot\|$ denotes the standard Euclidean norm. The search direction $\eta_k$ is called a descent direction if it satisfies $f_x(x_k)^T \eta_k < 0$. By the definition (2.1.3) of the angle $\theta_k$, a descent direction makes an angle of less than $\pi/2$ with $-f_x(x_k)$ and produces a decrease in $f$. More generally, the direction sequence $\{\eta_k\}$ is called gradient-related to the sequence $\{x_k\}$ if the following property holds: For any subsequence $\{x_k\}_{k\in\mathcal{K}}$ that converges to a non-stationary point of $f$, the corresponding subsequence $\{\eta_k\}_{k\in\mathcal{K}}$ is bounded and satisfies

$$\limsup_{k\to\infty, k\in\mathcal{K}} f_x(x_k)^T \eta_k < 0. \tag{2.1.4}$$

It is clear that a bounded sequence of descent directions is gradient-related. Several convergence results on sequences in $\mathbb{R}^n$ have been developed for optimization methods which generate gradient-related direction sequences. While the gradient and the conjugate gradient methods generate descent directions, Newton's method does not even generate a gradient-related direction sequence in general. In this thesis, we assume that a search direction vector is always a descent one when we consider line search methods. This means that when we apply Newton's method we take an initial point in a vicinity of a minimum point of the objective function. For more details of a gradient-related direction sequence, see [Ber99] (for Euclidean version) and [AMS08] (for Riemannian version).

There are also several choices of computing the step size $t_k > 0$. The step size is often computed so as to satisfy the Armijo or Wolfe condition with descent search directions. What to do in the line search methods at the $k$-th iterate $x_k$ is to search for the step size $t_k > 0$ on the half-line $x_k + t\eta_k,\ t > 0$, emanating from $x_k$ in the direction of $\eta_k$, in such a manner that the value $f(x_{k+1})$ of the objective function $f$ at the next iterate

$$x_{k+1} = x_k + t_k \eta_k \tag{2.1.5}$$

may sufficiently decrease. The steepest descent and the conjugate gradient methods produce a descent direction in every iteration if the step size is successfully computed. To the contrary, Newton's method does not necessarily produce a descent direction. In Newton's method, it is natural to fix the step size to $t_k = 1$. Newton's method with $t_k = 1$ is referred to as the pure form of Newton's method. In this thesis, we deal with the pure form of Newton's method and we call it simply Newton's method.

Apart from the line search strategy, the trust region strategy is also used, though we do not deal with it in this thesis. See [NW06] for more detail on the trust region strategy.

Some optimization methods have the global convergence property, that is, every accumulation point of a sequence with any initial point is a stationary point. Other methods such as Newton's method do not have the global convergence property. However, if a sequence generated by Newton's method converges, the speed of convergence is fast. The rate of convergence is defined as follows [Kel99, LY08]:

**Definition 2.1.1.** *Let $\{x_k\}$ be a sequence converging to a point $x_*$ on $\mathbb{R}^n$. The sequence $\{x_k\}$ is said to converge linearly to $x_*$ if there exist a constant $c \in (0, 1)$ and an integer $K \geq 0$ such that*

$$\|x_{k+1} - x_*\| \leq c\|x_k - x_*\|, \qquad k \geq K. \tag{2.1.6}$$

*The sequence $\{x_k\}$ is said to converge superlinearly to $x_*$ if*

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|} = 0. \tag{2.1.7}$$

*The sequence $\{x_k\}$ is said to converge to $x_*$ with order at least $q$, if there exist a real number $q > 1$, a constant $c > 0$, and an integer $K \geq 0$ such that*

$$\|x_{k+1} - x_*\| \leq c\|x_k - x_*\|^q, \qquad k \geq K. \tag{2.1.8}$$

*If $q = 2$ in (2.1.8), the $\{x_k\}$ is said to converge quadratically to $x_*$.*

### 2.1.2 Exact line search and the Armijo and the Wolfe conditions

Before proceeding to specific optimization methods, we review several line search strategies. How to choose a step size is critical for the convergence property of an optimization algorithm. We here suppose that the current iterate $x_k$ and the search direction $\xi_k$ are already obtained and we fix them throughout this subsection. We also assume that the $\xi_k$ is a descent direction. The line search at the iterate $x_k$ is to solve the following problem exactly or inexactly:

**Problem 2.1.2.**

$$\text{minimize} \quad f(x_k + t\eta_k), \tag{2.1.9}$$

$$\text{subject to} \quad t \in \mathbb{R},\ t > 0. \tag{2.1.10}$$

In the exact line search, the step size $t_k > 0$ is determined to be

$$t_k = \arg\min_{t>0} f(x_k + t\eta_k). \tag{2.1.11}$$

For some particular problems in which the objective function takes a simple form such as a quadratic one, the right-hand side of (2.1.11) can be written out explicitly, so that the exact line search can be easily performed. However, for a general objective function, the exact line search is difficult to apply, and we end up with solving Problem 2.1.2 approximately in general.

In inexact line search strategies, we need to find a step size $t_k > 0$ satisfying some reasonable conditions at the iterate $x_k$. We here review two of well-known criteria for $t_k > 0$. One is the Armijo condition given by

$$f(x_k + t_k\eta_k) \leq f(x_k) + c_1 t_k f_x(x_k)^T \eta_k \tag{2.1.12}$$

for some predetermined constant $c_1 \in (0, 1)$. The Armijo condition ensures that the determined step size gives rise to a sufficient decrease in the objective function. Especially, for $\bar{\alpha} > 0,\ \beta,\ \sigma \in (0, 1)$, the step size $t_k := \beta^m \bar{\alpha}$ to be defined with $m$ being the smallest nonnegative integer satisfying

$$f(x_k) - f(x_k + \beta^m \bar{\alpha}\eta_k) \geq -\sigma f_x(x_k)^T \beta^m \bar{\alpha}\eta_k \tag{2.1.13}$$

meets the Armijo condition. Such a $t_k$ is called the Armijo step size. The Armijo condition (2.1.12) is satisfied for a sufficiently small $t_k > 0$. However, if $t_k$ is too small, then $x_{k+1}$ is too close to $x_k$, and this may cause slow convergence.

Another approach to the line search is to use the Wolfe condition given by

$$f(x_k + t\eta_k) \leq f(x_k) + c_1 t_k f_x(x_k)^T \eta_k, \tag{2.1.14}$$

$$f_x(x_k + t\eta_k)^T \eta_k \geq c_2 f_x(x_k)^T \eta_k \tag{2.1.15}$$

for predetermined constant $c_1$ and $c_2$ with $0 < c_1 < c_2 < 1$. The Wolfe condition is a combination of the Armijo condition (2.1.12) and the curvature condition (2.1.15), which rules out excessively short steps. The Wolfe condition plays an important role in the conjugate gradient method. This is because Zoutendijk's theorem 2.1.1 [Zou70, NW06] to be stated in the below guarantees that algorithms with the Wolfe condition have a certain property which leads to the global convergence property of the conjugate gradient method. We introduce an assumption before stating Zoutendijk's theorem.

**Assumption 2.1.1.** *Let $f$ be bounded below on $\mathbb{R}^n$ and continuously differentiable in an open set $\mathcal{N}$ containing the sublevel set $\mathcal{L} := \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$, where $x_0 \in \mathbb{R}^n$ is the*

*initial point of the algorithm in question. Furthermore, the gradient $f_x$ of $f$ is Lipschitz continuous on $\mathcal{N}$, that is, there exists a Lipschitz constant $L > 0$ such that*

$$\|f_x(x) - f_x(y)\| \leq L\|x - y\|, \qquad x, y \in \mathcal{N}. \tag{2.1.16}$$

**Theorem 2.1.1.** *Consider Problem 2.1.1 with the objective function $f$ satisfying Assumption 2.1.1. Let $\{x_k\}$ be a sequence generated by Algorithm 2.1.1 with the Wolfe condition. If every search direction $\eta_k$ is a descent one, then it holds that*

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \|f_x(x_k)\|^2 < \infty, \tag{2.1.17}$$

*where the angles $\theta_k$ are defined by* (2.1.3)*.*

In the conjugate gradient method, the strong Wolfe condition, which is a strict version of the Wolfe condition, is often used. The strong Wolfe condition consists of two inequalities

$$f(x_k + t\eta_k) \leq f(x_k) + c_1 t_k f_x(x_k)^T \eta_k, \tag{2.1.18}$$
$$|f_x(x_k + t\eta_k)^T \eta_k| \leq c_2 |f_x(x_k)^T \eta_k|, \tag{2.1.19}$$

that is, the curvature condition (2.1.15) in the Wolfe condition is replaced by a more strict condition (2.1.19). We here note that $f_x(x_k)^T \eta_k < 0$ on account of the assumption that $\eta_k$ is a descent direction.

### 2.1.3 Steepest descent method

In the steepest descent method, the search direction $\eta_k$ at the iterate $x_k$ is determined by

$$\eta_k = -f_x(x_k), \tag{2.1.20}$$

which is the negative gradient of $f$ at $x_k$. The negative gradient $-f_x(x_k)$ is the steepest descent direction at $x_k$ in the sense that the unit vector $-f_x(x_k)/\|f_x(x_k)\|$ is the solution to the following problem:

**Problem 2.1.3.**

$$\text{minimize} \quad f_x(x_k)^T \eta \tag{2.1.21}$$
$$\text{subject to} \quad \|\eta\| = 1. \tag{2.1.22}$$

We here note that $f_x(x_k)^T \eta$ is the Fréchet derivative $\mathrm{D}f(x_k)[\eta]$ of $f$ at $x_k$ in the direction of $\eta$.

If $f$ satisfies Assumption 2.1.1, it is clear that the inequality (2.1.17) in Zoutendijk's theorem 2.1.1 with $\cos\theta_k = 1$ ensures that $\lim_{k\to\infty} \|f_x(x_k)\| = 0$ for a sequence $\{x_k\}$

generated by the steepest descent method with the Wolfe condition. However, the steepest descent method is shown to have the global convergence property, if only the Armijo condition without the curvature condition (2.1.15) is imposed in the algorithm, as is stated in the following. The steepest descent method with the Armijo condition is written in the form:

**Algorithm 2.1.2** Euclidean steepest descent method for Problem 2.1.1

1: Choose an initial point $x_0 \in \mathbb{R}^n$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Compute the search direction $\eta_k = -f_x(x_k)$ and the Armijo step size $t_k > 0$.
4: Compute the next iterate $x_{k+1} = x_k + t_k \eta_k$.
5: **end for**

**Theorem 2.1.2.** [Ber99] *Let $\{x_k\}$ be a sequence generated by Algorithm 2.1.2. Then, every accumulation point of $\{x_k\}$ is a stationary point.*

### 2.1.4 Newton's method

Newton's method is originally a method to solve the equation

$$g(x) = 0, \tag{2.1.23}$$

where the function $g : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable. Assume that the Jacobian matrix $J(x_k)$ of $g$ at the current iterate $x_k$ is invertible. The updating formula is then

$$x_{k+1} = x_k - J(x_k)^{-1} g(x_k). \tag{2.1.24}$$

From an optimization viewpoint, we need to find a point $x_*$ at which the gradient $f_x$ of the objective function $f$ vanishes. Then, Newton's method in optimization is applied to $g(x) = f_x(x)$, and the search direction $\eta_k \in \mathbb{R}^n$ is determined by Newton's equation

$$f_{xx}(x_k)[\eta_k] = -f_x(x_k), \tag{2.1.25}$$

where $f_{xx}(x_k)$ is the Hessian matrix of $f$ at the current iterate $x_k$. If the Hessian matrix $f_{xx}(x_k)$, hence the inverse $(f_{xx}(x_k))^{-1}$, is positive definite, the resulting Newton vector $\eta_k = -(f_{xx}(x_k))^{-1} f_x(x_k)$ is a descent direction since

$$f_x(x_k)^T \eta_k = -f_x(x_k)^T (f_{xx}(x_k))^{-1} f_x(x_k) < 0. \tag{2.1.26}$$

Then, the line search method discussed in Subsection 2.1.2 can be effectively combined with Newton's method. However, the Newton vector $\eta_k$ is not guaranteed to be a descent direction in general. In Newton's method for a generic problem, the step size is often fixed to $t_k = 1$. The resulting algorithm is stated as follows:

**Algorithm 2.1.3** Euclidean Newton's method for Problem 2.1.1

1: Choose an initial point $x_0 \in \mathbb{R}^n$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Compute the search direction $\eta_k$ as the solution to

$$f_{xx}(x_k)[\eta_k] = -f_x(x_k). \tag{2.1.27}$$

4: Compute the next iterate $x_{k+1} = x_k + \eta_k$.
5: **end for**

Newton's method does not generally have the global convergence property. However, a merit of Newton's method lies in its fast local convergence speed, as is shown in the following:

**Theorem 2.1.3.** [NW06] *Suppose that $f$ is twice differentiable and that the Hessian $f_{xx}$ is Lipschitz continuous in a neighborhood of a stationary point $x_*$. Suppose also that the Hessian $f_{xx}$ is Lipschitz continuous in a neighborhood of a solution $x_*$ at which the $f_{xx}(x_*)$ is positive definite. Then, any sequence $\{x_k\}$ generated by Algorithm 2.1.3 converges quadratically to $x_*$ if the initial point $x_0$ is sufficiently close to $x_*$.*

### 2.1.5 Conjugate gradient method

We again note that one of the main topics of this thesis is to propose a new, globally convergent conjugate gradient method on a Riemannian manifold. In this subsection, we carefully review the conjugate gradient method on the Euclidean space as a preparation.

The conjugate gradient method on $\mathbb{R}^n$ is originally developed as a tool for solving linear systems of equations [HS52], which minimizes the quadratic objective function $\phi(x) := x^T Ax/2 - b^T x$ of $x \in \mathbb{R}^n$, where $A$ and $b$ are an $n \times n$ symmetric positive-definite matrix and an $n$-dimensional column vector, respectively. The conjugate gradient method for this purpose is especially called the linear conjugate gradient method. Since the objective function $x^T Ax/2 - b^T x$ can be rewritten as

$$\frac{1}{2}x^T Ax - b^T x = \frac{1}{2}(Ax - b)^T A^{-1}(Ax - b) - \frac{1}{2}b^T A^{-1} b, \tag{2.1.28}$$

it can be easily observed that the minimum point of the objective function is a unique solution to $Ax = b$. In the linear conjugate gradient method, the initial search direction $\eta_0$ is chosen to be just the steepest descent direction $-\phi_x(x_0) = -(Ax_0 - b)$, and the search direction $\eta_k$ with $k \geq 1$ is computed from the steepest descent direction at $x_k$ and the previous search direction $\eta_{k-1}$ by

$$\eta_k = -\phi_x(x_k) + \beta_k \eta_{k-1} = -(Ax_k - b) + \beta_k \eta_{k-1}, \tag{2.1.29}$$

where the scalar $\beta_k$ is determined so that the current and previous search directions, $\eta_k$ and $\eta_{k-1}$, may be conjugate with respect to $A$, that is,

$$\eta_k^T A \eta_{k-1} = 0. \tag{2.1.30}$$

More specifically, the $\beta_k$ takes the form

$$\beta_k = \frac{(Ax_k - b)^T A \eta_{k-1}}{\eta_{k-1}^T A \eta_{k-1}}, \tag{2.1.31}$$

which is also obtained by applying the Gram-Schmidt orthogonalization procedure without normalization (in which the inner product of two vectors $a_1$ and $a_2$ are defined as $a_1^T A a_2$) to the vector $-\phi_x(x_k)$ and the preceding directions $\eta_0, \ldots, \eta_{k-1}$. In fact, the current direction $\eta_k$ satisfies

$$\eta_k^T A \eta_j = 0, \qquad j < k \leq n, \tag{2.1.32}$$

and the step size $t_k$ is calculated via the exact line search as

$$t_k = \arg\min_{t>0} \phi(x_k + t\eta_k) = -\frac{(Ax_k - b)^T \eta_k}{\eta_k^T A \eta_k}. \tag{2.1.33}$$

The $\beta_k$ defined by (2.1.31) can be put in the form

$$\beta_k = \frac{\phi_x(x_k)^T \phi_x(x_k)}{\phi_x(x_{k-1})^T \phi_x(x_{k-1})} = \frac{\phi_x(x_k)^T \left(\phi_x(x_k) - \phi_x(x_{k-1})\right)}{\phi_x(x_{k-1})^T \phi_x(x_{k-1})}, \tag{2.1.34}$$

where use has been made of the fact that $\phi_x(x_k)^T \phi_x(x_i) = 0$ with $i = 0, \ldots, k-1$, which can be easily proved by induction. The expression (2.1.34) of $\beta_k$ is a key to a generalization of the linear conjugate gradient method to a nonlinear conjugate gradient method.

The nonlinear conjugate gradient method can be applied for a generic objective function $f$ [NW06]. In the nonlinear conjugate gradient method, the search direction $\eta_k$ is computed after the manner of the linear conjugate method as

$$\eta_k = -f_x(x_k) + \beta_k \eta_{k-1}, \qquad k \geq 0, \tag{2.1.35}$$

where $\beta_0 = 0$, and where $\beta_k$ with $k \geq 1$ are determined in several possible manners. A possible choice for $\beta_k$ comes from (2.1.31),

$$\beta_k = \frac{f_x(x_k)^T f_{xx}(x_{k-1}) \eta_{k-1}}{\eta_{k-1}^T f_{xx}(x_{k-1}) \eta_{k-1}}. \tag{2.1.36}$$

Note that in the linear conjugate gradient method, we have $\phi_{xx} = A$. However, Eq. (2.1.36) for a general $f$ is impractical since the Hessian matrix $f_{xx}$ should be computed at each iterate. There are more practical choices of $\beta_k$ without reference to the Hessian matrix.

For example, $\beta_k$ are computed by

$$\beta_k^{\mathrm{FR}} = \frac{f_x(x_k)^T f_x(x_k)}{f_x(x_{k-1})^T f_x(x_{k-1})}, \tag{2.1.37}$$

or

$$\beta_k^{\mathrm{PR}} = \frac{f_x(x_k)^T \left(f_x(x_k) - f_x(x_{k-1})\right)}{f_x(x_{k-1})^T f_x(x_{k-1})}, \tag{2.1.38}$$

which are the generalizations of the right-hand sides of (2.1.34), where FR and PR are abbreviations of Fletcher-Reeves and Polak-Ribière, respectively [NW06]. Many other choices of $\beta_k$ have been also developed. The Fletcher-Reeves type conjugate gradient method on $\mathbb{R}^n$ is usually performed along with the strong Wolfe condition and is formulated to provide the algorithm:

---

**Algorithm 2.1.4** Euclidean Fletcher-Reeves type conjugate gradient method for Problem 2.1.1

---

1: Choose an initial point $x_0 \in \mathbb{R}^n$.
2: Set $\eta_0 = -f_x(x_0)$.
3: **for** $k = 0, 1, 2, \ldots$ **do**
4: Compute the step size $t_k > 0$ satisfying the strong Wolfe condition, consisting of (2.1.18) and (2.1.19) with $0 < c_1 < c_2 < 1/2$. Set

$$x_{k+1} = x_k + t_k \eta_k. \tag{2.1.39}$$

5: Set

$$\beta_{k+1} = \frac{\left(f_x(x_{k+1})\right)^T \left(f_x(x_{k+1})\right)}{\left(f_x(x_k)\right)^T \left(f_x(x_k)\right)}, \tag{2.1.40}$$

$$\eta_{k+1} = -f_x(x_{k+1}) + \beta_{k+1}\eta_k. \tag{2.1.41}$$

6: **end for**

---

The Fletcher-Reeves type conjugate gradient method has the global convergence property as the following theorem indicates:

**Theorem 2.1.4.** [AB85] *Suppose that the objective function $f$ satisfies Assumption 2.1.1. Then, the sequence $\{x_k\}$ generated by Algorithm 2.1.4 satisfies*

$$\liminf_{k\to\infty} \|f_x(x_k)\| = 0. \tag{2.1.42}$$

Theorem 2.1.4 is proved on the basis of Zoutendijk's theorem 2.1.1 and the following lemma.

**Lemma 2.1.1.** *Algorithm 2.1.4 generates descent directions $\eta_k$ which satisfy*

$$-\frac{1}{1-c_2} \leq \frac{f_x(x_k)^T \eta_k}{\|f_x(x_k)\|^2} \leq \frac{2c_2 - 1}{1 - c_2}, \qquad k = 0, 1, \ldots. \tag{2.1.43}$$

## 2.2 Riemannian optimization methods

As is discussed in Chapter 1, a number of optimization problems can be formulated on Riemannian manifolds. Let $M$ be a Riemannian manifold endowed with a Riemannian metric $\langle \cdot, \cdot \rangle$. An unconstrained optimization problem on $M$ is generally described as follows:

**Problem 2.2.1.**

$$\text{minimize} \quad f(x), \tag{2.2.1}$$
$$\text{subject to} \quad x \in M, \tag{2.2.2}$$

where $f$ is assumed to be smooth throughout this section unless otherwise noted.

When we generalize Euclidean optimization methods to Riemannian ones, several objects have to be generalized so as to make sense on Riemannian manifolds. For example, in the steepest descent method on $\mathbb{R}^n$, the search direction is determined as $-f_x(x_k)$. The Euclidean gradient $f_x(x_k)$ should be replaced by the gradient defined on $M$. Also, even if $M$ is a submanifold of $\mathbb{R}^n$ and $f_x(x_k)$ makes sense, the half line with the negative Euclidean gradient $-f_x(x_k)$ is not suitable for a search direction, since it does not generically lie in the submanifold $M$. With these matters in mind, we should take a search direction $\eta_k$ as a tangent vector on $M$ at $x_k$ and replace the search line by another concept to be defined on $M$.

### 2.2.1 Line search and retraction

If $M$ is the Euclidean space $\mathbb{R}^n$, the line search can be performed with the updating formula (2.1.5). However, Eq. (2.1.5) does not make sense on a general manifold $M$. Indeed, the operation of addition is not defined on $M$ in general. Even if $M$ is a submanifold of $\mathbb{R}^n$ and the addition can be defined, the resulting vector $x_k + t_k\eta_k$ is no longer sitting on $M$. In order to generalize the line search (2.1.5) on $\mathbb{R}^n$ to that on $M$, the addition in Eq. (2.1.5) should be replaced by another suitable operation. A natural alternative to the line search is a search along the geodesic emanating from $x_k$ in the direction of $\eta_k$, but the geodesic will cause computational difficulty except for a few particular manifolds where the geodesics admit a tractable closed-form expression. A computationally efficient way is to use the following retraction map introduced in [AMS08].

**Definition 2.2.1.** *Let $M$ and $TM$ be a manifold and the tangent bundle of $M$, respectively. Let $R : TM \to M$ be a smooth map and $R_x$ the restriction of $R$ to $T_xM$. The $R$ is called a retraction on $M$, if it has the following properties:*

1. *$R_x(0_x) = x$, where $0_x$ denotes the zero element of $T_xM$.*
2. *With the canonical identification $T_{0_x}T_xM \simeq T_xM$, $R_x$ satisfies*

$$\mathrm{D}R_x(0_x) = \mathrm{id}_{T_xM}, \tag{2.2.3}$$

*where* $\mathrm{D}R_x(0_x)$ *denotes the derivative of* $R_x$ *at* $0_x$*, and* $\mathrm{id}_{T_xM}$ *the identity map on* $T_xM$*.*

As is easily seen, the exponential map on $M$ is a typical example of a retraction. For a more detailed discussion about retractions, see [AM12].

Suppose that $x_k \in M$ and $\eta_k \in T_xM$ are the current iterate and the search direction, respectively, in an iterative optimization algorithm with a retraction $R$. Let $\gamma_k$ be a curve on $M$ defined by $\gamma_k(t) = R_{x_k}(t\eta_k)$. The first condition in Definition 2.2.1 means that $\gamma_k(0) = x_k$. The second condition implies that $\dot{\gamma}_k(0) = \eta_k$. Thus, the curve $\gamma_k$ proves to be emanating from $x_k$ in the direction of $\eta_k$. Therefore, at each iterate, the retraction gives rise to an appropriate curve on $M$ for searching the next iterate. In order to generalize the line search on $\mathbb{R}^n$ to an appropriate search on $M$, the line search (2.1.5) should be replaced by a search along the curve $\gamma_k$ so that for a suitable determined $t_k > 0$ the resulting next iterate

$$x_{k+1} = R_{x_k}(t_k\eta_k) \tag{2.2.4}$$

may produce sufficient decrease in $f$.

If we can find a computationally preferable retraction, we can perform a Riemannian optimization procedure as follows:

**Algorithm 2.2.1** The general framework of Riemannian optimization methods for Problem 2.2.1

1: Choose an initial point $x_0 \in M$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Compute the search direction $\eta_k \in T_{x_k}M$ and the step size $t_k > 0$.
4: Compute the next iterate by $x_{k+1} := R_{x_k}(t_k\eta_k)$, where $R$ is a retraction on $M$.
5: **end for**

As in the Euclidean optimization, the choice of a search direction and a step size depends on optimization methods.

We proceed to computing procedure for a step size. In what follows, we fix $x_k \in M$ and $\eta_k \in T_{x_k}M$ as a current iterate and a search direction, respectively. We also assume that $\eta_k$ is a descent direction, that is, $\langle \operatorname{grad} f(x_k), \eta_k \rangle_{x_k} < 0$. The step size in the exact search is determined by

$$t_k = \arg\min_{t>0} f(R_{x_k}(t\eta_k)). \tag{2.2.5}$$

Since it is difficult to find $t_k$ in general, inexact search strategies are of practical use.

The Armijo condition (2.1.12) on $\mathbb{R}^n$ is generalized to that on $M$ which is expressed as

$$f(R_{x_k}(t\eta_k)) \leq f(x_k) + c_1 t_k \langle \operatorname{grad} f(x_k), \eta_k \rangle_{x_k}, \tag{2.2.6}$$

where $c_1 \in (0, 1)$ is constant uniformly for all $k \geq 0$. From a numerical viewpoint, we perform a backtracking algorithm to find a step size satisfying the Armijo condition. That

is, for given parameters $\bar{\alpha} > 0,\ \beta,\ \sigma \in (0,1)$, the step size $t_k$ is determined by $t_k := \beta^m \bar{\alpha}$ in such a way that $m$ may be the smallest nonnegative integer satisfying

$$f(x_k) - f(R_{x_k}(\beta^m \bar{\alpha} \eta_k)) \geq -\sigma \langle \operatorname{grad} f(x_k), \beta^m \bar{\alpha} \eta_k \rangle_{x_k}. \tag{2.2.7}$$

The $t_k = \beta^m \bar{\alpha}$ thus determined is called the Armijo step size as in Euclidean optimization.

The Wolfe condition should be also generalized from $\mathbb{R}^n$ to $M$. In [RW12] and [SIarb], the Wolfe condition on $M$ is discussed, which is closely related to the notion of a vector transport (see Subsection 2.2.2) and crucial for the convergence property of the conjugate gradient method on $M$. We will discuss the (strong) Wolfe condition on $M$ in more detail in Subsection 2.2.5 and Section 3.2.

### 2.2.2 Vector transport

In the (nonlinear) conjugate gradient method on the Euclidean space $\mathbb{R}^n$, the search directions $\eta_k$ are computed by (2.1.35). However, on a Riemannian manifold $M$, $\operatorname{grad} f(x_k) \in T_{x_k} M$ and $\eta_{k-1} \in T_{x_{k-1}} M$ belong to different tangent spaces, so that $-\operatorname{grad} f(x_k) + \beta_k \eta_{k-1}$ in Eq. (2.1.35) fails to make sense on $M$. In order to modify the vector addition into a suitable operation on $M$, Smith proposed to use the parallel translation of tangent vectors along a geodesic [Smi94]. However, no computationally efficient formula is known for the parallel translation along a geodesic even for the Stiefel manifold $\mathrm{St}(p,n)$ except when it reduces to the sphere ($p = 1$) or the orthogonal group ($p = n$). Absil *et al.* [AMS08] proposed the notion of a vector transport as an alternative to the parallel translation as follows:

**Definition 2.2.2.** *A vector transport $\mathcal{T}$ on a manifold $M$ is a smooth map*

$$TM \oplus TM \to TM : (\eta_x, \xi_x) \mapsto \mathcal{T}_{\eta_x}(\xi_x) \in TM \tag{2.2.8}$$

*satisfying the following properties for all $x \in M$:*

1. *There exists a retraction $R$, called the retraction associated with $\mathcal{T}$, such that*

$$\pi\left(\mathcal{T}_{\eta_x}(\xi_x)\right) = R_x\left(\eta_x\right), \tag{2.2.9}$$

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

2. *$\mathcal{T}_{0_x}(\xi_x) = \xi_x$ for all $\xi_x \in T_x M$,*

3. *$\mathcal{T}_{\eta_x}(a\xi_x + b\zeta_x) = a\mathcal{T}_{\eta_x}(\xi_x) + b\mathcal{T}_{\eta_x}(\zeta_x)$.*

The vector transport is a generalization of the parallel translation and can enhance computational efficiency of algorithms, if defined suitably.

One of the most reasonable choices for vector transport is the differentiated retraction $\mathcal{T}^R$ defined by

$$\mathcal{T}^R_{\eta_x}(\xi_x) := \mathrm{D}R_x(\eta_x)[\xi_x], \qquad \eta_x, \xi_x \in T_xM, \tag{2.2.10}$$

which will be used to propose a new Riemannian conjugate gradient method in Chapter 3.

Another simple vector transport $\mathcal{T}^P$ is defined by means of the orthogonal projection $P$, if $M$ is an embedded submanifold of the Euclidean space $\mathbb{R}^n$ with an inner product $(\cdot, \cdot)$. That is, if a vector $y \in \mathbb{R}^n$ is decomposed into

$$y = y_T + y_N, \qquad y_T \in T_xM,\ y_N \in N_xM, \tag{2.2.11}$$

then $P_x(y) = y_T$, where the normal space $N_xM$ at $x$ is the orthogonal complement of $T_xM$ in $T_x\mathbb{R}^n \simeq \mathbb{R}^n$, which is defined by $N_xM := \{\eta \in \mathbb{R}^n \,|\, (\xi, \eta) = 0,\ \forall \xi \in T_xM\}$. The vector transport $\mathcal{T}^P$ is then defined by

$$\mathcal{T}^P_{\eta_x}(\xi_x) := P_{R_x(\eta_x)}(\xi_x). \tag{2.2.12}$$

All the three conditions of Definition 2.2.2 are easily verified for both $\mathcal{T} = \mathcal{T}^R$ and $\mathcal{T} = \mathcal{T}^P$ by using the definitions of the retraction and the orthogonal projection, respectively.

We here have to note that though the parallel translation is an isometry, a vector transport is not required to preserve the norm of vectors in general. It will be found later that the convergence property of the conjugate gradient method depends crucially on whether the vector transport increases the norm of vectors or not. In order to make a given vector transport $\mathcal{T}$ not to increase the norm of the transported vector, we will define the notion of a scaled vector transport in Subsection 3.2.1.

### 2.2.3 Steepest descent method

In the steepest descent method on a Riemannian manifold $M$, the search direction $\eta_k \in T_{x_k}M$ is determined as

$$\eta_k = -\operatorname{grad} f(x_k), \tag{2.2.13}$$

where $\operatorname{grad} f$ is the gradient of $f$ on $M$ with respect to the endowed metric $\langle \cdot, \cdot \rangle$, that is, $\operatorname{grad} f(x)$ is a unique tangent vector to $x \in M$ which satisfies

$$\langle \operatorname{grad} f(x), \xi \rangle_x = \mathrm{D}f(x)[\xi] \tag{2.2.14}$$

for any $\xi \in T_xM$. Since the $\eta_k$ given by (2.2.13) is in the steepest descent direction, the Armijo condition can be used. In this thesis, we treat the following steepest descent method:

**Algorithm 2.2.2** Riemannian steepest descent method for Problem 2.2.1

1: Choose an initial point $x_0 \in M$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Compute the search direction $\eta_k = -\operatorname{grad} f(x_k)$ and the Armijo step size $t_k > 0$.
4: Compute the next iterate by $x_{k+1} := R_{x_k}(t_k \eta_k)$, where $R$ is a retraction on $M$.
5: **end for**

According to [AMS08], a convergence property for the steepest descent method is stated as follows:

**Theorem 2.2.1.** *Let $\{x_k\}$ be a sequence of iterates generated by Algorithm 2.2.2. Then, every accumulation point of $\{x_k\}$ is a critical point of the objective function $f$.*

### 2.2.4 Newton's method

In Newton's method on the Riemannian manifold $M$, the search direction $\eta_k$ is determined as the solution of Newton's equation

$$\operatorname{Hess} f(x_k)[\eta_k] = -\operatorname{grad} f(x_k), \tag{2.2.15}$$

where the Hessian $\operatorname{Hess} f(x)$ of $f$ at $x$ is defined through the covariant derivative $\nabla_\eta \operatorname{grad} f$ with respect to the Levi-Civita connection $\nabla$ on $M$ by

$$\operatorname{Hess} f(x)[\eta] := \nabla_\eta \operatorname{grad} f. \tag{2.2.16}$$

In Newton's method, search directions are not necessarily descent ones. Thus, we fix $t_k := 1$ for any $k$, without performing the line search. The resulting algorithm is as follows:

**Algorithm 2.2.3** Riemannian Newton's method for Problem 2.2.1

1: Choose an initial point $x_0 \in M$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Compute the search direction $\eta_k$ as a solution to Newton's equation

$$\operatorname{Hess} f(x_k)[\eta_k] = -\operatorname{grad} f(x_k). \tag{2.2.17}$$

4: Compute the next iterate by $x_{k+1} := R_{x_k}(\eta_k)$, where $R$ is a retraction on $M$.
5: **end for**

According to [ABM08], the convergence property of Newton's method is stated as follows:

**Theorem 2.2.2.** *Let $x_c \in M$ be a critical point of $f$; $\operatorname{grad} f(x_c) = 0$. Assume that $\operatorname{Hess} f(x_c)$ is non-degenerate at $x_c \in M$. Then there exists a neighborhood $U$ of $x_c$ in*

*$M$ such that for all $x_0 \in U$ the sequence $\{x_k\}$ generated by Algorithm 2.2.3 converges quadratically to $x_c$.*

We note that Riemannian Newton's method does not have a global convergence property.

### 2.2.5 Conjugate gradient method

In order to generalize the nonlinear conjugate gradient method on $\mathbb{R}^n$ to that on $M$, it is not adequate to determine the search direction $\eta_k \in T_{x_k}M$ as

$$\eta_k = -\operatorname{grad} f(x_k) + \beta_k \eta_{k-1}, \tag{2.2.18}$$

if the $\operatorname{grad} f$ is the gradient of $f$ on $M$ with respect to the endowed Riemannian metric, and if $\beta_k$ is defined to be

$$\beta_k = \frac{\|\operatorname{grad} f(x_k)\|_{x_k}^2}{\|\operatorname{grad} f(x_{k-1})\|_{x_{k-1}}^2} \tag{2.2.19}$$

in correspondence to Euclidean Fletcher Reeves $\beta^{\mathrm{FR}}$ in (2.1.37), for example, where $\|\cdot\|_x$ denotes the norm of a tangent vector to $x$ with respect to the metric $\langle\cdot,\cdot\rangle$. Actually, the right-hand side of (2.2.18) makes no sense since $\operatorname{grad} f(x_k) \in T_{x_k}M$ and $\eta_{k-1} \in T_{x_{k-1}}M$ are in different tangent spaces.

In [Smi94], Smith proposed to use the parallel translation along a geodesic in order to transport the second term in (2.2.18) from $T_{x_{k-1}}M$ into $T_{x_k}M$. However, computing the parallel translation is often difficult. Alternatively, in [AMS08], Absil *et al.* introduced the notion of vector transport (see Subsection 2.2.2). The resulting algorithm is described as follows:

**Algorithm 2.2.4** Riemannian conjugate gradient method for Problem 2.2.1

1: Choose an initial point $x_0 \in M$.
2: Set $\eta_0 = -\operatorname{grad} f(x_0)$.
3: **for** $k = 0, 1, 2, \ldots$ **do**
4: Compute the step size $t_k > 0$. Set

$$x_{k+1} = R_{x_k}(t_k \eta_k), \tag{2.2.20}$$

where $R$ is a retraction on $M$.
5: Compute the $\beta_{k+1}$ and set

$$\eta_{k+1} = -\operatorname{grad} f(x_{k+1}) + \beta_{k+1} \mathcal{T}_{t_k \eta_k}(\eta_k), \tag{2.2.21}$$

where $\mathcal{T}$ is a vector transport.
6: **end for**

In [RW12], Ring and Wirth assumed that the vector transport $\mathcal{T}^R$ as the differentiated

retraction does not increase the norm of search directions, that is,

$$\|\mathcal{T}^R_{R_{x_k}(t_k\eta_k)}(\eta_k)\|_{x_{k+1}} \leq \|\eta_k\|_{x_k} \tag{2.2.22}$$

for all $k \in \mathbb{N}$. Under this assumption, they proved the convergence of the Fletcher-Reeves type of the above algorithm with a vector transport $\mathcal{T}^R$ and the strong Wolfe step condition,

$$f\left(R_{x_k}(t_k\eta_k)\right) \leq f(x_k) + c_1 t_k \mathrm{D}f(x_k)[\eta_k], \tag{2.2.23}$$

$$|\mathrm{D}f\left(R_{x_k}(t_k\eta_k)\right)[\mathcal{T}^R_{R_{x_k}(t_k\eta_k)}(\eta_k)]| \leq -c_2 \mathrm{D}f(x_k)[\eta_k]. \tag{2.2.24}$$

Their theorem is stated as follows:

**Theorem 2.2.3.** *Consider Algorithm 2.2.4 with the vector transport $\mathcal{T}^R$ defined by* (2.2.10) *and the strong Wolfe condition consisting of* (2.2.23) *and* (2.2.24)*. The coefficient $\beta_k$ is computed in the form* (2.2.19) *of the Fletcher-Reeves type. If the condition* (2.2.22) *holds for all $k \in \mathbb{N}$, then*

$$\liminf_{k\to\infty} \|\mathrm{grad}\, f(x_k)\|_{x_k} = 0. \tag{2.2.25}$$

However, the condition (2.2.22) does not always hold. Such an example will be shown in Section 3.5. Hence, the algorithm in Thm. 2.2.3 does not generally have a global convergence property. In order to resolve this difficulty, we will introduce the notion of a scaled vector transport and propose a new, globally convergent algorithm in Chapter 3.

## 2.3 Riemannian geometry of the real Stiefel manifold

Let $n$ and $p$ be positive integers with $n \geq p$. Let $\mathrm{St}(p,n)$ denote the set of all $n \times p$ orthonormal matrices, that is,

$$\mathrm{St}(p,n) = \left\{Y \in \mathbb{R}^{n\times p} \,|\, Y^T Y = I_p\right\}. \tag{2.3.1}$$

The set $\mathrm{St}(p,n)$ can be endowed with a natural manifold structure and then is called the Stiefel manifold [AMS08, EAS98, HM94]. There are many practical optimization problems defined on the Stiefel manifold such as Problems 1.1.5 and 1.1.8. In this thesis, the singular value decomposition is formulated as an optimization problem on the product manifold $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$. The geometry of the product manifold $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ will be treated in Chapter 4, where full use will be made of the geometry of a single Stiefel manifold $\mathrm{St}(p,n)$. In this section, we make a review of several geometrical objects on $\mathrm{St}(p,n)$ which are necessary for optimization methods.

Before reviewing the geometry of the Stiefel manifold, we introduce an important matrix decomposition called the QR decomposition, in which the Stiefel manifold naturally comes out. The standard QR decomposition of a full-rank $n \times p$ real matrix $B$ is put in the

form [GVL12, TBI97]

$$B = Q_0 R_0 = Q_0 \begin{pmatrix} R_1 \\ \\ 0 \end{pmatrix}, \qquad Q \in O(n),\ R_0 \in \mathbb{R}^{n\times p},\ R_1 \in S^+_{\mathrm{upp}}(p), \tag{2.3.2}$$

where $S^+_{\mathrm{upp}}(p)$ denotes the set of all $p \times p$ upper triangular matrices with strictly positive diagonal entries. Removing the zero block of $R_0$ in (2.3.2), we can also put the decomposition in the form

$$B = QR, \qquad Q \in \mathrm{St}(p,n),\ R \in S^+_{\mathrm{upp}}(p). \tag{2.3.3}$$

The decomposition of the form (2.3.3) is called the thin QR decomposition. In this thesis, we call the thin QR decomposition (2.3.3) simply the QR decomposition. The QR decomposition (2.3.3) proves to be unique and the columns of $Q$ can be explicitly written out under the Gram-Schmidt orthonormalization procedure. Then, we can define the map $\mathrm{qf}(B)$ of $\mathbb{R}^{n\times p}_*$ to $\mathrm{St}(p,n)$ by $\mathrm{qf}(B) = Q$. The qf is effectively used to define a retraction on the Stiefel manifold, which is a key notion to an iterative Riemannian optimization method.

We can verify the following useful property of qf about its derivative [AMS08]:

**Proposition 2.3.1.** *Suppose that $B$ is a full-rank $n \times p$ matrix with $p \leq n$ and is decomposed into* (2.3.3)*. Let $Z$ be an arbitrary $n \times p$ matrix. Then, the action on $Z$ of the derivative of* qf *at $B$ can be written using the $Q$ and $R$ factors in* (2.3.3) *as*

$$\mathrm{D}\,\mathrm{qf}(B)[Z] = B\rho_{\mathrm{skew}}(Q^T Z R^{-1}) + (I_n - BB^T)ZR^{-1}, \tag{2.3.4}$$

*where $\rho_{\mathrm{skew}}(\cdot)$ denotes the skew-symmetric part of the decomposition of the matrix in the parentheses into the sum of a skew-symmetric matrix and an upper triangular matrix (such a decomposition turns out to be unique).*

### 2.3.1 Tangent spaces and the induced metric

**Proposition 2.3.2.** *The tangent space $T_Y\mathrm{St}(p,n)$ at $Y \in \mathrm{St}(p,n)$ is given by*

$$T_Y\mathrm{St}(p,n) = \left\{\xi \in \mathbb{R}^{n\times p} \,|\, \xi^T Y + Y^T \xi = 0\right\}. \tag{2.3.5}$$

*Proof.* Let $\xi$ be an element of $T_Y\mathrm{St}(p,n)$. By differentiation, it follows from $Y^TY = I_p$ that $\xi^TY + Y^T\xi = 0$. Therefore, one has

$$T_Y\mathrm{St}(p,n) \subset \left\{\xi \in \mathbb{R}^{n\times p} \,|\, \xi^T Y + Y^T \xi = 0\right\}. \tag{2.3.6}$$

It remains to show that for any $n \times p$ matrix $\xi$ in the right-hand side of (2.3.6), there is a curve $Y(t)$ on $\mathrm{St}(p,n)$ emanating from $Y$ in the direction of $\xi$. We can specifically

construct such a curve $Y(t)$ as

$$Y(t) = \mathrm{qf}(Y + t\xi), \tag{2.3.7}$$

It is easy to see that $Y(t) \in \mathrm{St}(p,n)$ and $Y(0) = \mathrm{qf}(Y) = Y$. To complete the proof, we have only to prove $\dot{Y}(0) = \xi$. Note that since $Y \in \mathrm{St}(p,n)$, the QR decomposition of $Y$ is $Y = YI_p$. It follows from Prop. 2.3.1 with $Q = Y$ and $R = I_p$ that

$$\dot{Y}(0) = \mathrm{D}\,\mathrm{qf}(Y)[\xi] = Y\rho_{\mathrm{skew}}(Y^T\xi) + (I_n - YY^T)\xi = YY^T\xi + (I - YY^T)\xi = \xi, \tag{2.3.8}$$

where use has been made of $\rho_{\mathrm{skew}}(Y^T\xi) = Y^T\xi$, which is obtained from the fact that $Y^T\xi$ is skew-symmetric since $\xi^T Y + Y^T\xi = 0$. This ends the proof. □

Since the Stiefel manifold is a submanifold of the matrix Euclidean space $\mathbb{R}^{n\times p}$, it can be endowed with the Riemannian metric through

$$\langle \xi, \eta \rangle_Y := \mathrm{tr}\left(\xi^T\eta\right), \qquad \xi, \eta \in T_Y\mathrm{St}(p,n), \tag{2.3.9}$$

which is induced from the natural metric on $\mathbb{R}^{n\times p}$,

$$\langle B, C \rangle := \mathrm{tr}\left(B^T C\right), \qquad B, C \in \mathbb{R}^{n\times p}. \tag{2.3.10}$$

The orthogonal projection onto the tangent space $T_Y\mathrm{St}(p,n)$ will be of great help in optimization procedure.

**Proposition 2.3.3.** *The orthogonal projection operator $P_Y$ onto the tangent space $T_Y\mathrm{St}(p,n)$ is given, for any matrix $B \in \mathbb{R}^{n\times p}$, by*

$$P_Y(B) = (I_n - YY^T)B + Y\,\mathrm{skew}(Y^T B). \tag{2.3.11}$$

*Proof.* Let $\xi$ denote the right-hand side of (2.3.11). Since

$$\xi^T Y + Y^T\xi = \left(\mathrm{skew}(Y^T B)\right)^T + \mathrm{skew}(Y^T B) = 0, \tag{2.3.12}$$

$\xi$ is a tangent vector to $\mathrm{St}(p,n)$ at $Y \in \mathrm{St}(p,n)$. It remains to prove that $B - \xi$ is a normal vector at $Y$. To see this, for an arbitrary tangent vector $\eta \in T_Y\mathrm{St}(p,n)$, we calculate the inner product of $B - \xi$ and $\eta$ to obtain

$$\langle B - \xi, \eta \rangle_Y = \mathrm{tr}\left(B - \xi)^T\eta\right) = \mathrm{tr}\left(\left(Y\,\mathrm{sym}(Y^T B)\right)^T \eta\right) = \mathrm{tr}\left(\mathrm{sym}(Y^T B)Y^T\eta\right) = 0, \tag{2.3.13}$$

where we have used the fact that $Y^T\eta$ is skew-symmetric and the trace of the product of symmetric and skew-symmetric matrices is zero. This completes the proof. □

### 2.3.2 Geodesics

We shall find explicitly the exponential map on the Stiefel manifold by solving the geodesic equation.

**Proposition 2.3.4.** *The geodesic equation on the Stiefel manifold* $\mathrm{St}(p,n)$ *is expressed as*

$$\ddot{Y}(t) + Y(t)\dot{Y}(t)^T\dot{Y}(t) = 0. \tag{2.3.14}$$

Several proofs are known for Prop. 2.3.4, among which we shall give a proof after [EAS98].

*Proof.* Let $Y(t)$ be a geodesic on $\mathrm{St}(p,n)$. Differentiating $Y(t)^TY(t) = I_p$, we obtain

$$\ddot{Y}(t)^TY(t) + 2\dot{Y}(t)^T\dot{Y}(t) + Y(t)^T\ddot{Y}(t) = 0. \tag{2.3.15}$$

Since for the geodesic $Y(t)$, the second derivative $\ddot{Y}(t)$ with $t$ arbitrarily fixed is in the normal space to $Y(t)$, we have

$$0 = P_{Y(t)}(\ddot{Y}(t)) = \ddot{Y}(t) - Y(t)\,\mathrm{sym}(Y(t)^T\ddot{Y}(t)). \tag{2.3.16}$$

Let $S(t) = \mathrm{sym}(Y(t)^T\ddot{Y}(t))$, which is symmetric. In terms of $S(t)$, Eq. (2.3.15) takes the form

$$S(t) = -\dot{Y}(t)^T\dot{Y}(t). \tag{2.3.17}$$

Eqs. (2.3.17) and (2.3.16) are put together to result in (2.3.14). This completes the proof.□

We can describe solutions to the geodesic equation (2.3.14) as follows.

**Proposition 2.3.5.** *Let* $Y(t)$ *be a geodesic on the Stiefel manifold emanating from* $Y$ *in the direction of* $\xi \in T_Y\mathrm{St}(p,n)$*. Then,* $Y(t)$ *is expressed as*

$$Y(t) = \begin{pmatrix} Y & \xi \end{pmatrix} \exp\left( t \begin{pmatrix} Y^T\xi & -\xi^T\xi \\ I_p & Y^T\xi \end{pmatrix} \right) \begin{pmatrix} I_p \\ 0 \end{pmatrix} \exp(-Y^T\xi t), \tag{2.3.18}$$

*where* exp *denotes the matrix exponential.*

*Proof.* Let $Y_1(t)$ denote the right-hand side of (2.3.18). Differentiating $Y_1(t)$ with respect to $t$, we obtain

$$\dot{Y}_1(t) = \begin{pmatrix} Y & \xi \end{pmatrix} \exp\left( t \begin{pmatrix} Y^T\xi & -\xi^T\xi \\ I_p & Y^T\xi \end{pmatrix} \right) \begin{pmatrix} 0 \\ I_p \end{pmatrix} \exp(-Y^T\xi t) \tag{2.3.19}$$

and

$$\ddot{Y}_1(t) = \begin{pmatrix} Y & \xi \end{pmatrix} \exp\left( t \begin{pmatrix} Y^T\xi & -\xi^T\xi \\ I_p & Y^T\xi \end{pmatrix} \right) \begin{pmatrix} -\xi^T\xi \\ 0 \end{pmatrix} \exp(-Y^T\xi t). \tag{2.3.20}$$

Then, a straightforward calculation shows that $Y_1(t)$ satisfies the geodesic equation (2.3.14). As for initial values of $Y_1(t)$ and $\dot{Y}_1(t)$, it is obvious that $Y_1(0) = Y$ and $\dot{Y}_1(0) = \xi$. Thus, the theorem on existence and uniqueness of solutions to ordinary differential equations ensures that $Y(t) = Y_1(t)$. This completes the proof. □

### 2.3.3 Retractions

As was discussed in Subsection 2.2.1, the notion of a retraction provides a way to determine a next iterate with a given search direction. A typical example of a retraction is the exponential map. From Prop. 2.3.5, we can put the exponential map on the Stiefel manifold in the form

$$\mathrm{Exp}_Y(\xi) = \begin{pmatrix} Y & \xi \end{pmatrix} \exp\left( \begin{pmatrix} Y^T\xi & -\xi^T\xi \\ I_p & Y^T\xi \end{pmatrix} \right) \begin{pmatrix} I_p \\ 0 \end{pmatrix} \exp(-Y^T\xi), \qquad \xi \in T_Y\mathrm{St}(p,n). \tag{2.3.21}$$

We call the map $R : T\mathrm{St}(p,n) \to \mathrm{St}(p,n)$, determined by $R_Y = \mathrm{Exp}_Y$, the exponential retraction.

There is another retraction on the Stiefel manifold, which is based on the QR decomposition. By means of the map qf, we give the retraction based on the QR decomposition as follows:

**Proposition 2.3.6.** *Let $R_Y$ be the map of $T_Y\mathrm{St}(p,n)$ to $\mathrm{St}(p,n)$ defined by*

$$R_Y(\xi) = \mathrm{qf}(Y + \xi), \qquad \xi \in T_Y\mathrm{St}(p,n), \tag{2.3.22}$$

*Then, the collection of $R_Y$ for all $Y \in \mathrm{St}(p,n)$ forms a retraction $R : T\mathrm{St}(p,n) \to \mathrm{St}(p,n)$.*

*Proof.* It is clear that $R_Y(\xi) \in \mathrm{St}(p,n)$ from the definition of qf. The remaining task is to show that the $R_Y$ given by (2.3.22) satisfies the two conditions imposed in Definition 2.2.1. The first condition in Definition 2.2.1 is easy to verify; $R_Y(0) = \mathrm{qf}(Y) = Y$. In the same manner as in (2.3.8), the second condition in Definition 2.2.1 is also confirmed. This completes the proof. □

We call the $R$ defined through (2.3.22) the QR-based retraction.

### 2.3.4 Vector transport

There are several choices of vector transports on the Stiefel manifold $\mathrm{St}(p,n)$. We here introduce two vector transports $\mathcal{T}^R$ and $\mathcal{T}^P$ defined by (2.2.10) and (2.2.12) with $M = \mathrm{St}(p,n)$, respectively.

**Proposition 2.3.7.** *Let $R$ be the QR retraction defined through* (2.3.22) *on the Stiefel manifold* $\mathrm{St}(p,n)$*. We denote $R_Y(\eta)$ by $Q$ for short. Then, the corresponding vector transport $\mathcal{T}^R$ as the differentiated retraction and the vector transport $\mathcal{T}^P$ defined by* (2.2.12) *are written out as*

$$\mathcal{T}^R_\eta(\xi) = Q\rho_{\mathrm{skew}}\left(Q^T\xi\left(Q^T(Y+\eta)\right)^{-1}\right) + \left(I_n - QQ^T\right)\xi\left(Q^T(Y+\eta)\right)^{-1}, \tag{2.3.23}$$

*and*

$$\mathcal{T}^P_\eta(\xi) = (I_n - QQ^T)\xi + Q\,\mathrm{skew}(Q^T\xi), \tag{2.3.24}$$

*respectively.*

*Proof.* The proof is straightforward. Both (2.3.23) and (2.3.24) are verified immediately by using the formulas (2.3.4) and (2.3.11), respectively. □

# Chapter 3

# A New, Globally Convergent Riemannian Conjugate Gradient Method

## 3.1 Introduction

The conjugate gradient method was first developed by Hestenes and Stiefel as a tool for solving the linear equation $Ax = b$, where $A$ is an $n \times n$ positive definite matrix [HS52]. The strategy of the linear conjugate gradient method is to minimize the quadratic function $x^T Ax/2 - b^T x$ of $x$ in the successive search directions which are generated in such a manner that those directions are mutually conjugate with respect to $A$ and eventually span the whole $\mathbb{R}^n$. As this method is generalized to be applicable to functions which are not restricted to those quadratic in $x$, the conjugate gradient method in its original form is particularly called the linear conjugate gradient method.

According to a nonlinear conjugate gradient method for minimizing a smooth function $f$ which is not necessarily quadratic, the search direction $\eta_k$ is determined by

$$\eta_k = -\operatorname{grad} f(x_k) + \beta_k \eta_{k-1}, \tag{3.1.1}$$

where $\beta_k$ is a parameter to be defined suitably. Fletcher and Reeves [FR64] proposed to define $\beta_k$ by $\beta_k := \|\operatorname{grad} f(x_k)\|^2 / \|\operatorname{grad} f(x_{k-1})\|^2$ (see [NW06] for another way to determine $\beta_k$).

On the other hand, iterative optimization methods on $\mathbb{R}^n$ have been developed so as to be applicable on Riemannian manifolds [AMS08, EAS98]. Riemannian optimization methods provide procedures for minimizing objective functions defined on a Riemannian manifold $M$. In a Riemannian optimization method, the usual line search should be replaced [AMS08], as the concept of a line is generalized on a Riemannian manifold. Absil, Mahony, and Sepulchre proposed to use a retraction map to perform a search on a curve

on $M$ in place of the line search. As for the conjugate gradient method, Smith provided in [Smi94] a conjugate gradient method on $M$ along with other optimization algorithms on $M$. The difficulty we encounter in generalizing the conjugate gradient method to that on a manifold is that Eq. (3.1.1) makes no longer sense. This is because $\operatorname{grad} f(x_k)$ and $\eta_{k-1}$ belong to tangent spaces at different points on $M$ in general, so that they cannot be added. Smith proposed to use the parallel translation along the geodesic at each iteration in order to make possible the addition of two tangent vectors and thereby to extend the iteration procedure (3.1.1). However, using the parallel translation on $M$ is not computationally effective in general. A way to perform the conjugate gradient method on $M$ in an efficient manner is to use a vector transport [AMS08]. The global convergence in the conjugate gradient method with a vector transport on $M$ has been recently discussed by Ring and Wirth [RW12]. They proved the global convergence under the condition that the vector transport in use does not increase the norm of the search direction vector. To the contrary, the present chapter provides numerical evidence to show that if the assumption is not satisfied, the conjugate gradient method with a general vector transport may fail to generate a globally converging series. In order to relax the assumption in [RW12], the notion of a "scaled" vector transport is introduced in this chapter and a new conjugate gradient algorithm is proposed with only a mild computational overhead per iteration.

The organization of this chapter is as follows: The scaled vector transport is introduced in Section 3.2 after a brief review of some useful existing concepts. How to compute the step size is also discussed in this section. In Section 3.3, a brief review is made of the conjugate gradient method on a Riemannian manifold $M$, and then a new algorithm is proposed, in which the scaled vector transport is applied only if the vector transport increases the norm of the previous search direction. In Section 3.4, the global convergence for the proposed algorithm is proved in a manner similar to the usual one performed on $\mathbb{R}^n$, where the scaled vector transport used on a fitting occasion makes a generated sequence into a globally convergent one. Section 3.5 provides numerical experiments on simple problems which the existing algorithm cannot solve efficiently but the proposed algorithm can do. The numerical experiments show why the present algorithm can generate convergent sequences. Section 3.6 includes concluding remarks. It is shown in Appendix 3.7 that the Lipschitzian condition referred to in Subsection 3.4.1 is satisfied for some practical Riemannian optimization problems.

## 3.2 Setup for Riemannian optimization

### 3.2.1 Vector transport and scaled vector transport

In a (nonlinear) conjugate gradient method on the Euclidean space $\mathbb{R}^n$, the search directions $\eta_k$ are chosen to be

$$\eta_k = -\operatorname{grad} f(x_k) + \beta_k \eta_{k-1}, \qquad k \geq 0, \tag{3.2.1}$$

where $\beta_0 = 0$, and where $\beta_k$ with $k \geq 1$ are determined in several possible manners. For example, $\beta_k$ are determined by

$$\beta_k^{\mathrm{FR}} = \frac{\operatorname{grad} f(x_k)^T \operatorname{grad} f(x_k)}{\operatorname{grad} f(x_{k-1})^T \operatorname{grad} f(x_{k-1})}, \tag{3.2.2}$$

or

$$\beta_k^{\mathrm{PR}} = \frac{\operatorname{grad} f(x_k)^T (\operatorname{grad} f(x_k) - \operatorname{grad} f(x_{k-1}))}{\operatorname{grad} f(x_{k-1})^T \operatorname{grad} f(x_{k-1})}, \tag{3.2.3}$$

where FR and PR are abbreviations of Fletcher-Reeves and Polak-Ribière, respectively [NW06].

However, if $\mathbb{R}^n$ is replaced by a Riemannian manifold $M$, $\operatorname{grad} f(x_k) \in T_{x_k}M$ and $\eta_{k-1} \in T_{x_{k-1}}M$ belong to different tangent spaces, so that $-\operatorname{grad} f(x_k) + \beta_k \eta_{k-1}$ in Eq. (3.2.1) does not make sense. The quantity $\operatorname{grad} f(x_k) - \operatorname{grad} f(x_{k-1})$ in Eq. (3.2.3) makes no sense on $M$ either. In order to modify the vector addition in Eqs. (3.2.1) and (3.2.3) into a suitable operation on $M$, Smith proposed to use the parallel translation of tangent vectors along a geodesic [Smi94]. However, no computationally efficient formula is known for the parallel translation along a geodesic even for the Stiefel manifold except when it reduces to the sphere or the orthogonal group. Absil *et al.* [AMS08] proposed the notion of a vector transport as an alternative to the parallel translation. The vector transport is a generalization of the parallel translation and can enhance computational efficiency of algorithms, if defined suitably.

In this chapter, we focus on the differentiated retraction $\mathcal{T}^R$ as a vector transport, which is defined to be

$$\mathcal{T}^R_{\eta_x}(\xi_x) := \mathrm{D}R_x(\eta_x)[\xi_x], \qquad \eta_x, \xi_x \in T_xM, \tag{3.2.4}$$

where $R$ is a retraction on $M$. We here note that $\mathcal{T}^R$ satisfies the conditions in the definition of a vector transport, as is easily verified [AMS08].

In what follows, we assume that $M$ is a Riemannian manifold and denote the Riemannian metric evaluated at $x \in M$ by $\langle \cdot, \cdot \rangle_x$. The norm of a tangent vector $\xi_x \in T_xM$ evaluated at $x \in M$ is defined to be $\|\xi_x\|_x = \sqrt{\langle \xi_x, \xi_x \rangle}$. We here have to note that though the parallel translation is an isometry, a vector transport is not required to preserve the norm of vectors in general. The differentiated retraction $\mathcal{T}^R$ is not always an isometry either. In analyzing the convergence for the conjugate gradient method later, it will be crucial whether the vector transport $\mathcal{T}^R$ increases the norm of vectors or not. In order to prevent the vector transport $\mathcal{T}^R$ from increasing the norm of vectors, we define the scaled vector transport $\mathcal{T}^0 : TM \oplus TM \to TM$ associated with $\mathcal{T}^R$ as follows:

**Definition 3.2.1.** *Let $R$ be a retraction on a Riemannian manifold $M$. Let $\mathcal{T}^R$ be a vector transport defined by* (3.2.4) *with respect to $R$. The scaled vector transport $\mathcal{T}^0$ associated*

*with $\mathcal{T}^R$ is defined as*

$$\mathcal{T}^0_{\eta_x}(\xi_x) = \frac{\|\xi_x\|_x}{\|\mathcal{T}^R_{\eta_x}(\xi_x)\|_{R_x(\eta_x)}}\mathcal{T}^R_{\eta_x}(\xi_x), \qquad \eta_x, \xi_x \in T_xM. \tag{3.2.5}$$

The scaled vector transport $\mathcal{T}^0$ thus defined is no longer a vector transport since it is not linear. However, $\mathcal{T}^0$ satisfies

$$\|\mathcal{T}^0_{\eta_x}(\xi_x)\|_{R_x(\eta_x)} = \|\xi_x\|_x, \qquad \eta_x, \xi_x \in T_xM, \tag{3.2.6}$$

which is a key property for the global convergence of the algorithm we will propose.

### 3.2.2 Strong Wolfe conditions

In computing the step size $\alpha_k$ in the conjugate gradient method on $\mathbb{R}^n$, the strong Wolfe conditions are often used [NW06], which require $\alpha_k$ to satisfy

$$f(x_k + \alpha_k\eta_k) \leq f(x_k) + c_1\alpha_k \operatorname{grad} f(x_k)^T\eta_k, \tag{3.2.7}$$
$$|\operatorname{grad} f\,(x_k + \alpha_k\eta_k)^T\,\eta_k| \leq c_2|\operatorname{grad} f(x_k)^T\eta_k|, \tag{3.2.8}$$

with $0 < c_1 < c_2 < 1$. In particular, $c_1$ and $c_2$ are often taken so as to satisfy $0 < c_1 < c_2 < 1/2$ in the conjugate gradient method. In order to extend the strong Wolfe conditions on $\mathbb{R}^n$ to those on $M$, we start by reviewing the strong Wolfe conditions (3.2.7) and (3.2.8). For a current point $x_k$ and a search direction $\eta_k$, one performs a line search for the function defined by

$$\phi(\alpha) = f(x_k + \alpha\eta_k), \qquad \alpha > 0. \tag{3.2.9}$$

Requiring $\alpha_k$ to give a sufficient decrease in the value of $f$, one imposes the condition

$$\phi(\alpha_k) \leq \phi(0) + c_1\alpha_k\phi'(0), \tag{3.2.10}$$

which yields (3.2.7). In order to prevent $\alpha_k$ from being excessively short, the $\alpha_k$ is required to satisfy

$$|\phi'(\alpha_k)| \leq c_2|\phi'(0)|, \tag{3.2.11}$$

which implies (3.2.8).

In order to generalize the strong Wolfe conditions to those on $M$, we define a function $\phi$ on $M$, in an analogous manner to (3.2.9), to be

$$\phi(\alpha) = f\left(R_{x_k}(\alpha\eta_k)\right), \qquad \alpha > 0, \tag{3.2.12}$$

where $R$ is a retraction on $M$. The conditions (3.2.10) and (3.2.11) applied to (3.2.12) give

rise to

$$f\left(R_{x_k}(\alpha_k\eta_k)\right) \leq f(x_k) + c_1\alpha_k\langle \operatorname{grad} f(x_k), \eta_k\rangle_{x_k}, \tag{3.2.13}$$

$$|\langle \operatorname{grad} f\left(R_{x_k}(\alpha_k\eta_k)\right), \mathrm{D}R_{x_k}\left(\alpha_k\eta_k\right)[\eta_k]\rangle_{R_{x_k}(\alpha_k\eta_k)}| \leq c_2|\langle \operatorname{grad} f(x_k), \eta_k\rangle_{x_k}|, \tag{3.2.14}$$

respectively, where $0 < c_1 < c_2 < 1$. We call the conditions (3.2.13) and (3.2.14) the strong Wolfe conditions. The existence of a step size satisfying (3.2.13) and (3.2.14) can be shown by an almost verbatim repetition of that for the strong Wolfe conditions on $\mathbb{R}^n$ (see [NW06]).

**Proposition 3.2.1.** *Let $M$ be a Riemannian manifold with a retraction $R$. If a smooth objective function $f$ on $M$ is bounded below on $\{R_{x_k}(\alpha\eta_k)|\alpha > 0\}$ for $x_k \in M$ and for a descent direction $\eta_k \in T_{x_k}M$, and if constants $c_1$ and $c_2$ satisfy $0 < c_1 < c_2 < 1$, then there exists a step size $\alpha_k$ which satisfies the strong Wolfe conditions* (3.2.13) *and* (3.2.14)*.*

We note that the strong Wolfe conditions (3.2.13) and (3.2.14) together with the existence of a step size satisfying them are also discussed in [RW12].

We now look into the second condition (3.2.14). If we introduce a vector transport $\mathcal{T}^R$ as the differentiated retraction given by (3.2.4), then Eq. (3.2.14) can be expressed as

$$|\langle \operatorname{grad} f\left(R_{x_k}(\alpha_k\eta_k)\right), \mathcal{T}^R_{\alpha_k\eta_k}(\eta_k)\rangle_{R_{x_k}(\alpha_k\eta_k)}| \leq c_2|\langle \operatorname{grad} f(x_k), \eta_k\rangle_{x_k}|. \tag{3.2.15}$$

An idea for further generalization of this condition to that in an algorithm with a general vector transport $\mathcal{T}$ is to replace (3.2.15) by

$$|\langle \operatorname{grad} f\left(R_{x_k}(\alpha_k\eta_k)\right), \mathcal{T}_{\alpha_k\eta_k}(\eta_k)\rangle_{R_{x_k}(\alpha_k\eta_k)}| \leq c_2|\langle \operatorname{grad} f(x_k), \eta_k\rangle_{x_k}|. \tag{3.2.16}$$

However, if $\mathcal{T} \neq \mathcal{T}^R$, the existence of a step size satisfying both (3.2.13) and (3.2.16) is unclear in general. In view of this, the differentiated retraction $\mathcal{T}^R$ is considered to be a natural choice of a vector transport $\mathcal{T}$, for which a step size satisfying (3.2.13) and (3.2.16) is shown to exist. In what follows, we use the differentiated retraction $\mathcal{T}^R$ and the scaled one $\mathcal{T}^0$.

## 3.3 A new conjugate gradient method on a Riemannian manifold

If a Riemannian manifold $M$ is given a retraction $R$ and the corresponding vector transport $\mathcal{T}^R$, a standard Fletcher-Reeves type conjugate gradient method on $M$ is described as follows [AMS08, RW12]:

**Algorithm 3.3.1** A standard Fletcher-Reeves type conjugate gradient method for Problem 2.2.1 on a Riemannian manifold $M$

1: Choose an initial point $x_0 \in M$.
2: Set $\eta_0 = -\operatorname{grad} f(x_0)$.
3: **for** $k = 0, 1, 2, \ldots$ **do**
4: Compute the step size $\alpha_k > 0$ satisfying the strong Wolfe conditions (3.2.13) and (3.2.14) with $0 < c_1 < c_2 < 1/2$. Set

$$x_{k+1} = R_{x_k}(\alpha_k \eta_k), \tag{3.3.1}$$

where $R$ is a retraction on $M$.
5: Set

$$\beta_{k+1} = \frac{\|\operatorname{grad} f(x_{k+1})\|^2_{x_{k+1}}}{\|\operatorname{grad} f(x_k)\|^2_{x_k}}, \tag{3.3.2}$$

$$\eta_{k+1} = -\operatorname{grad} f(x_{k+1}) + \beta_{k+1} \mathcal{T}^R_{\alpha_k \eta_k}(\eta_k), \tag{3.3.3}$$

where $\mathcal{T}^R$ is the differentiated retraction defined by (3.2.4).
6: **end for**

In [RW12], the convergence property of Algorithm 3.3.1 is verified under the assumption that the inequality

$$\|\mathcal{T}^R_{\alpha_k \eta_k}(\eta_k)\|_{x_{k+1}} \leq \|\eta_k\|_{x_k} \tag{3.3.4}$$

holds for all $k \in \mathbb{N}$. However, the assumption does not always hold in general. For example, the assumption does not hold on the sphere endowed with the orthographic retraction [AM12]. In Section 3.5, we will numerically treat such a case.

We wish to relax the assumption (3.3.4) by using a scaled vector transport. An idea for improving Algorithm 3.3.1 is to replace $\mathcal{T}^R$ by the scaled vector transport $\mathcal{T}^0$ defined by (3.2.5). However, this causes difficulty in computing effectively a step size $\alpha_k$ satisfying (3.2.16) with $\mathcal{T} = \mathcal{T}^0$.

A simple but effective idea for improving Algorithm 3.3.1 is that each step size is always computed so as to satisfy the strong Wolfe conditions (3.2.13) and (3.2.14), but the scaled vector transport $\mathcal{T}^0$ is adopted if it is necessary for the purpose of convergence. More specifically, we use the scaled vector transport $\mathcal{T}^0$ only if the vector transport $\mathcal{T}^R$ increases the norm of the previous search direction vector, that is, we introduce $\mathcal{T}^{(k)}$ defined by

$$\mathcal{T}^{(k)}_{\alpha_k \eta_k}(\eta_k) = \begin{cases} \mathcal{T}^R_{\alpha_k \eta_k}(\eta_k), & \text{if } \|\mathcal{T}^R_{\alpha_k \eta_k}(\eta_k)\|_{x_{k+1}} \leq \|\eta_k\|_{x_k}, \\ \mathcal{T}^0_{\alpha_k \eta_k}(\eta_k), & \text{otherwise,} \end{cases} \tag{3.3.5}$$

as a substitute for $\mathcal{T}^R$ in Step 5 of Algorithm 3.3.1. This idea is realized in the following algorithm.

**Algorithm 3.3.2** A scaled Fletcher-Reeves type conjugate gradient method for Problem 2.2.1 on a Riemannian manifold $M$

1: Choose an initial point $x_0 \in M$.
2: Set $\eta_0 = -\operatorname{grad} f(x_0)$.
3: **for** $k = 0, 1, 2, \ldots$ **do**
4: Compute the step size $\alpha_k > 0$ satisfying the strong Wolfe conditions (3.2.13) and (3.2.14) with $0 < c_1 < c_2 < 1/2$. Set

$$x_{k+1} = R_{x_k}(\alpha_k \eta_k), \tag{3.3.6}$$

where $R$ is a retraction on $M$.
5: Set

$$\beta_{k+1} = \frac{\|\operatorname{grad} f(x_{k+1})\|^2_{x_{k+1}}}{\|\operatorname{grad} f(x_k)\|^2_{x_k}}, \tag{3.3.7}$$

$$\eta_{k+1} = -\operatorname{grad} f(x_{k+1}) + \beta_{k+1} \mathcal{T}^{(k)}_{\alpha_k \eta_k}(\eta_k), \tag{3.3.8}$$

where $\mathcal{T}^{(k)}$ is defined by (3.3.5), and where $\mathcal{T}^R$ and $\mathcal{T}^0$ are the differentiated retraction and the associated scaled vector transport defined by (3.2.4) and (3.2.5), respectively.
6: **end for**

We will prove in Section 3.4 the global convergence property of the proposed algorithm, and give in Section 3.5 numerical examples in which the inequality (3.3.4) does not hold for all $k \in \mathbb{N}$ but our Algorithm 3.3.2 indeed has an advantage in generating convergent sequences.

## 3.4 Convergence analysis of the new algorithm

In this section, we verify the convergence property of Algorithm 3.3.2.

### 3.4.1 Zoutendijk's theorem

Zoutendijk's theorem about a series associated with search directions on $\mathbb{R}^n$ is not only valid for the conjugate gradient method but also valid for general descent algorithms [NW06]. This theorem can be generalized so as to be applicable to a general descent algorithm (Algorithm 2.2.1) on a Riemannian manifold $M$. In the same manner as in $\mathbb{R}^n$, we define on a Riemannian manifold $M$ the angle $\theta_k$ between the steepest descent direction $-\operatorname{grad} f(x_k)$ and the search direction $\eta_k$ through

$$\cos\theta_k = -\frac{\langle \operatorname{grad} f(x_k), \eta_k \rangle_{x_k}}{\|\operatorname{grad} f(x_k)\|_{x_k} \|\eta_k\|_{x_k}}. \tag{3.4.1}$$

Then, Zoutendijk's theorem on $M$ is stated as follows:

**Theorem 3.4.1.** *Suppose that in Algorithm 2.2.1 on a Riemannian manifold $M$, a descent direction $\eta_k$ and a step size $\alpha_k$ satisfy the strong Wolfe conditions* (3.2.13) *and* (3.2.14)*. If the objective function $f$ is bounded below and of $C^1$-class, and if there exists a Lipschitzian constant $L > 0$ such that*

$$|\mathrm{D}(f \circ R_x)(t\eta)[\eta] - \mathrm{D}(f \circ R_x)(0)[\eta]| \leq Lt, \qquad \eta \in T_xM \text{ with } \|\eta\|_x = 1,\ x \in M,\ t \geq 0, \tag{3.4.2}$$

*then the following series converges;*

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \|\mathrm{grad}\, f(x_k)\|_{x_k}^2 < \infty. \tag{3.4.3}$$

The proof of this theorem can be performed in the same manner as that for Zoutendijk's theorem on $\mathbb{R}^n$. See [RW12] for more detail.

**Remark 3.4.1.** *We remark that the inequality* (3.4.2) *is a weaker condition than the Lipschitz continuous differentiability of $f \circ R_x$. We will show in Appendix 3.7 that Eq.* (3.4.2) *holds for objective functions in practical Riemannian optimization problems. A further discussion on the relation with the standard Lipschitz continuous differentiability will be also made in the same appendix.*

### 3.4.2 Global convergence

We first extend a lemma in [AB85] so as to be applicable to Algorithm 3.3.2 as follows:

**Lemma 3.4.1.** *The search direction $\eta_k$ determined in Algorithm 3.3.2 is a descent direction satisfying*

$$-\frac{1}{1-c_2} \leq \frac{\langle \mathrm{grad}\, f(x_k), \eta_k \rangle_{x_k}}{\|\mathrm{grad}\, f(x_k)\|_{x_k}^2} \leq \frac{2c_2 - 1}{1 - c_2}. \tag{3.4.4}$$

*Proof.* The proof runs by induction. For $k = 0$, the inequality (3.4.4) clearly holds on account of

$$\frac{\langle \mathrm{grad}\, f(x_0), \eta_0 \rangle_{x_0}}{\|\mathrm{grad}\, f(x_0)\|_{x_0}^2} = \frac{\langle \mathrm{grad}\, f(x_0), -\mathrm{grad}\, f(x_0) \rangle_{x_0}}{\|\mathrm{grad}\, f(x_0)\|_{x_0}^2} = -1. \tag{3.4.5}$$

We here note that $0 < c_1 < c_2 < 1/2$. Suppose that $\eta_k$ is a descent direction satisfying (3.4.4) for some $k$. Note that on account of Eq. (3.3.8) with Eq. (3.3.5), $\mathcal{T}^R$ and $\mathcal{T}^{(k)}$ are related by $\|\mathcal{T}^{(k)}_{\alpha_k\eta_k}(\eta_k)\|_{x_{k+1}} \leq \|\mathcal{T}^R_{\alpha_k\eta_k}(\eta_k)\|_{x_{k+1}}$ in each case. Since $\mathcal{T}^{(k)}_{\alpha_k\eta_k}(\eta_k)$ and $\mathcal{T}^R_{\alpha_k\eta_k}(\eta_k)$ are in the same direction with the inequality $\|\mathcal{T}^{(k)}_{\alpha_k\eta_k}(\eta_k)\|_{x_{k+1}} \leq \|\mathcal{T}^R_{\alpha_k\eta_k}(\eta_k)\|_{x_{k+1}}$ in norm, we have

$$|\langle \mathrm{grad}\, f(x_{k+1}), \mathcal{T}^{(k)}_{\alpha_k\eta_k}(\eta_k) \rangle_{x_{k+1}}| \leq |\langle \mathrm{grad}\, f(x_{k+1}), \mathcal{T}^R_{\alpha_k\eta_k}(\eta_k) \rangle_{x_{k+1}}|. \tag{3.4.6}$$

We also note that the vector transport $\mathcal{T}^R$ is defined to be $\mathcal{T}^R_{\eta_x}(\xi_x) = \mathrm{D}R_x(\eta_x)[\xi_x]$ in the algorithm. It then follows from (3.2.14) and (3.4.6) that

$$c_2\langle \operatorname{grad} f(x_k), \eta_k\rangle_{x_k} \leq \langle \operatorname{grad} f(x_{k+1}), \mathcal{T}^{(k)}_{\alpha_k\eta_k}(\eta_k)\rangle_{x_{k+1}} \leq -c_2\langle \operatorname{grad} f(x_k), \eta_k\rangle_{x_k}, \tag{3.4.7}$$

where it is to be noted that $\eta_k$ is in a descent direction. The middle term in (3.4.4) with $k+1$ for $k$ is computed as

$$\begin{aligned}\frac{\langle \operatorname{grad} f(x_{k+1}), \eta_{k+1}\rangle_{x_{k+1}}}{\|\operatorname{grad} f(x_{k+1})\|^2_{x_{k+1}}} =& \frac{\langle \operatorname{grad} f(x_{k+1}), -\operatorname{grad} f(x_{k+1}) + \beta_{k+1}\mathcal{T}^{(k)}_{\alpha_k\eta_k}(\eta_k)\rangle_{x_{k+1}}}{\|\operatorname{grad} f(x_{k+1})\|^2_{x_{k+1}}} \\ =& -1 + \frac{\langle \operatorname{grad} f(x_{k+1}), \mathcal{T}^{(k)}_{\alpha_k\eta_k}(\eta_k)\rangle_{x_{k+1}}}{\|\operatorname{grad} f(x_k)\|^2_{x_k}},\end{aligned} \tag{3.4.8}$$

where the definition (3.3.7) of $\beta_{k+1}$ has been used. Therefore, we obtain from (3.4.7) and (3.4.8)

$$-1 + c_2\frac{\langle \operatorname{grad} f(x_k), \eta_k\rangle_{x_k}}{\|\operatorname{grad} f(x_k)\|^2_{x_k}} \leq \frac{\langle \operatorname{grad} f(x_{k+1}), \eta_{k+1}\rangle_{x_{k+1}}}{\|\operatorname{grad} f(x_{k+1})\|^2_{x_{k+1}}} \leq -1 - c_2\frac{\langle \operatorname{grad} f(x_k), \eta_k\rangle_{x_k}}{\|\operatorname{grad} f(x_k)\|^2_{x_k}}. \tag{3.4.9}$$

The inequality (3.4.4) for $k+1$ immediately follows from the induction hypothesis. □

We proceed to the global convergence property of Algorithm 3.3.2. The convergence of the conjugate gradient method has been already proved on $\mathbb{R}^n$ by Al-Baali [AB85]. Exploiting the idea of the proof used in [AB85], we show that Algorithm 3.3.2 generates converging sequences on a Riemannian manifold.

**Theorem 3.4.2.** *Consider Algorithm 3.3.2. Suppose that $f$ is bounded below and of $C^1$-class. If* (3.4.2) *and hence* (3.4.3) *hold, then*

$$\liminf_{k\to\infty} \|\operatorname{grad} f(x_k)\|_{x_k} = 0. \tag{3.4.10}$$

*Proof.* If $\operatorname{grad} f(x_k) = 0$ for some $k$, let $k_0$ be the smallest integer among such $k$. Then, we have $\beta_{k_0} = 0$ and $\eta_{k_0} = 0$ from (3.3.7) and (3.3.8) with $k_0 = k+1$, so that $x_{k_0+1} = R_{x_{k_0}}(\alpha_{k_0}\eta_{k_0}) = R_{x_{k_0}}(0) = x_{k_0}$. It then follows that $\operatorname{grad} f(x_k) = 0$ for all $k \geq k_0$. Eq. (3.4.10) clearly holds in such a case.

We shall consider the case in which $\operatorname{grad} f(x_k) \neq 0$ for all $k$ and prove (3.4.10) by contradiction. Assume that (3.4.10) does not hold, that is, there exists a constant $\gamma > 0$ such that

$$\|\operatorname{grad} f(x_k)\|_{x_k} \geq \gamma > 0, \qquad \forall k \geq 0. \tag{3.4.11}$$

Now from (3.4.1) and (3.4.4), we obtain

$$\cos\theta_k \geq \frac{1-2c_2}{1-c_2}\frac{\|\operatorname{grad} f(x_k)\|_{x_k}}{\|\eta_k\|_{x_k}}. \tag{3.4.12}$$

On account of Thm. 3.4.1, Eqs. (3.4.3) and (3.4.12) are put together to provide

$$\sum_{k=0}^{\infty} \frac{\|\mathrm{grad}\, f(x_k)\|_{x_k}^4}{\|\eta_k\|_{x_k}^2} < \infty. \tag{3.4.13}$$

On the other hand, Eqs. (3.4.6), (3.4.4), and the strong Wolfe condition (3.2.14) are put together to give

$$\begin{aligned} |\langle \mathrm{grad}\, f(x_k), \mathcal{T}^{(k-1)}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1})\rangle_{x_k}| \leq & |\langle \mathrm{grad}\, f(x_k), \mathcal{T}^{R}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1})\rangle_{x_k}| \\ \leq & -c_2\langle \mathrm{grad}\, f(x_{k-1}), \eta_{k-1}\rangle_{x_{k-1}} \\ \leq & \frac{c_2}{1-c_2}\|\mathrm{grad}\, f(x_{k-1})\|_{x_{k-1}}^2. \end{aligned} \tag{3.4.14}$$

Using this inequality and the definition of $\beta_k$, we obtain the recurrence inequality for $\|\eta_k\|_{x_k}^2$ as follows:

$$\begin{aligned} & \|\eta_k\|_{x_k}^2 \\ =& \|-\mathrm{grad}\, f(x_k) + \beta_k \mathcal{T}^{(k-1)}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1})\|_{x_k}^2 \\ \leq & \|\mathrm{grad}\, f(x_k)\|_{x_k}^2 + 2\beta_k |\langle \mathrm{grad}\, f(x_k), \mathcal{T}^{(k-1)}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1})\rangle_{x_k}| + \beta_k^2 \|\mathcal{T}^{(k-1)}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1})\|_{x_k}^2 \\ \leq & \|\mathrm{grad}\, f(x_k)\|_{x_k}^2 + \frac{2c_2}{1-c_2}\beta_k \|\mathrm{grad}\, f(x_{k-1})\|_{x_{k-1}}^2 + \beta_k^2 \|\eta_{k-1}\|_{x_{k-1}}^2 \\ =& c\|\mathrm{grad}\, f(x_k)\|_{x_k}^2 + \beta_k^2 \|\eta_{k-1}\|_{x_{k-1}}^2, \end{aligned} \tag{3.4.15}$$

where we have used the fact that $\|\mathcal{T}^{(k-1)}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1})\|_{x_k} \leq \|\eta_{k-1}\|_{x_{k-1}}$ and put $c := (1+c_2)/(1-c_2) > 1$. The successive use of this inequality together with the definition of $\beta_k$ results in

$$\begin{aligned} & \|\eta_k\|_{x_k}^2 \\ \leq & c\left(\|\mathrm{grad}\, f(x_k)\|_{x_k}^2 + \beta_k^2 \|\mathrm{grad}\, f(x_{k-1})\|_{x_{k-1}}^2 + \cdots + \beta_k^2\beta_{k-1}^2 \cdots \beta_2^2 \|\mathrm{grad}\, f(x_1)\|_{x_1}^2\right) \\ & + \beta_k^2\beta_{k-1}^2 \cdots \beta_1^2 \|\eta_0\|_{x_0}^2 \\ =& c\|\mathrm{grad}\, f(x_k)\|_{x_k}^4 \left(\|\mathrm{grad}\, f(x_k)\|_{x_k}^{-2} + \|\mathrm{grad}\, f(x_{k-1})\|_{x_{k-1}}^{-2} + \cdots + \|\mathrm{grad}\, f(x_1)\|_{x_1}^{-2}\right) \\ & + \|\mathrm{grad}\, f(x_k)\|_{x_k}^4 \|\mathrm{grad}\, f(x_0)\|_{x_0}^{-2} \\ < & c\|\mathrm{grad}\, f(x_k)\|_{x_k}^4 \sum_{j=0}^{k} \|\mathrm{grad}\, f(x_j)\|_{x_j}^{-2} \leq \frac{c}{\gamma^2}\|\mathrm{grad}\, f(x_k)\|_{x_k}^4 (k+1), \end{aligned} \tag{3.4.16}$$

where use has been made of (3.4.11) in the last inequality. The inequality (3.4.16) gives

rise to

$$\sum_{k=0}^{\infty} \frac{\|\operatorname{grad} f(x_k)\|_{x_k}^4}{\|\eta_k\|_{x_k}^2} \geq \sum_{k=0}^{\infty} \frac{\gamma^2}{c} \frac{1}{k+1} = \infty. \tag{3.4.17}$$

This contradicts (3.4.13) and the proof is completed. □

## 3.5 Numerical experiments

In this section, we compare Algorithm 3.3.2 with Algorithm 3.3.1 by numerical experiments. As is shown in [RW12], if the vector transport $\mathcal{T}^R$ as the differentiated retraction satisfies the inequality (3.3.4), the convergence property of Algorithm 3.3.1 is proved. However, if (3.3.4) does not hold, it is not always ensured that sequences generated by Algorithm 3.3.1 converge. In contrast with this, Algorithm 3.3.2 indeed works well even if (3.3.4) fails to hold, as is verified in Thm. 3.4.2. In the following, we give two examples which show that Algorithm 3.3.2 works better than Algorithm 3.3.1. One of the examples is somewhat artificial but well illustrates the situation in which a sequence generated by Algorithm 3.3.1 is unlikely to converge. The other is a more natural example encountered in a practical problem.

In both of two examples, we consider the following Rayleigh quotient minimization problem on the sphere $S^{n-1} := \left\{x \in \mathbb{R}^n \,|\, x^T x = 1\right\}$ [AMS08, HM94]:

**Problem 3.5.1.**

$$\text{minimize} \quad f(x) = x^T A x, \tag{3.5.1}$$
$$\text{subject to} \ \ x \in S^{n-1}, \tag{3.5.2}$$

where $A := \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ with $\lambda_1 < \lambda_2 < \cdots < \lambda_n$. The optimal solutions of this problem are $\pm(1, 0, 0, \ldots, 0)^T$, which are the unit eigenvectors of $A$ associated with the smallest eigenvalue $\lambda_1$.

### 3.5.1 A sphere endowed with a peculiar metric

Consider Problem 3.5.1 with $n = 20$ and $A = \operatorname{diag}(1, 2, \ldots, 20)$. A Riemannian metric $g(\cdot, \cdot)$ on $S^{n-1}$ is here defined by

$$g_x(\xi_x, \eta_x) := \xi_x^T G_x \eta_x, \qquad \xi_x, \eta_x \in T_x S^{n-1}, \tag{3.5.3}$$

where $G_x := \operatorname{diag}(10000(x^{(1)})^2 + 1, 1, 1, \ldots, 1)$, and where $x^{(1)}$ denotes the first component of the column vector $x$. It is to be noted that this metric is not the standard one on $S^{n-1}$. The norm $\|\xi_x\|_x$ of $\xi_x \in T_x S^{n-1}$ is then defined to be $\|\xi_x\|_x = \sqrt{g_x(\xi_x, \xi_x)}$. If $x$ is close to the optimal solutions $\pm(1, 0, 0, \ldots, 0)$, then $(x^{(1)})^2$ is nearly 1. Since the first diagonal

element of $G_x$ is large because of the coefficient 10000, the closer $x$ is to $\pm(1,0,0,\ldots,0)$, the larger the norm $\|\xi_x\|_x$ tends to be.

With respect to the metric (3.5.3), the gradient of $f$ is described as

$$\operatorname{grad} f(x) = 2\left(I - \frac{G_x^{-1}xx^T}{x^T G_x^{-1}x}\right) G_x^{-1}Ax. \tag{3.5.4}$$

Indeed, the right-hand side of (3.5.4) belongs to $T_xS^{n-1} = \left\{\xi \in \mathbb{R}^n \mid x^T\xi = 0\right\}$ and it holds that

$$g_x\left(2\left(I - \frac{G_x^{-1}xx^T}{x^T G_x^{-1}x}\right) G_x^{-1}Ax,\ \xi\right) = 2x^TA\xi = \mathrm{D}f(x)[\xi] \tag{3.5.5}$$

for any $\xi \in T_xS^{n-1}$. Let $R$ be the retraction on $S^{n-1}$ defined by

$$R_x(\xi) = \frac{x+\xi}{\sqrt{(x+\xi)^T(x+\xi)}}, \qquad \xi \in T_xS^{n-1},\ x \in S^{n-1}, \tag{3.5.6}$$

which is the special case of the QR retraction (3.7.5) on the Stiefel manifold defined in Appendix 3.7. For this $R$, the differentiated retraction $\mathcal{T}^R$ defined by (3.2.4) is written out as

$$\mathcal{T}^R_\eta(\xi) = \frac{1}{\sqrt{(x+\eta)^T(x+\eta)}}\left(I - \frac{(x+\eta)(x+\eta)^T}{(x+\eta)^T(x+\eta)}\right)\xi, \qquad \eta,\xi \in T_xS^{n-1},\ x \in S^{n-1}. \tag{3.5.7}$$

We note that though the metric endowed with is not the standard one, the Lipschitzian condition (3.4.2) holds, as is mentioned in Rem. 3.7.2 in Appendix 3.7. Hence from Thm. 3.4.2, Algorithm 3.3.2 works well in theory.

Figs. 3.5.1, 3.5.2, and 3.5.3 show numerical results from applying Algorithm 3.3.1 to Problem 3.5.1 with the initial point $x_0 = (1,1,\ldots,1)^T/2\sqrt{5} \in S^{n-1}$ with $n = 20$. The vertical axes of Figs. 3.5.1, 3.5.2, and 3.5.3 carry values of $f(x_k)$ at $x_k$, values of the first components $x_k^{(1)}$ of $x_k$, and values of the ratios $\|\mathcal{T}^R_{\alpha_k\eta_k}(\eta_k)\|_{x_{k+1}}/\|\eta_k\|_{x_k}$, respectively. Note that for the optimal solution $x_* = (1,0,0,\ldots,0)^T \in S^{n-1}$ which the current generated sequence $\{x_k\}$ is expected to approach, the target value is $f(x_*) = x_*^{(1)} = 1$ in both Figs. 3.5.1 and 3.5.2. Though the $\{x_k\}$ seems to come close to $x_*$ bit by bit, the convergence is not observed even after $10^5$ iterations. At the iteration number $10^5$, $f(x_k)$ is far from $f(x_*) = 1$, as is seen from Fig. 3.5.1. Fig. 3.5.2 shows that the sequence is intermittently repelled from the target point, when approaching it. If more iterations, say $10^7$, are performed, the graph of $\{x_k^{(1)}\}$ has almost the same shape, that is, sharp peaks repeatedly appear in Fig. 3.5.2 with extended iterations. If $\|\mathcal{T}^R_{\alpha_k\eta_k}(\eta_k)\|_{x_{k+1}}/\|\eta_k\|_{x_k} \leq 1$ for all $k \in \mathbb{N}$, the sequence $\{x_k\}$ would converge. However, as is shown in Fig. 3.5.3, the ratio $\|\mathcal{T}^R_{\alpha_k\eta_k}(\eta_k)\|_{x_{k+1}}/\|\eta_k\|_{x_k}$ intermittently exceeds the value 1. This fact seems to prevent the sequence from converging, as long as numerical experiments suggest. To gain insight into the non-convergence problem, we put Figs. 3.5.2 and 3.5.3 together into Fig. 3.5.4, which shows that the peaks of two

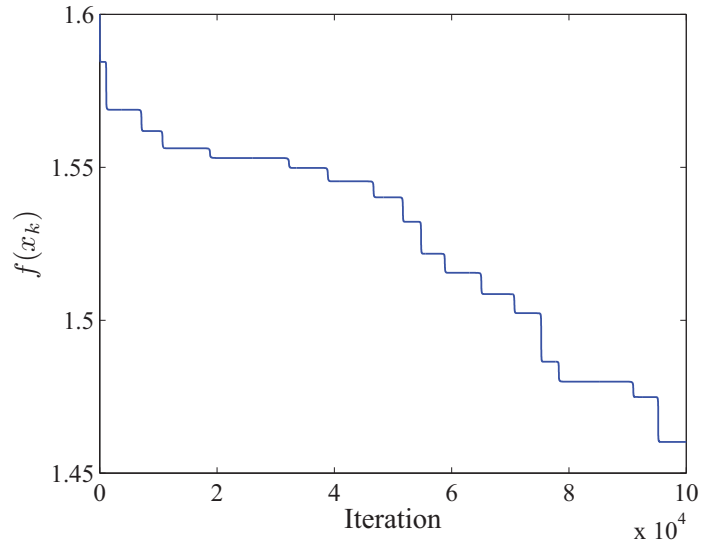



Figure 3.5.1: The sequence of the values $f(x_k)$ of the objective function $f$ evaluated on the sequence $\{x_k\}$ generated by Algorithm 3.3.1.

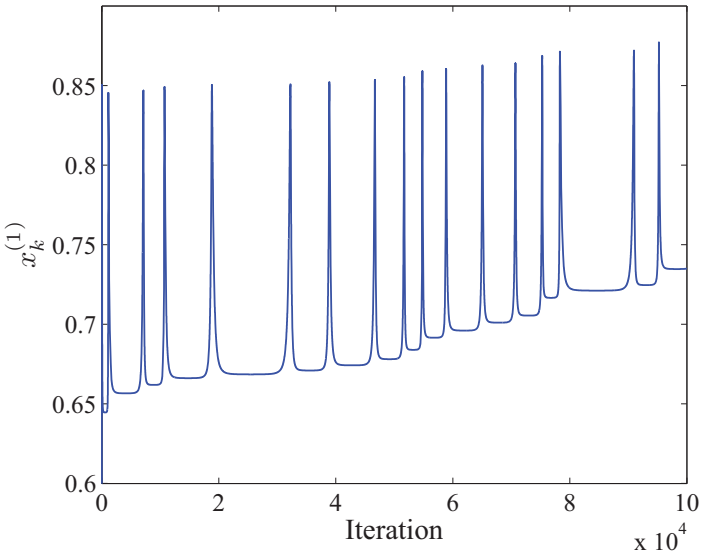



Figure 3.5.2: The sequence of the first components $x_k^{(1)}$ from the sequence $\{x_k\}$ generated by Algorithm 3.3.1.

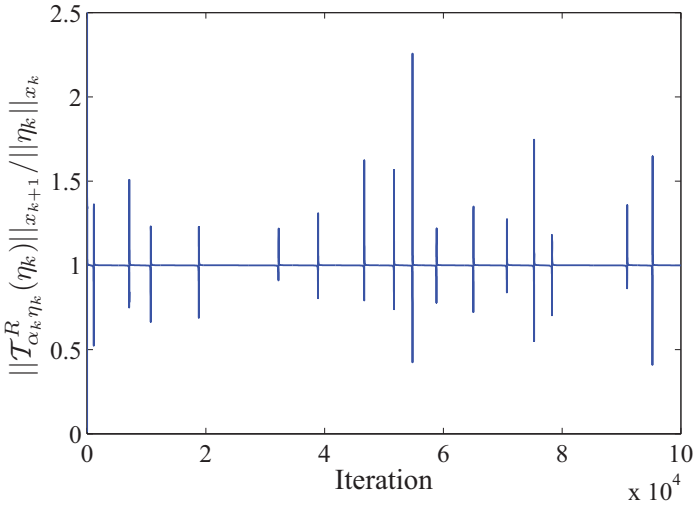



Figure 3.5.3: Ratios $\|\mathcal{T}^R_{\alpha_k\eta_k}(\eta_k)\|_{x_{k+1}}/\|\eta_k\|_{x_k}$ evaluated on the sequences $\{x_k\}$ and $\{\eta_k\}$ generated by Algorithm 3.3.1.

graphs synchronize. This suggests that the violation of the inequality (3.3.4) makes the sequence fail to approach the optimal solution $x_*$. This phenomenon is caused by the large first diagonal element of $G_x$ in the neighborhood of $x_*$.

In contrast with this, in Algorithm 3.3.2, the vector transport $\mathcal{T}^R$ is scaled if necessary, and thereby generated sequences converge to solve Problem 3.5.1. In comparison with Fig. 3.5.2, Fig. 3.5.5 shows that the present algorithm generates a converging sequence, resolving the difficulty of being repelled from the optimal solution. We here note that the inequality $\|\mathcal{T}^{(k)}_{\alpha_k\eta_k}(\eta_k)\|_{x_{k+1}} \leq \|\eta_k\|_{x_k}$ is never violated in this algorithm.

We now investigate the performance of Algorithm 3.3.2 in more detail with interest in comparison with a restart strategy in the conjugate gradient method. As is well known, in a nonlinear conjugate gradient method on the Euclidean space, the iteration is often restarted at every $N$ steps by taking a steepest descent search direction, where $N$ is usually chosen to be the dimension of the search space in the problem. To gain a sight of the performance of the restart method on a Riemannian manifold, we introduce a similar restart strategy into Algorithms 3.3.1 and 3.3.2, that is, we set $\beta_{k+1} = 0$ in Step 5 of each algorithm at every $N$ steps. A choice for $N$ is 19, which is the dimension of $S^{n-1}$ with $n = 20$. For comparison, the both algorithms with restarts are also performed for $N = 50$ and $N = 100$. The results from Algorithm 3.3.2 with and without restart are shown in Fig. 3.5.6. The vertical axis of Fig. 3.5.6 carries $\sqrt{(x_k - x_*)^T(x_k - x_*)}$, which is an approximation of the distance between $x_k$ and $x_*$ on $S^{n-1}$. We can observe from the graphs in Fig. 3.5.6 that Algorithm 3.3.2 with and without restart has a superlinear convergence property. Fig. 3.5.6 shows further that Algorithm 3.3.2 without restart exhibits better performance than Algorithm

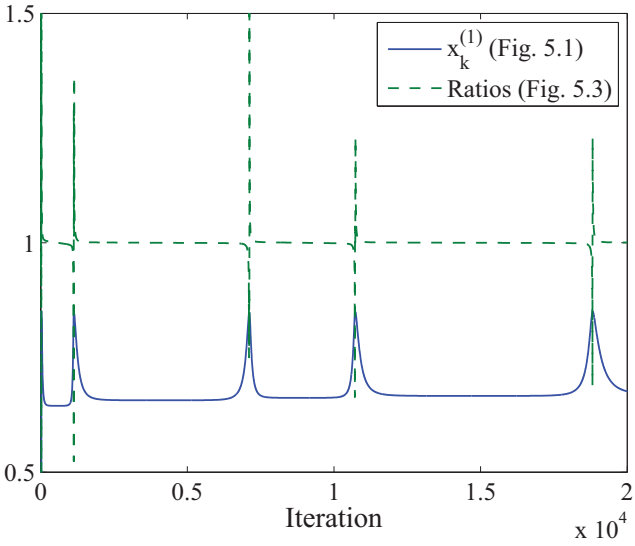



Figure 3.5.4: $x_k^{(1)}$ and $\|\mathcal{T}^R_{\alpha_k\eta_k}(\eta_k)\|_{x_{k+1}}/\|\eta_k\|_{x_k}$ by Algorithm 3.3.1.

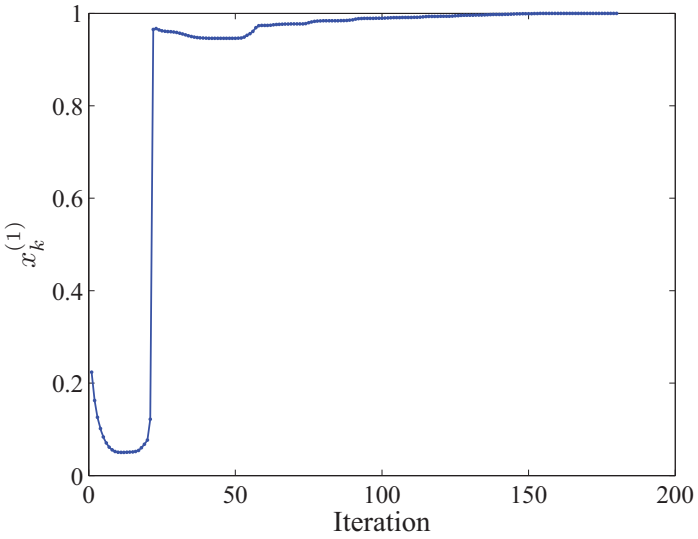



Figure 3.5.5: The sequence of the first components $x_k^{(1)}$ from the sequence $\{x_k\}$ generated by Algorithm 3.3.2.

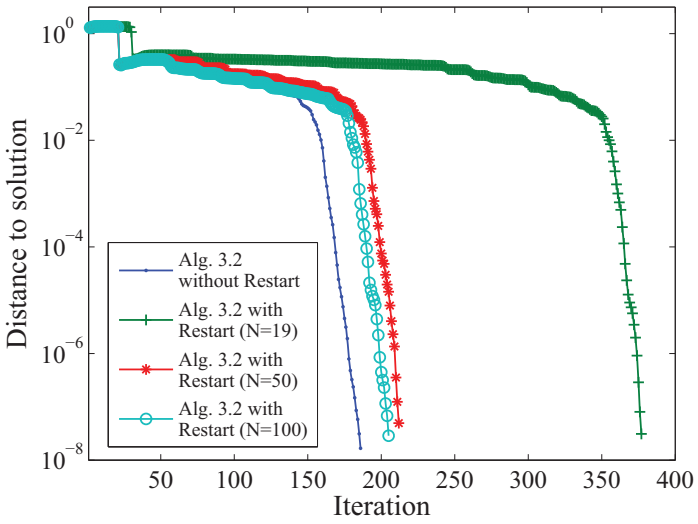



Figure 3.5.6: The sequences of the distances between $x_k$ and $x_*$ with respect to the sequences $\{x_k\}$ generated by Algorithm 3.3.2 with several restarting strategies.

3.3.2 with a few variants of restarts, which means that the restart strategy fails to improve the performance of Algorithm 3.3.2.

On the contrary, the restart strategy improves the performance of Algorithm 3.3.1, but the resultant performance is not comparable to Algorithm 3.3.2 without restart yet. A numerical evidence is shown in Fig. 3.5.7.

### 3.5.2 The sphere endowed with the orthographic retraction

We give a more natural example, in which the inequality (3.3.4) is never satisfied. Consider Problem 3.5.1 with $n = 100$ and $A = \mathrm{diag}(1, 2, \ldots, 100)/100$. The difference from the example in Subsection 3.5.1 is the choice of a Riemannian metric and a retraction. We in turn endow the sphere $S^{n-1}$ with the induced metric $\langle \cdot, \cdot \rangle$ from the natural inner product on $\mathbb{R}^n$:

$$\langle \xi_x, \eta_x \rangle_x := \xi_x^T \eta_x, \qquad \xi_x, \eta_x \in T_x S^{n-1}. \tag{3.5.8}$$

The norm of $\xi_x \in T_x S^{n-1}$ is then defined to be $\|\xi_x\|_x = \sqrt{\xi_x^T \xi_x}$ as usual. With the natural metric $\langle \cdot, \cdot \rangle$, the gradient of $f$ is written out as

$$\mathrm{grad}\, f(x) = 2(I - xx^T)Ax. \tag{3.5.9}$$

We consider the orthographic retraction $R$ on $S^{n-1}$ [AM12], which is defined to be

$$R_x(\xi) = \sqrt{1 - \xi^T \xi}\, x + \xi, \qquad \xi \in T_x S^{n-1} \text{ with } \|\xi\|_x < 1. \tag{3.5.10}$$

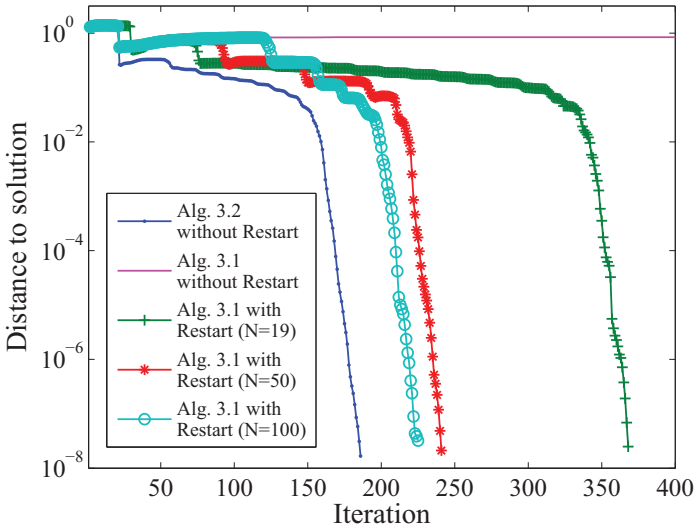



Figure 3.5.7: The sequences of the distances between $x_k$ and $x_*$ with respect to the sequences $\{x_k\}$ generated by Algorithm 3.3.2 and Algorithm 3.3.1 with several restarting strategies.

Associated with this $R$, the vector transport $\mathcal{T}^R$ is written out as

$$\mathcal{T}^R_\eta(\xi) = \xi - \frac{\eta^T \xi}{\sqrt{1-\eta^T\eta}} x, \qquad \eta, \xi \in T_x S^{n-1} \text{ with } \|\eta\|_x, \|\xi\|_x < 1,\ x \in S^{n-1}. \tag{3.5.11}$$

For this $\mathcal{T}^R$, the norm $\|\mathcal{T}^R_\eta(\xi)\|_{R_x(\eta)}$ is evaluated as

$$\|\mathcal{T}^R_\eta(\xi)\|^2_{R_x(\eta)} = \|\xi\|^2_x + \frac{(\eta^T\xi)^2}{1-\|\eta\|^2_x} \geq \|\xi\|^2_x, \tag{3.5.12}$$

where use has been made of $x^T x = 1$ and $x^T \xi = 0$. Thus, the inequality (3.3.4), which is the key condition for the proof of the global convergence property of Algorithm 3.3.1, is violated unless $\eta_k = 0$. In spite of this fact, we may try to perform Algorithm 3.3.1 for this problem. If the generated sequence does not diverge, we can compare the result with that obtained by Algorithm 3.3.2. We performed Algorithms 3.3.1 and 3.3.2 and obtained Fig. 3.5.8, whose vertical axis carries $\sqrt{(x_k - x_*)^T(x_k - x_*)}$. The figure shows the superiority of the proposed algorithm.

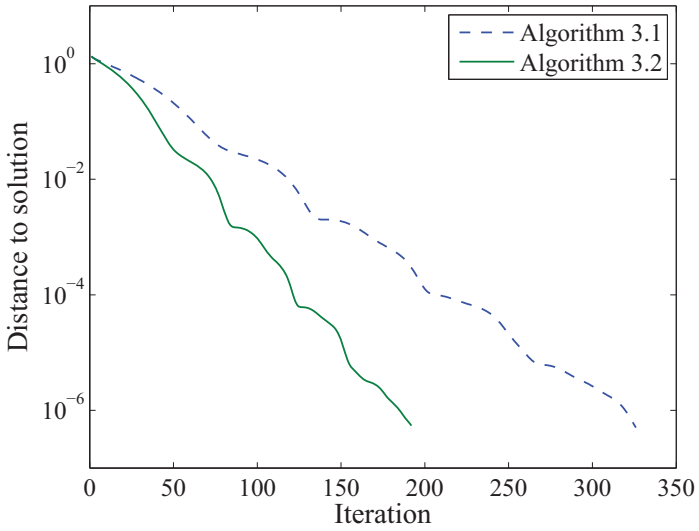



Figure 3.5.8: The sequences of distances between $x_k$ and $x_*$ for the sequences $\{x_k\}$ generated by Algorithms 3.3.1 and 3.3.2 with the orthographic retraction.

## 3.6 Summary

We have dealt with the global convergence of the conjugate gradient method with the Fletcher-Reeves $\beta$. Though the conjugate gradient method generates globally converging sequences in the Euclidean space, the conjugate gradient method on a Riemannian manifold $M$ has not been shown to have a convergence property in general, but under the assumption that the vector transport $\mathcal{T}^R$ as the differentiated retraction does not increase the norm of the tangent vector, the convergence is proved in [RW12]. If the parallel translation is adopted as a vector transport, the conjugate gradient method is shown to generate converging sequences, as is given in [Smi94]. However, the parallel translation is not convenient for computational effectiveness. For computational efficiency, we have introduced a vector transport, in place of the parallel translation, with a modification that the vector transport $\mathcal{T}^R$ is replaced by the scaled vector transport $\mathcal{T}^0$ only when $\mathcal{T}^R$ increases the norm of the search direction vector. The idea is simple but effective. We have achieved a balance between computational efficiency and the global convergence by proposing Algorithm 3.3.2. We have shown the convergence of the present algorithm both in the theoretical and the numerical viewpoints. In particular, we have performed numerical experiments to show that the present algorithm can solve problems for which the existing algorithm cannot work well because of the violation of the assumption about the vector transport.

## 3.7 Appendix: Examples in which the condition (3.4.2) holds

In Thm. 3.4.1, we assume that the condition (3.4.2) holds. We here compare (3.4.2) with the condition that $f \circ R_x$ is Lipschitz continuously differentiable uniformly for $x$, that is, there exists a Lipschitz constant $L > 0$ such that

$$\|\mathrm{D}(f \circ R_x)(\xi) - \mathrm{D}(f \circ R_x)(\zeta)\| \leq L\|\xi - \zeta\|_x, \qquad \xi, \zeta \in T_xM, x \in M, \tag{3.7.1}$$

where the $\|\cdot\|$ of the left-hand side means the operator norm (see [RW12] for detail). The condition (3.7.1) is equivalent to

$$\sup_{\|\eta\|_x=1} |(\mathrm{D}(f \circ R_x)(\xi) - \mathrm{D}(f \circ R_x)(\zeta))[\eta]| \leq L\|\xi - \zeta\|_x, \qquad \xi, \zeta \in T_xM, x \in M. \tag{3.7.2}$$

In particular, setting $\zeta = 0$ and $\xi = t\eta$ in (3.7.2) yields (3.4.2). In this sense, the condition (3.4.2) is a weaker form of (3.7.1). The assumption (3.4.2) is of practical use. For example, the problem of minimizing the Brockett cost function on the Stiefel manifold $\mathrm{St}(p, n)$ with the natural induced metric [AMS08] has this property, as is shown below.

Let $n, p$ be positive integers with $n \geq p$. The Stiefel manifold $\mathrm{St}(p, n)$ is defined to be $\mathrm{St}(p, n) := \left\{X \in \mathbb{R}^{n\times p} \,|\, X^TX = I_p\right\}$. We consider $\mathrm{St}(p, n)$ as a Riemannian submanifold of $\mathbb{R}^{n\times p}$ endowed with the natural induced metric

$$\langle \xi, \eta \rangle_X := \mathrm{tr}(\xi^T\eta), \qquad \xi, \eta \in T_X\mathrm{St}(p, n). \tag{3.7.3}$$

Let $A$ be an $n \times n$ symmetric matrix and $N := \mathrm{diag}(\mu_1, \mu_2, \ldots, \mu_p)$ with $0 < \mu_1 < \mu_2 < \cdots < \mu_p$. The Brockett cost function $f$ is defined on $\mathrm{St}(p, n)$ to be

$$f(X) = \mathrm{tr}\left(X^TAXN\right). \tag{3.7.4}$$

Further, the QR decomposition-based retraction (which we call the QR retraction) $R$ is defined to be

$$R_X(\xi) := \mathrm{qf}(X + \xi), \qquad \xi \in T_X\mathrm{St}(p, n),\ X \in \mathrm{St}(p, n), \tag{3.7.5}$$

where $\mathrm{qf}(B)$ denotes the Q-factor of the QR decomposition of a full rank matrix $B \in \mathbb{R}^{n\times p}$. That is, if $B$ is decomposed into $B = QR$, where $Q \in \mathrm{St}(p, n)$ and $R$ is an upper triangular $p \times p$ matrix with positive diagonal elements, then $\mathrm{qf}(B) = Q$.

**Proposition 3.7.1.** *The inequality* (3.4.2) *holds for the Brockett cost function* (3.7.4) *on* $M = \mathrm{St}(p, n)$*, where* $\mathrm{St}(p, n)$ *is endowed with the natural induced metric* (3.7.3)*, and where the QR retraction* (3.7.5) *is adopted.*

*Proof.* Since the function (3.7.4) is smooth, we have only to show that

$$\left|\frac{d^2}{dt^2}\left(f \circ R_X\right)(t\eta)\right| \leq L, \qquad \eta \in T_X\mathrm{St}(p,n) \text{ with } \|\eta\|_X = 1,\ X \in \mathrm{St}(p,n),\ t \geq 0. \quad (3.7.6)$$

In fact, Eq. (3.4.2) is a straightforward consequence of this inequality. Let $Q(t)$ be a curve defined by $R_X(t\eta) = \mathrm{qf}(X + t\eta)$, and $x_k, \eta_k, q_k(t)$ denote the $k$-th column vectors of $X, \eta, Q(t)$, respectively. Then, through the Gram-Schmidt orthonormalization process, we obtain

$$q_k(t) = \frac{x_k + t\eta_k - \sum_{i=1}^{k-1}(q_i(t), x_k + t\eta_k)q_i(t)}{\|x_k + t\eta_k - \sum_{i=1}^{k-1}(q_i(t), x_k + t\eta_k)q_i(t)\|}, \quad (3.7.7)$$

where $(a,b) := a^T b$ and $\|a\| := \sqrt{(a,a)}$ for $n$-dimensional vectors $a, b$. By induction on $k$, we can take vector-valued polynomials $g_k(t)$ in $t$ satisfying

$$q_k(t) = \frac{g_k(t)}{\|g_k(t)\|}, \qquad t \geq 0. \quad (3.7.8)$$

Indeed, for $k = 1$, (3.7.8) holds with $g_1(t) = x_1 + t\eta_1$. Suppose that (3.7.8) holds for $1, \ldots, k-1$. Then we can write out $q_k(t)$ as

$$q_k(t) = \frac{\prod_{j=1}^{k-1}\|g_j(t)\|^2(x_k + t\eta_k) - \sum_{i=1}^{k-1}\prod_{j\neq i}\|g_j(t)\|^2(g_i(t), x_k + t\eta_k)g_i(t)}{\|\prod_{j=1}^{k-1}\|g_j(t)\|^2(x_k + t\eta_k) - \sum_{i=1}^{k-1}\prod_{j\neq i}\|g_j(t)\|^2(g_i(t), x_k + t\eta_k)g_i(t)\|}. \quad (3.7.9)$$

Denoting by $g_k(t)$ the numerator of the right-hand side of (3.7.9), which is a polynomial in $t$, we obtain (3.7.8).

Let

$$h(X, \eta, t) = \frac{d^2}{dt^2}(f \circ R_X)(t\eta). \quad (3.7.10)$$

Then, the $h(X, \eta, t)$ is written out as

$$h(X, \eta, t) = \sum_{k=1}^{p} \mu_k \frac{d^2}{dt^2}\left(q_k(t)^T A q_k(t)\right). \quad (3.7.11)$$

Since $q_k(t)^T A q_k(t) = g_k(t)^T A g_k(t)/\|g_k(t)\|^2$, and since the degree of the numerator polynomial in $t$ is not more than that of the denominator polynomial, the degree of the numerator polynomial from the right-hand side of (3.7.11) is less than that of the denominator polynomial, so that one has, as $t \to \infty$,

$$\lim_{t\to\infty} h(X, \eta, t) = 0. \quad (3.7.12)$$

This implies that $h(X, \eta, t)$ is bounded with respect to $t \geq 0$. Moreover, the $h(X, \eta, t)$ is

continuous with respect to $X$ and $\eta$ on the compact set $\{(X,\eta) \in T\,\mathrm{St}(p,n) \mid \|\eta\|_X = 1\}$. It then turns out that $h(X,\eta,t)$ is bounded on the whole domain, which implies that there exists $L > 0$ such that (3.7.6) holds. This completes the proof. □

**Remark 3.7.1.** *Reviewing the proof, we observe that since the QR retraction is irrespective of the metric with which the* $\mathrm{St}(p,n)$ *is endowed, and since the set* $\{(X,\eta) \in T\,\mathrm{St}(p,n) \mid \|\eta\|_X = 1\}$ *is compact with respect to any metric on* $\mathrm{St}(p,n)$*, the inequality* (3.4.2) *with* $R$ *being the QR retraction* (3.7.5) *holds for the Brockett cost function* (3.7.4) *independently of the choice of a metric.*

**Remark 3.7.2.** *We also note that Prop. 3.7.1 and Rem. 3.7.1 cover both the Rayleigh quotient on the sphere* $S^{n-1}$ *as* $p = 1$ *and the Brockett cost function on the orthogonal group as* $p = n$*. In particular, the inequality* (3.4.2) *holds for the function* (3.5.1)*, though the sphere* $S^{n-1}$ *is endowed with the non-standard metric* (3.5.3)*.*

Another example for (3.4.2) comes from the problem of maximizing the function

$$F(U,V) = \mathrm{tr}(U^T A V N) \tag{3.7.13}$$

on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$, where $A$ is an $m \times n$ matrix and $N = \mathrm{diag}(\mu_1, \ldots, \mu_p)$ with $\mu_1 > \cdots > \mu_p > 0$ (see Chapter 4). An optimal solution to this problem gives the singular value decomposition of $A$. Let $m, n, p$ be positive integers with $m \geq n \geq p$. We consider $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ as a Riemannian submanifold of $\mathbb{R}^{m\times p} \times \mathbb{R}^{n\times p}$ endowed with the natural induced metric;

$$\begin{aligned} &\langle (\xi_1,\eta_1), (\xi_2,\eta_2) \rangle_{(U,V)} := \mathrm{tr}(\xi_1^T \xi_2) + \mathrm{tr}(\eta_1^T \eta_2), \\ &(\xi_1,\eta_1), (\xi_2,\eta_2) \in T_{(U,V)}(\mathrm{St}(p,m) \times \mathrm{St}(p,n))\,. \end{aligned} \tag{3.7.14}$$

As in the previous example on $\mathrm{St}(p,n)$, the QR retraction on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ is defined by

$$R_{(U,V)}(\xi,\eta) := (\mathrm{qf}(U+\xi), \mathrm{qf}(V+\eta))\,, \qquad (\xi,\eta) \in T_{(U,V)}(\mathrm{St}(p,m) \times \mathrm{St}(p,n)) \tag{3.7.15}$$

for $(U,V) \in \mathrm{St}(p,m) \times \mathrm{St}(p,n)$.

**Proposition 3.7.2.** *The inequality* (3.4.2) *holds for the objective function* (3.7.13) *on* $M = \mathrm{St}(p,m) \times \mathrm{St}(p,n)$*, where* $M$ *is endowed with the natural induced metric* (3.7.14) *and with the QR retraction* (3.7.15)*.*

*Proof.* We shall show that

$$\left| \frac{d^2}{dt^2} \left( F \circ R_{(U,V)} \right) (t(\xi,\eta)) \right| \leq L \tag{3.7.16}$$

for $(\xi,\eta)\in T_{(U,V)}(\mathrm{St}(p,m)\times\mathrm{St}(p,n))$ with $\|(\xi,\eta)\|_{(U,V)}=1$, $(U,V)\in\mathrm{St}(p,m)\times\mathrm{St}(p,n)$, $t\geq 0$. Put $Q(t)=\mathrm{qf}(U+t\xi)$, $S(t)=\mathrm{qf}(V+t\eta)$. Let $q_k(t)$ and $s_k(t)$ denote the $k$-th column vectors of $Q(t)$ and $S(t)$, respectively. From Prop. 3.7.1 and its course of the proof, there exist vector-valued polynomials $g_k(t)$ and $h_k(t)$ such that

$$q_k(t)=\frac{g_k(t)}{\|g_k(t)\|},\ s_k(t)=\frac{h_k(t)}{\|h_k(t)\|}. \tag{3.7.17}$$

Let

$$H(U,V,\xi,\eta,t)=\frac{d^2}{dt^2}\left(F\circ R_{(U,V)}\right)(t(\xi,\eta)). \tag{3.7.18}$$

Then we have

$$H(U,V,\xi,\eta,t)=\sum_{k=1}^{p}\mu_k\frac{d^2}{dt^2}\left(q_k(t)^TAs_k(t)\right). \tag{3.7.19}$$

Since $q_k(t)^TAs_k(t)=g_k(t)^TAh_k(t)/(\|g_k(t)\|\|h_k(t)\|)$, by the same reasoning as that for $h(X,\xi,t)$ in Prop. 3.7.1, we have

$$\lim_{t\to\infty}H(U,V,\xi,\eta,t)=0, \tag{3.7.20}$$

so that $H(U,V,\xi,\eta,t)$ is bounded with respect to $t\geq 0$. Further, $H(U,V,\xi,\eta,t)$ is continuous with respect to $(U,V,\xi,\eta)$ on the compact set
$\left\{(U,V,\xi,\eta)\in T\left(\mathrm{St}(p,m)\times\mathrm{St}(p,n)\right)\mid\|(\xi,\eta)\|_{(U,V)}=1\right\}$. Hence $H(U,V,\xi,\eta,t)$ is bounded on the whole domain. This completes the proof. □

A remark similar to Rem. 3.7.1 can be made on the metric to be endowed with on $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$. The validity of (3.4.2) is independent of the choice of a metric.

We here note that Prop. 3.7.2 together with Thm. 3.4.2 ensures that Algorithm 3.3.2 for the problem of maximizing $F$ (see Problems 4.2.1 and 4.2.2 in Chapter 4) has a global convergence property.

Returning to the case of a general Riemannian manifold $M$, we make a further comment on (3.4.2). We are interested in the range of $t\geq 0$. Assume that $M$ is compact and $f$ is smooth. A smooth function on a compact set is Lipschitz continuously differentiable. However, the set $\{(x,\eta,t)\in TM\times\mathbb{R}\mid\|\eta\|_x=1,t\geq 0\}$ is not compact even though $M$ is compact. Therefore, it is not so clear that the inequality (3.4.2) holds in general. We here note that the inequality (3.4.2) is used in the form

$$\mathrm{D}(f\circ R_{x_k})(\alpha_k\eta_k)[\eta_k]-\mathrm{D}(f\circ R_{x_k})(0)[\eta_k]\leq\alpha_kL\|\eta_k\|_{x_k}^2 \tag{3.7.21}$$

for the proof of Thm. 3.4.1. A question then arises as to under what condition the inequality (3.7.21) holds. If it is ensured that there exists a constant $m>0$ such that $\alpha_k\|\eta_k\|_{x_k}\leq m$ for all $k$, then we can prove (3.7.21). Indeed, in order to prove (3.7.21) in such a case, the range of $t$ in (3.4.2) can be restricted to $0\leq t\leq m$, and the inequality we need to prove

as a counterpart to (3.4.2) is written as

$$|\mathrm{D}(f \circ R_x)(t\eta)[\eta] - \mathrm{D}(f \circ R_x)(0)[\eta]| \leq Lt, \quad \eta \in T_xM \text{ with } \|\eta\|_x = 1,\ x \in M,\ 0 \leq t \leq m. \tag{3.7.22}$$

In order that (3.7.22) hold, it is sufficient that there exists a constant $L > 0$ satisfying

$$\left| \frac{d^2}{dt^2} \left( f \circ R_x \right)(t\eta) \right| \leq L, \qquad \eta \in T_xM \text{ with } \|\eta\|_x = 1,\ x \in M,\ 0 \leq t \leq m. \tag{3.7.23}$$

Since the left-hand side of the inequality (3.7.23) is continuous with respect to $t$ on a compact set $\{t \in \mathbb{R} \,|\, 0 \leq t \leq m\}$, there exists $L_{x,\eta}$ for each $(x, \eta) \in \mathcal{M}$ such that (3.7.23) with $L = L_{x,\eta}$ holds, where $\mathcal{M} = \{(x, \eta) \in TM \,|\, \|\eta\|_x = 1\}$. The compactness of the set $\mathcal{M}$ ensures the existence of $L := \sup_{(x,\eta)\in\mathcal{M}} L_{x,\eta}$ and the $L$ thus defined satisfies (3.7.23).

# Chapter 4

# A Riemannian Optimization Approach to the Matrix Singular Value Decomposition

## 4.1 Introduction

The truncated singular value decomposition, which is composed of the $p\,(\leq \min\{m,n\})$ dominant singular values and the associated vectors, of an $m \times n$ matrix $A$ can be put as an optimization problem of maximizing the objective function $\mathrm{tr}\left(U^T A V N\right)$ of $U \in \mathbb{R}^{m\times p}$ and $V \in \mathbb{R}^{n\times p}$ subject to the condition that $U^T U = V^T V = I_p$, where $N \in \mathbb{R}^{p\times p}$ is a constant diagonal matrix. The orthogonal constraints lead to the concept of the Stiefel manifold $\mathrm{St}(p,n) = \left\{Y \in \mathbb{R}^{n\times p} \,|\, Y^T Y = I_p\right\}$. Then, the constraints prove to be equivalent to $(U,V) \in \mathrm{St}(p,m) \times \mathrm{St}(p,n)$. Thus, the problem is set up on the Riemannian manifold $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ without constraints.

Unconstrained optimization methods on the Euclidean space, such as the steepest descent, the conjugate gradient, and Newton's methods are generalized to those on a Riemannian manifold. This chapter deals with optimization algorithms on the product manifold $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ to solve the singular value decomposition problem from this point of view. Though Newton's method on this manifold generates quadratically convergent sequences, the convergence domain for an optimal solution is restricted to a neighborhood of the target solution. If a good approximation of the singular value decomposition of a matrix is obtained, then Newton's method is performed to obtain more accurate singular value decomposition quickly.

The organization of this chapter is as follows: In Section 4.2, the singular value decomposition of a rectangular matrix $A$ is formulated as an optimization problem on $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$. The fact that the optimization problem is indeed equivalent to the truncated singular value decomposition problem is proved via the Lagrange multiplier method. Section 4.3 is concerned with the geometry of the product manifold $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$. Re-

tractions and the gradient and the Hessian of the objective function for the singular value decomposition problem are set up on $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$, which will be used in describing algorithms in later sections. The steepest descent, the conjugate gradient, and Newton's methods for the objective function on $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$ are described in Section 4.4. As is expected, the steepest descent method does not generate quickly convergent sequences. The conjugate gradient method generates sequences converging much more quickly than those generated by the steepest descent method. Newton's method generates the most quickly converging sequences among the three methods, but the sequences do not necessarily converge to global optimal solutions. In addition, Newton's equation for the present problem is practically difficult to solve unless $p=1$, but it is feasible in practice if $p=1$. In Section 4.5, Newton's method with $p=1$ and the conjugate gradient method are put together at first to provide a new Riemannian optimization approach on $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$ to the singular value decomposition. The problem of solving Newton's equation with $p\neq 1$ can be divided into a set of the problems with $p=1$, if suitable initial data are given. In view of this, for the problem with $p\neq 1$, the conjugate gradient method is combined with the set of Newton's methods with $p=1$ to provide a new algorithm for the singular value decomposition. Numerical experiments with these algorithms are performed for a matrix with $m=500,\ n=300,\ p=10$, which show that the last-stated method achieves the highest efficiency. Aside from the present method, Newton's method can be combined with existing algorithms. For example, when the singular value decomposition obtained by MATLAB's svd function is set as an initial decomposition, Newton's method serves to generate a sequence converging to a global optimal solution. Put another way, the MATLAB solution is improved by the present Newton's method. Degenerate optimal solutions are studied in Section 4.6 to show that those solutions form a submanifold diffeomorphic to the product of orthogonal groups and Stiefel manifolds of smaller dimension. It then turns out that according to whether the singular values are distinct or degenerate the optimal solution set is a discrete finite set or a disconnected submanifold. Section 4.7 contains some remarks on the present results.

## 4.2 The singular value decomposition and a Riemannian optimization problem

For an $m\times n$ matrix $A$ with $m\geq n$, the singular value decomposition of $A$ takes the form

$$A=U_0\Sigma_0V_0^T,\qquad U_0\in O(m),\ V_0\in O(n),\ \Sigma_0=\begin{pmatrix}\Sigma_1\\ \\ 0\end{pmatrix},\tag{4.2.1}$$

where $\Sigma_1 = \mathrm{diag}\,(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$, and where $\sigma_i,\ i = 1, \ldots, n$ are called the singular values of $A$ [GVL12, TBI97]. Let $u_1, \ldots, u_m$ and $v_1, \ldots, v_n$ denote the columns of $U_0$ and $V_0$ from the left, respectively; $U_0 = (u_1, \ldots, u_m)$, $V_0 = (v_1, \ldots, v_n)$. The corresponding columns $u_i$ and $v_i$ of $U_0$ and $V_0$ are called the left and right singular vectors of $A$, respectively. The singular value decomposition of $A$ is also expressed in terms of $u_i$ and $v_i$ as

$$A = \sum_{i=1}^{n} \sigma_i u_i v_i^T. \tag{4.2.2}$$

In this equation, $u_{n+1}, \ldots, u_m$ do not appear. Thus, for $U_1 = (u_1, \ldots, u_n)$ and $V_1 = V_0$, we rewrite Eq. (4.2.2) as

$$A = U_1 \Sigma_1 V_1^T. \tag{4.2.3}$$

The decomposition (4.2.3) is called the thin [GVL12] or the reduced [TBI97] singular value decomposition.

Like the Rayleigh quotient associated with the eigenvalue problem for a symmetric matrix [AMS08, EAS98, HM94], the following optimization problem is closely related to the singular value decomposition of a rectangular matrix.

**Problem 4.2.1.**

$$\text{maximize} \quad \mathrm{tr}\left(U^T A V N\right), \tag{4.2.4}$$

$$\text{subject to} \quad U \in \mathbb{R}^{m \times p},\ V \in \mathbb{R}^{n \times p},\ U^T U = V^T V = I_p, \tag{4.2.5}$$

where $N = \mathrm{diag}(\mu_1, \ldots, \mu_p)$ with $\mu_1 > \cdots > \mu_p > 0$ and $1 \leq p \leq n$.

A global optimal solution to Problem 4.2.1 provides a collection of $p$ dominant left and right singular vectors of $A$.

**Proposition 4.2.1.** *Let $(U_*, V_*)$ be a global optimal solution to Problem 4.2.1 for an $m \times n$ matrix $A$ with $m \geq n$. Then, the columns of $U_*$ and of $V_*$ are a collection of $p$ dominant left and right singular vectors of $A$, respectively.*

To prove this proposition, we start with the following lemma.

**Lemma 4.2.1.** *Let $C$ and $D$ be $n \times n$ mutually commuting matrices. Assume that $D$ takes the diagonal matrix form $D = \mathrm{diag}(d_1, \ldots, d_n)$ with $d_1, \ldots, d_n$ being mutually distinct. Then, the matrix $C$ is also diagonal.*

*Proof.* Denoting the $(i, j)$ component of $C$ by $c_{ij}$, we obtain $(CD)_{ij} = c_{ij} d_j$ and $(DC)_{ij} = c_{ij} d_i$. Then the commutativity condition $CD = DC$ provides $c_{ij}\,(d_i - d_j) = 0$. Since $d_1, \ldots, d_n$ are all distinct, we have $c_{ij} = 0$ for $i \neq j$. This completes the proof. □

We proceed to the proof of Prop.4.2.1.

*Proof of Prop. 4.2.1.* We take the Lagrange multiplier method for Problem 4.2.1. Let $L(U, V, \Lambda, \Omega)$ be the function defined by

$$L(U, V, \Lambda, \Omega) = \mathrm{tr}\left(U^T A V N\right) + \mathrm{tr}\left(\Lambda\left(U^T U - I_p\right)\right) + \mathrm{tr}\left(\Omega\left(V^T V - I_p\right)\right), \tag{4.2.6}$$

where $\Lambda$ and $\Omega$ are Lagrange multipliers, and where they should be symmetric matrices on account of the fact that $U^T U - I_p$ and $V^T V - I_p$ are symmetric. Let $L_U$ and $L_V$ denote the partial derivatives of $L$ with respect to $U$ and $V$, respectively. Put another way, $L_U$ is the $m \times p$ matrix whose $(i, j)$ component is $\partial L(U, V, \Lambda, \Omega)/\partial U_{ij}$ for example. Performing the derivation with respect to $U$, we obtain

$$L_U = AVN + 2U\Lambda. \tag{4.2.7}$$

Similarly, we obtain the expressions of $L_V, L_\Lambda$, and $L_\Omega$ as

$$L_V = A^T U N + 2V\Omega, \qquad L_\Lambda = U^T U - I_p, \qquad L_\Omega = V^T V - I_p. \tag{4.2.8}$$

Let $\Lambda_*$ and $\Omega_*$ be Lagrange multipliers corresponding to a global optimal solution $(U_*, V_*)$. It then follows from (4.2.7) and (4.2.8) that

$$AV_*N + 2U_*\Lambda_* = 0, \tag{4.2.9}$$
$$A^T U_* N + 2V_*\Omega_* = 0, \tag{4.2.10}$$
$$U_*^T U_* = V_*^T V_* = I_p. \tag{4.2.11}$$

Multiplying Eq. (4.2.9) by $U_*^T$ from the left, we have

$$\Lambda_* = -\frac{1}{2} U_*^T A V_* N. \tag{4.2.12}$$

Similarly, we have from Eq. (4.2.10)

$$\Omega_* = -\frac{1}{2} V_*^T A^T U_* N. \tag{4.2.13}$$

Substituting Eqs. (4.2.12) and (4.2.13) into Eqs. (4.2.9) and (4.2.10), respectively, and multiplying the resultant equations by $N^{-1}$ from the right, we have

$$AV_* = U_* U_*^T A V_*, \tag{4.2.14}$$
$$A^T U_* = V_* V_*^T A^T U_*. \tag{4.2.15}$$

Since $\Lambda$ and $\Omega$ are symmetric, we obtain from (4.2.12) and (4.2.13)

$$U_*^T A V_* N = N V_*^T A^T U_*, \tag{4.2.16}$$

$$V_*^T A^T U_* N = N U_*^T A V_*, \tag{4.2.17}$$

respectively. These two equations are put together to provide

$$U_*^T A V_* N^2 = N^2 U_*^T A V_*. \tag{4.2.18}$$

Since $N^2$ is a diagonal matrix with mutually distinct diagonal entries, Eq. (4.2.18) implies that $U_*^T A V_*$ is also diagonal on account of Lemma 4.2.1. Since $U_*^T A V_*$ is diagonal, it is a symmetric matrix, so that $U_*^T A V_* = V_*^T A^T U_*$. Let $U_*^T A V_* = V_*^T A^T U_* = \mathrm{diag}\,(s_1, \ldots, s_p)$ and $U_* = (u_1, \ldots, u_p)\,, V_* = (v_1, \ldots, v_p)$. Then, Eqs. (4.2.14) and (4.2.15) take the form

$$A v_i = s_i u_i, \quad A^T u_i = s_i v_i, \qquad i = 1, \ldots, p, \tag{4.2.19}$$

respectively. The objective function is then evaluated at $(U_*, V_*)$ as

$$\mathrm{tr}\left(U_*^T A V_* N\right) = \sum_{i=1}^{p} s_i \mu_i, \tag{4.2.20}$$

where $\mu_1 > \cdots > \mu_p > 0$. Since $(U, V) = (U_*, V_*)$ is a maximizer, we can conclude that $s_1 \geq \cdots \geq s_p \geq 0$. Further, Eq. (4.2.19) implies that

$$A^T A v_i = s_i A^T u_i = s_i^2 v_i. \tag{4.2.21}$$

This means that $s_i^2$ and $v_i$ are an eigenvalue and the corresponding eigenvector of $A^T A$, respectively. Therefore, $s_i$ and $v_i$ are the $i$-th dominant singular value and the corresponding right singular vector for each $i = 1, \ldots, p$. Similarly, $u_i$ proves to be the left singular vector associated with $s_i$ for each $i = 1, \ldots, p$. This completes the proof. □

We make a remark on Eq. (4.2.20). In the course of deriving Eq. (4.2.20), we have only required that the objective function takes a critical value. If we do not require that $(U_*, V_*)$ is a maximizer, we do not have to put the singular values in the order $s_1 \geq s_2 \geq \cdots \geq s_p$. In particular, for $p = 1$, the objective function takes the value $\mathrm{tr}\left(U_*^T A V_* N\right) = \mu_1 s_1$, which is a multiple of one of the singular values. We will use this fact in Section 4.5. We also note that global optimal solutions to Problem 4.2.1 form a finite or an infinite set, as will be seen in Section 4.6.

A Stiefel manifold is defined to be $\mathrm{St}(p, n) = \left\{Y \in \mathbb{R}^{n \times p} \,|\, Y^T Y = I_p\right\}$. On account of the constraint (4.2.5), the set of all feasible points of Problem 4.2.1 is the product manifold $\mathrm{St}(p, m) \times \mathrm{St}(p, n)$. In the optimization theory, a maximization problem is often converted into a minimization problem. We shall work with the following minimization problem equivalent to Problem 4.2.1 in what follows.

**Problem 4.2.2.**

$$\begin{aligned} \text{minimize} \quad & F(U,V) = -\operatorname{tr}\left(U^T A V N\right), && (4.2.22)\\ \text{subject to} \quad & (U,V) \in \mathrm{St}(p,m) \times \mathrm{St}(p,n). && (4.2.23) \end{aligned}$$

This is a Riemannian optimization problem on $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$. A review of optimization techniques on a generic Riemannian manifold is given in Section 2.2.

## 4.3 The Riemannian geometry of $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$

We deal with the Riemannian geometry of $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ for the purpose of our optimization problem. For the Riemannian geometry of $\mathrm{St}(p,n)$, see Section 2.3 and [AMS08, EAS98].

### 4.3.1 Tangent spaces

Since the tangent space $T_Y\mathrm{St}(p,n)$ at $Y \in \mathrm{St}(p,n)$ is expressed as

$$T_Y\mathrm{St}(p,n) = \left\{\xi \in \mathbb{R}^{n\times p} \,|\, \xi^T Y + Y^T \xi = 0\right\}, \tag{4.3.1}$$

the tangent space $T_{(U,V)}(\mathrm{St}(p,m) \times \mathrm{St}(p,n))$ at $(U,V) \in \mathrm{St}(p,m) \times \mathrm{St}(p,n)$ is written as

$$\begin{aligned} &T_{(U,V)}(\mathrm{St}(p,m) \times \mathrm{St}(p,n)) \simeq T_U\mathrm{St}(p,m) \times T_V\mathrm{St}(p,n)\\ &= \left\{(\xi,\eta) \in \mathbb{R}^{m\times p} \times \mathbb{R}^{n\times p} \,|\, \xi^T U + U^T \xi = \eta^T V + V^T \eta = 0\right\}. \end{aligned} \tag{4.3.2}$$

Since the $\mathrm{St}(p,n)$ is a submanifold of the matrix Euclidean space $\mathbb{R}^{n\times p}$, it can be endowed with the Riemannian metric

$$\langle \xi_1, \xi_2 \rangle_Y := \operatorname{tr}\left(\xi_1^T \xi_2\right), \qquad \xi_1, \xi_2 \in T_Y\mathrm{St}(p,n), \tag{4.3.3}$$

which is induced from the natural metric (Frobenius inner product) on $\mathbb{R}^{n\times p}$,

$$\langle B, C \rangle := \operatorname{tr}\left(B^T C\right), \qquad B, C \in \mathbb{R}^{n\times p}. \tag{4.3.4}$$

We view the product manifold $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ as a Riemannian submanifold of $\mathbb{R}^{m\times p} \times \mathbb{R}^{n\times p}$, which is endowed with the Riemannian metric

$$\begin{aligned} \left\langle (\xi_1,\eta_1), (\xi_2,\eta_2) \right\rangle_{(U,V)} &:= \langle \xi_1, \xi_2 \rangle_U + \langle \eta_1, \eta_2 \rangle_V = \operatorname{tr}\left(\xi_1^T \xi_2\right) + \operatorname{tr}\left(\eta_1^T \eta_2\right),\\ &(\xi_1,\eta_1), (\xi_2,\eta_2) \in T_{(U,V)}(\mathrm{St}(p,m) \times \mathrm{St}(p,n)). \end{aligned} \tag{4.3.5}$$

Using the metric thus defined, we give the expression of the orthogonal projection onto the tangent space $T_{(U,V)}(\mathrm{St}(p,m) \times \mathrm{St}(p,n))$. Since $T_U\mathrm{St}(p,m) \times T_V\mathrm{St}(p,n)$ is isomorphic

with $T_{(U,V)}\left(\mathrm{St}(p,m)\times\mathrm{St}(p,n)\right)$, the following proposition is easily verified.

**Proposition 4.3.1.** *For any $(B,C)\in\mathbb{R}^{m\times p}\times\mathbb{R}^{n\times p}$, the orthogonal projection operator $P_{(U,V)}$ onto the tangent space $T_{(U,V)}(\mathrm{St}(p,m)\times\mathrm{St}(p,n))$ at $(U,V)\in\mathrm{St}(p,m)\times\mathrm{St}(p,n)$ is given by*

$$P_{(U,V)}(B,C)=\left(P_U(B),P_V(C)\right), \tag{4.3.6}$$

*where*

$$P_U(B)=B-U\,\mathrm{sym}\left(U^TB\right),\ \ P_V(C)=C-V\,\mathrm{sym}\left(V^TC\right) \tag{4.3.7}$$

*are orthogonal projections onto $T_U\mathrm{St}(p,m)$ and $T_V\mathrm{St}(p,n)$, respectively, and where $\mathrm{sym}(B):=(B+B^T)/2$ denotes the symmetric part of $B$ [EAS98].*

### 4.3.2 Geodesics

**Proposition 4.3.2.** *Let $(U(t),V(t))$ be the geodesic on the product manifold $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$ emanating from $(U,V)\in\mathrm{St}(p,m)\times\mathrm{St}(p,n)$ in the direction of $(\xi,\eta)\in T_{(U,V)}(\mathrm{St}(p,m)\times\mathrm{St}(p,n))$. Then, the component matrices of $(U(t),V(t))$ are expressed as*

$$U(t)=\begin{pmatrix}U & \xi\end{pmatrix}\exp\left(t\begin{pmatrix}U^T\xi & -\xi^T\xi\\ I_p & U^T\xi\end{pmatrix}\right)\begin{pmatrix}I_p\\ 0\end{pmatrix}\exp\left(-tU^T\xi\right), \tag{4.3.8a}$$

$$V(t)=\begin{pmatrix}V & \eta\end{pmatrix}\exp\left(t\begin{pmatrix}V^T\eta & -\eta^T\eta\\ I_p & V^T\eta\end{pmatrix}\right)\begin{pmatrix}I_p\\ 0\end{pmatrix}\exp\left(-tV^T\eta\right), \tag{4.3.8b}$$

*respectively, where* exp *denotes the matrix exponential.*

*Proof.* Since the Riemannian metric (4.3.5) is the direct product of the metrics of $\mathrm{St}(p,m)$ and $\mathrm{St}(p,n)$, $(U(t),V(t))$ is a geodesic on $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$ if and only if $U(t)$ and $V(t)$ are geodesics on $\mathrm{St}(p,m)$ and on $\mathrm{St}(p,n)$, respectively. Since the right-hand sides of (4.3.8) are geodesics on $\mathrm{St}(p,m)$ and $\mathrm{St}(p,n)$ which emanate from $U$ and $V$ in the direction of $\xi$ and $\eta$, respectively (Prop. 2.3.4), the pair $(U(t),V(t))$ provides the geodesic on $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$. This completes the proof. □

We note here that a geodesic $Y(t)$ on the Stiefel manifold $\mathrm{St}(p,n)$ is a solution to the geodesic equation

$$\ddot{Y}(t)+Y(t)\dot{Y}(t)^T\dot{Y}(t)=0. \tag{4.3.9}$$

The pair $(U(t), V(t))$ is a geodesic on $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$, if and only if $U(t)$ and $V(t)$ satisfy

$$\ddot{U}(t) + U(t)\dot{U}(t)^T\dot{U}(t) = 0, \qquad \ddot{V}(t) + V(t)\dot{V}(t)^T\dot{V}(t) = 0, \tag{4.3.10}$$

respectively.

### 4.3.3 Retractions

The exponential map defined on a Riemannian manifold $M$ through geodesics emanating from each point in all directions determines a retraction on $M$. We call this map the exponential retraction. From Prop. 4.3.2, we can put the exponential retraction $R$ on $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$ in the form

$$\begin{aligned} &R_{(U,V)}(\xi,\eta) = \mathrm{Exp}_{(U,V)}(\xi,\eta) \\ &= \begin{pmatrix} \begin{pmatrix} U & \xi \end{pmatrix} \exp\left( \begin{pmatrix} U^T\xi & -\xi^T\xi \\ I_p & U^T\xi \end{pmatrix} \right) \begin{pmatrix} I_p \\ 0 \end{pmatrix} \exp\left(-U^T\xi\right), \\ \begin{pmatrix} V & \eta \end{pmatrix} \exp\left( \begin{pmatrix} V^T\eta & -\eta^T\eta \\ I_p & V^T\eta \end{pmatrix} \right) \begin{pmatrix} I_p \\ 0 \end{pmatrix} \exp\left(-V^T\eta\right) \end{pmatrix}, \end{aligned} \tag{4.3.11}$$

where $(\xi,\eta)\in T_{(U,V)}(\mathrm{St}(p,m)\times\mathrm{St}(p,n))$.

After the QR-based retraction on the single Stiefel manifold discussed in Subsection 2.3.3, we give another retraction on $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$ by means of the QR decomposition as follows:

**Proposition 4.3.3.** *Let $R_{(U,V)}$ be a map of $T_{(U,V)}(\mathrm{St}(p,m)\times\mathrm{St}(p,n))$ to $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$ defined at $(U,V)\in\mathrm{St}(p,m)\times\mathrm{St}(p,n)$ by*

$$R_{(U,V)}(\xi,\eta) = (\mathrm{qf}(U+\xi), \mathrm{qf}(V+\eta)), \qquad (\xi,\eta)\in T_{(U,V)}(\mathrm{St}(p,m)\times\mathrm{St}(p,n)), \tag{4.3.12}$$

*where the* qf *returns the Q factor of the QR decomposition of the matrix concerned (see the first part of Section 2.3). Then, the collection of $R_{(U,V)}$ for all $(U,V)\in\mathrm{St}(p,m)\times\mathrm{St}(p,n)$ forms a retraction $R: T(\mathrm{St}(p,m)\times\mathrm{St}(p,n))\to\mathrm{St}(p,m)\times\mathrm{St}(p,n)$.*

*Proof.* We first note that $(\mathrm{qf}(U+\xi), \mathrm{qf}(V+\eta))\in\mathrm{St}(p,m)\times\mathrm{St}(p,n)$. We then check two conditions in Definition 2.2.1. By the definition of the QR decomposition, the first condition is easily verified as

$$R_{(U,V)}(0,0) = (\mathrm{qf}(U), \mathrm{qf}(V)) = (U,V). \tag{4.3.13}$$

Since $\mathrm{D}\,\mathrm{qf}(Y)[\Delta] = \Delta$ for any $\Delta \in T_Y\mathrm{St}(p,n)$ from Prop. 2.3.1, we obtain

$$\mathrm{D}R_{(U,V)}(0,0)[(\xi,\eta)] = (\mathrm{D}\,\mathrm{qf}(U)[\xi], \mathrm{D}\,\mathrm{qf}(V)[\eta]) = (\xi,\eta). \tag{4.3.14}$$

This completes the proof. □

We call the $R$ defined through (4.3.12) the QR-based retraction on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$.

### 4.3.4 The gradient and the Hessian of the objective function

The gradient and the Hessian of an objective function are basic concepts in optimization methods. The gradient, $\mathrm{grad}\, F(U,V)$, of an objective function $F$ at $(U,V) \in \mathrm{St}(p,m) \times \mathrm{St}(p,n)$ is defined to be a unique tangent vector which satisfies

$$\langle \mathrm{grad}\, F(U,V), (\xi,\eta)\rangle_{(U,V)} = \mathrm{D}F(U,V)[(\xi,\eta)],\ (\xi,\eta) \in T_{(U,V)}(\mathrm{St}(p,m) \times \mathrm{St}(p,n))\,. \tag{4.3.15}$$

The Hessian, $\mathrm{Hess}\, F(U,V)$, of $F$ at $(U,V)$ is defined to be a linear transformation of the tangent space $T_{(U,V)}(\mathrm{St}(p,m) \times \mathrm{St}(p,n))$ through the covariant derivative $\nabla_{(\xi,\eta)}\,\mathrm{grad}\, F$ of $\mathrm{grad}\, F$ evaluated at $(U,V)$,

$$\mathrm{Hess}\, F(U,V)[(\xi,\eta)] := \nabla_{(\xi,\eta)}\,\mathrm{grad}\, F, \qquad (\xi,\eta) \in T_{(U,V)}(\mathrm{St}(p,m) \times \mathrm{St}(p,n))\,, \tag{4.3.16}$$

where the covariant derivative is defined through the Levi-Civita connection $\nabla$ on $\mathrm{St}(p,m)\times \mathrm{St}(p,n)$.

In what follows, we take up the objective function

$$F(U,V) = -\,\mathrm{tr}(U^T AVN). \tag{4.3.17}$$

**Proposition 4.3.4.** *The gradient of* (4.3.17) *at* $(U,V) \in \mathrm{St}(p,m) \times \mathrm{St}(p,n)$ *is expressed as*

$$\mathrm{grad}\, F(U,V) = \left(U\,\mathrm{sym}\left(U^T AVN\right) - AVN, V\,\mathrm{sym}\left(V^T A^T UN\right) - A^T UN\right). \tag{4.3.18}$$

*Proof.* Since $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ is a Riemannian submanifold of $\mathbb{R}^{m\times p} \times \mathbb{R}^{n\times p}$ endowed with the induced metric, $\mathrm{grad}\, F(U,V)$ is equal to the orthogonal projection of the Euclidean gradient $F_{(U,V)}$ of $F$ at $(U,V)$ onto $T_{(U,V)}(\mathrm{St}(p,m) \times \mathrm{St}(p,n))$. Hence, by using the projection $P_{(U,V)}$ given in (4.3.6) and (4.3.7), we obtain

$$\begin{aligned}
&\mathrm{grad}\, F(U,V) = P_{(U,V)}(F_{(U,V)}) = P_{(U,V)}\left(-AVN, -A^T UN\right)\\
&= \left(-P_U(AVN), -P_V(A^T UN)\right)\\
&= \left(U\,\mathrm{sym}\left(U^T AVN\right) - AVN, V\,\mathrm{sym}\left(V^T A^T UN\right) - A^T UN\right).
\end{aligned} \tag{4.3.19}$$

This completes the proof. □

**Proposition 4.3.5.** *Let $(\xi, \eta)$ be a tangent vector at $(U, V) \in \mathrm{St}(p, m) \times \mathrm{St}(p, n)$. Let $S_1$ and $S_2$ denote the matrices* $\mathrm{sym}\left(U^T A V N\right)$ *and* $\mathrm{sym}\left(V^T A^T U N\right)$*, respectively. The Hessian of* (4.3.17) *at $(U, V)$ is expressed as a linear map on $T_{(U,V)}(\mathrm{St}(p, m) \times \mathrm{St}(p, n))$ and given by*

$$\begin{aligned}\mathrm{Hess}\, F(U,V)[(\xi,\eta)] = \Big(&\xi S_1 - A\eta N - U\,\mathrm{sym}\left(U^T\left(\xi S_1 - A\eta N\right)\right),\\ &\eta S_2 - A^T\xi N - V\,\mathrm{sym}\left(V^T\left(\eta S_2 - A^T\xi N\right)\right)\Big). \end{aligned} \tag{4.3.20}$$

*Proof.* Let $(U(t), V(t))$ be the geodesic emanating from $(U(0), V(0)) = (U, V)$ in the direction of $(\dot{U}(0), \dot{V}(0)) = (\xi, \eta)$. Note that $U(t)$ and $V(t)$ satisfy Eq. (4.3.10) and hence

$$\ddot{U}(0) = -U\xi^T\xi, \qquad \ddot{V}(0) = -V\eta^T\eta. \tag{4.3.21}$$

Since $\langle \mathrm{Hess}\, F(U,V)[(\xi,\eta)], (\xi,\eta)\rangle_{(U,V)}$ is the covariant derivative of $\dfrac{d}{dt}F(U,V)$ at $t = 0$, as is seen from (4.3.16), and since $(U(t), V(t))$ is a geodesic, the quantity is written out as

$$\begin{aligned}\langle \mathrm{Hess}\, F(U,V)[(\xi,\eta)], (\xi,\eta)\rangle_{(U,V)} &= \frac{d^2}{dt^2}F\left(U(t), V(t)\right)\Big|_{t=0}\\ &= -\,\mathrm{tr}\left(\ddot{U}(0)^T A V(0) N + U(0)^T A \ddot{V}(0) N + 2\dot{U}(0)^T A \dot{V}(0) N\right)\\ &= \mathrm{tr}\left(\xi^T\xi U^T A V N + U^T A V \eta^T \eta N - 2\xi^T A \eta N\right). \end{aligned} \tag{4.3.22}$$

Since the Hessian operator is symmetric and linear on $T_{(U,V)}(\mathrm{St}(p, m) \times \mathrm{St}(p, n))$, the Hessian symmetric form in tangent vectors $(\xi, \eta)$ and $(\zeta, \chi)$ is expressed and written out as

$$\begin{aligned}&\langle \mathrm{Hess}\, F(U,V)[(\xi,\eta)], (\zeta,\chi)\rangle_{(U,V)}\\ =&\frac{1}{2}\Big(\langle \mathrm{Hess}\, F(U,V)[(\xi,\eta)+(\zeta,\chi)], (\xi,\eta)+(\zeta,\chi)\rangle_{(U,V)}\\ &\quad - \langle \mathrm{Hess}\, F(U,V)[(\xi,\eta)], (\xi,\eta)\rangle_{(U,V)} - \langle \mathrm{Hess}\, F(U,V)[(\zeta,\chi)], (\zeta,\chi)\rangle_{(U,V)}\Big)\\ =&\frac{1}{2}\,\mathrm{tr}\Big(\left(\xi^T\zeta + \zeta^T\xi\right)U^T A V N + U^T A V\left(\eta^T\chi + \chi^T\eta\right)N - 2\left(\xi^T A\chi + \zeta^T A\eta\right)N\Big)\\ =&\frac{1}{2}\,\mathrm{tr}\Big(\zeta^T\left(\xi N V^T A^T U + \xi U^T A V N - 2A\eta N\right) + \chi^T\left(\eta N U^T A V + \eta V^T A^T U N - 2A^T\xi N\right)\Big)\\ =&\,\mathrm{tr}\left(\zeta^T\left(\xi S_1 - A\eta N\right) + \chi^T\left(\eta S_2 - A^T\xi N\right)\right)\\ =&\left\langle P_{(U,V)}\left(\left(\xi S_1 - A\eta N\right), \left(\eta S_2 - A^T\xi N\right)\right), (\zeta,\chi)\right\rangle_{(U,V)}. \end{aligned} \tag{4.3.23}$$

Since the orthogonal projection operator $P_{(U,V)}$ is given by Eqs. (4.3.6) and (4.3.7), we have

$$\mathrm{Hess}\, F(U,V)[(\xi,\eta)] = P_{(U,V)}\left(\left(\xi S_1 - A\eta N\right), \left(\eta S_2 - A^T\xi N\right)\right)$$

$$=\Big(\xi S_1 - A\eta N - U\,\mathrm{sym}\left(U^T(\xi S_1 - A\eta N)\right), \eta S_2 - A^T\xi N - V\,\mathrm{sym}\left(V^T\left(\eta S_2 - A^T\xi N\right)\right)\Big). \tag{4.3.24}$$

This completes the proof. □

## 4.4 Optimization algorithms on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$

So far we have obtained requisites for optimization algorithms. In this section, we develop the steepest descent, the conjugate gradient, and Newton's methods for Problem 4.2.2.

### 4.4.1 The steepest descent method on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$

In the steepest descent method for a general Riemannian unconstrained optimization problem 2.2.1, the negative gradient of $f$ at a current iterate $x_k \in M$ is chosen as a search direction $\Delta_k \in T_{x_k}M$ at $x_k$, that is, $\Delta_k = -\operatorname{grad} f(x_k)$. Then, the updating formula is expressed as

$$x_{k+1} = R_{x_k}(t_k\Delta_k), \tag{4.4.1}$$

where $R$ is a retraction and $t_k$ is an Armijo step size.

In what follows, we specialize in the steepest descent method on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ for the objective function $F$ given in (4.3.17). From (4.3.18), the negative gradient of $F$ at $(U_k, V_k) \in \mathrm{St}(p,m) \times \mathrm{St}(p,n)$ is expressed as

$$-\operatorname{grad} F(U_k,V_k) = \left(AV_kN - U_k\,\mathrm{sym}\left(U_k^TAV_kN\right), A^TU_kN - V_k\,\mathrm{sym}\left(V_k^TA^TU_kN\right)\right). \tag{4.4.2}$$

With this expression taken into account, the algorithm for the steepest descent method for Problem 4.2.2 is described as follows:

**Algorithm 4.4.1** Steepest Descent Method for Problem 4.2.2

1: Choose an initial point $(U_0, V_0) \in \mathrm{St}(p,m) \times \mathrm{St}(p,n)$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Compute the search direction $(\xi_k, \eta_k) \in T_{(U,V)}\left(\mathrm{St}(p,m) \times \mathrm{St}(p,n)\right)$ by

$$\xi_k = AV_kN - U_k\,\mathrm{sym}\left(U_k^TAV_kN\right),\ \eta_k = A^TU_kN - V_k\,\mathrm{sym}\left(V_k^TA^TU_kN\right). \tag{4.4.3}$$

4: Compute the Armijo step size $t_k > 0$.
5: Compute the next iterate $(U_{k+1}, V_{k+1}) = R_{(U_k,V_k)}\left(t_k\left(\xi_k, \eta_k\right)\right)$, where $R$ is a retraction on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$.
6: **end for**

In the above algorithm, as is seen already, a possible choice for the retraction $R$ is the exponential retraction (4.3.11) or the QR-based retraction (4.3.12).

If the manifold in question is compact, a convergence result for the steepest descent method is stated in general as follows (see also Thm. 2.2.1):

**Theorem 4.4.1.** *Consider the problem of minimizing an objective function $f$ on a Riemannian manifold $M$. Let $\{x_k\}$ be an infinite sequence of iterates generated by the steepest descent method with the Armijo step size. If $M$ is compact, then*

$$\lim_{k\to\infty} \|\operatorname{grad} f(x_k)\|_{x_k} = 0. \tag{4.4.4}$$

Since the manifold $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$ is compact, the sequence generated by Algorithm 4.4.1 converges to a critical point of $F$.

A numerical experiment with Algorithm 4.4.1 is performed for $F$ with a $500\times 300$ matrix $A$ and the result is shown in Fig. 4.4.1, where the initial point is randomly chosen. The vertical axis of Fig. 4.4.1 carries the differences between the values $F(U_k,V_k)$ and the minimum value $F_{\min}$ of $F$. Here, we notice that because of the choice of a matrix

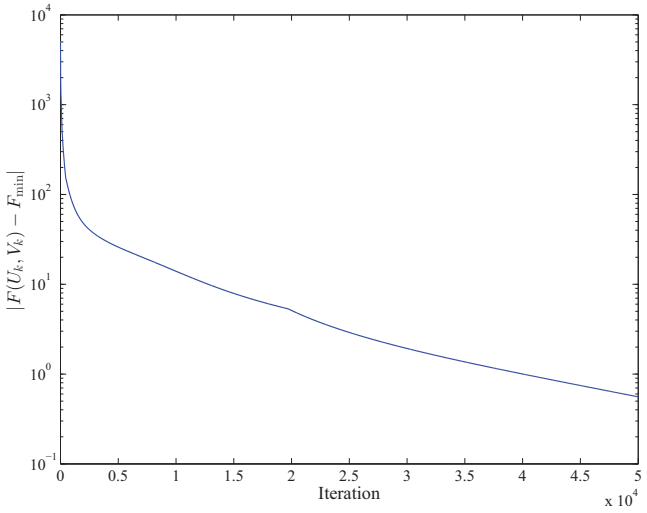



Figure 4.4.1: $m = 500$, $n = 300$, $p = 10$, $N = \operatorname{diag}(10,\dots,2,1)$, and $A = U_{\mathrm{r}}\operatorname{diag}(300,\dots,2,1)V_{\mathrm{r}}^T$, where $U_{\mathrm{r}}\in\mathbb{R}^{m\times n}$ and $V_{\mathrm{r}}\in\mathbb{R}^{n\times n}$ are orthonormal matrices with randomly chosen elements.

$A$, we know the optimal solution of this problem. This figure shows that the sequence $\{(U_k,V_k)\}$ is linearly convergent, as is expected, but the convergence is very slow, so that this algorithm is far from practical use for the present problem. We will treat a faster algorithm, the conjugate gradient method, in the next subsection.

### 4.4.2 The conjugate gradient method on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$

As was mentioned in Chapter 2, the conjugate gradient method on $\mathbb{R}^N$ was originally developed as a tool for solving linear systems of equations [HS52], and generalized to a nonlinear conjugate gradient method, which can be applied for a generic objective function [NW06], and further generalized to a similar method on a Riemannian manifold $M$ [Smi94, AMS08]. In [Smi94], the parallel translation along a geodesic on $M$ is used in computing the search direction. If we choose the exponential map as a retraction, the $\mathcal{T}_{\eta_x}(\xi_x)$ is realized as the parallel translation of $\xi_x$ along the geodesic segment $\mathrm{Exp}_x(t\eta_x)$ emanating from $x$ in the direction of $\eta_x$ with $0 \leq t \leq 1$ [Smi94]. In contrast with this, the vector transport, which is a generalization of the parallel translation, is introduced in [AMS08], for the sake of computational efficiency. For the definition of a vector transport, see Def. 2.2.2.

We can define vector transports $\mathcal{T}$ on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ by choosing the QR-based retraction:

**Proposition 4.4.1.** *Let $M = \mathrm{St}(p,m) \times \mathrm{St}(p,n)$. Define maps $\mathcal{T}^{\mathcal{R}}, \mathcal{T}^{\mathcal{P}} : TM \oplus TM \to TM$ by*

$$\begin{aligned}\mathcal{T}^R_{(\xi,\eta)}(\zeta,\chi) := &\left(Q_1 \rho_{\mathrm{skew}}\left(Q_1^T\zeta\left(Q_1^T(U+\xi)\right)^{-1}\right) + \left(I_m - Q_1Q_1^T\right)\zeta\left(Q_1^T(U+\xi)\right)^{-1},\right.\\ &\left. Q_2 \rho_{\mathrm{skew}}\left(Q_2^T\chi\left(Q_2^T(V+\eta)\right)^{-1}\right) + \left(I_n - Q_2Q_2^T\right)\chi\left(Q_2^T(V+\eta)\right)^{-1}\right), \quad (4.4.5)\end{aligned}$$

*and*

$$\mathcal{T}^P_{(\xi,\eta)}(\zeta,\chi) := \left(\zeta - Q_1\,\mathrm{sym}\left(Q_1^T\zeta\right), \chi - Q_2\,\mathrm{sym}\left(Q_2^T\chi\right)\right), \quad (4.4.6)$$

*respectively, where $(\xi,\eta),(\zeta,\chi) \in T_{(U,V)}\left(\mathrm{St}(p,m) \times \mathrm{St}(p,n)\right)$, and where $Q_1 := \mathrm{qf}(U+\xi), Q_2 := \mathrm{qf}(V+\eta)$. Then, $\mathcal{T}^R$ and $\mathcal{T}^P$ are vector transports on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$.*

*Proof.* For arbitrary vector transports $\mathcal{T}^{(1)}$ on $\mathrm{St}(p,m)$ and $\mathcal{T}^{(2)}$ on $\mathrm{St}(p,n)$, it is easily seen that the $\mathcal{T}$ defined by

$$\mathcal{T}_{(\xi,\eta)}(\zeta,\chi) := \left(\mathcal{T}^{(1)}_{\xi}(\zeta), \mathcal{T}^{(2)}_{\eta}(\chi)\right), \qquad (\xi,\eta),(\zeta,\chi) \in T_{(U,V)}\left(\mathrm{St}(p,m) \times \mathrm{St}(p,n)\right), \quad (4.4.7)$$

is a vector transport on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$. We can take the vector transports $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ as those given in Prop. 2.3.7 with appropriate sizes, completing the proof. □

In the nonlinear conjugate gradient method on the Euclidean space, the Wolfe step size is often used [NW06]. For the conjugate gradient method on a manifold $M$, we define the Wolfe point as follows:

**Definition 4.4.1.** *Let $f$ be an objective function on a Riemannian manifold $M$ with a retraction $R$. Given a point $x \in M$, a tangent vector $\Delta \in T_xM$, and scalars $\bar{\alpha} > 0$, $\beta$, $\sigma \in (0,1)$, the Wolfe point is determined to be $\Delta^W := \beta^m\bar{\alpha}\Delta$ in such a way that $m$ may be the*

*smallest nonnegative integer satisfying both*

$$f(x) - f(R_x(\beta^m \bar{\alpha}\Delta)) \geq -\sigma \langle \operatorname{grad} f(x), \beta^m \bar{\alpha}\Delta \rangle_x, \tag{4.4.8}$$

*and*

$$\langle \operatorname{grad} f\left(R_x(\beta^m \bar{\alpha}\Delta)\right), \mathcal{T}_{\beta^m \bar{\alpha}\Delta}(\Delta) \rangle_{R_x(\beta^m \bar{\alpha}\Delta)} \geq \rho \langle \operatorname{grad} f(x), \Delta \rangle_x, \tag{4.4.9}$$

*where* $0 < \sigma < \rho < 1$ *and* $\mathcal{T}$ *is a vector transport on* $M$*.*

For $\Delta^W$ thus determined, we call $t^W := \beta^m \bar{\alpha}$ the Wolfe step size, that is, the Wolfe step size is a step size which is obtained with a backtracking procedure to satisfy the Wolfe condition. We consider that a natural choice of a vector transport $\mathcal{T}$ for the Wolfe condition is the differentiated retraction $\mathcal{T}^R$, as is discussed in Chapter 3.

In contrast with the steepest descent method, the search direction in the standard conjugate gradient method is determined by the negative gradient of the objective function $f$ at a current iterate together with the vector transport of the previous search direction;

$$\eta_{k+1} = -\operatorname{grad} f(x_{k+1}) + \beta_{k+1} \mathcal{T}_{\alpha_k \eta_k}(\eta_k), \tag{4.4.10}$$

where $\alpha_k$ is a Wolfe step size with $R_{x_k}(\alpha_k \eta_k) = x_{k+1}$ and where several choices are possible for $\beta_{k+1}$. We proposed the scaled Fletcher-Reeves type conjugate gradient method in Chapter 3, in which (4.4.10) is replaced with

$$\eta_{k+1} = -\operatorname{grad} f(x_{k+1}) + \min\left(1, \frac{\|\eta_k\|_{x_k}}{\|\mathcal{T}_{\alpha_k \eta_k}(\eta_k)\|_{x_{k+1}}}\right) \beta_{k+1} \mathcal{T}_{\alpha_k \eta_k}(\eta_k). \tag{4.4.11}$$

It is to be noted that the present algorithm is shown to have a global convergence property.

With these matters in mind, we describe a conjugate gradient algorithm 4.4.2 for Problem 4.2.2 with a variety of choosing $\beta_{k+1}$, $\mathcal{T}$, independently of whether the scaling is performed or not, on introducing the notation

$$\left(\bar{\xi}_k, \bar{\eta}_k\right) := \operatorname{grad} F\left(U_k, V_k\right) \tag{4.4.12}$$

for simplicity of expression. We note also that we have also choices of a retraction $R$, though we use the QR retraction in Algorithm 4.4.2. For more details of other retractions, see [AM12].

**Algorithm 4.4.2** Conjugate gradient method for Problem 4.2.2

1: Choose an initial point $(U_0, V_0) \in \mathrm{St}(p,m) \times \mathrm{St}(p,n)$.
2: Set

$$\begin{aligned}(\xi_0, \eta_0) &= -\operatorname{grad} F(U_0, V_0) \\ &= \left(AV_0N - U_0 \operatorname{sym}\left(U_0^T A V_0 N\right), A^T U_0 N - V_0 \operatorname{sym}\left(V_0^T A^T U_0 N\right)\right)\end{aligned} \tag{4.4.13}$$

and $\left(\bar{\xi}_0, \bar{\eta}_0\right) = -(\xi_0, \eta_0)$.
3: **for** $k = 0, 1, 2, \ldots$ **do**
4: Compute the Wolfe step size $\alpha_k$ and set

$$\begin{aligned}(U_{k+1}, V_{k+1}) &= R_{(U_k, V_k)}(\alpha_k(\xi_k, \eta_k)) \\ &= (\operatorname{qf}(U_k + \alpha_k \xi_k), \operatorname{qf}(V_k + \alpha_k \eta_k)).\end{aligned} \tag{4.4.14}$$

5: Compute

$$\begin{aligned}\left(\bar{\xi}_{k+1}, \bar{\eta}_{k+1}\right) &= \operatorname{grad} F(U_{k+1}, V_{k+1}) \\ &= \left(U_{k+1} \operatorname{sym}\left(U_{k+1}^T A V_{k+1} N\right) - A V_{k+1} N,\right. \\ &\qquad \left. V_{k+1} \operatorname{sym}\left(V_{k+1}^T A^T U_{k+1} N\right) - A^T U_{k+1} N\right).\end{aligned} \tag{4.4.15}$$

6: Compute $C_{k+1}$ and $\beta_{k+1}$.
7: Set

$$(\xi_{k+1}, \eta_{k+1}) = -\operatorname{grad} F(U_{k+1}, V_{k+1}) + C_{k+1}\beta_{k+1}\mathcal{T}_{\alpha_k(\xi_k, \eta_k)}(\xi_k, \eta_k) \tag{4.4.16}$$

8: **end for**

In Algorithm 4.4.2, possible choices of a vector transport $\mathcal{T}$ are $\mathcal{T}^R$ and $\mathcal{T}^P$ given in Prop. 4.4.1. In Step 6 of the algorithm, a real number $C_k$ is fixed to $C_k := 1$ for all $k$ in the standard conjugate gradient method, whereas $C_k$ is taken as a scaling factor defined by

$$C_k := \min\left(1, \frac{\|(\xi_k, \eta_k)\|_{(U_k, V_k)}}{\|\mathcal{T}_{\alpha_k(\xi_k, \eta_k)}(\xi_k, \eta_k)\|_{(U_{k+1}, V_{k+1})}}\right) \tag{4.4.17}$$

in the scaled conjugate gradient method, which we proposed in Chapter 3. In the same step of the algorithm, we can choose the $\beta$ of Fletcher-Reeves or Polak-Ribière, which are given as

$$\begin{aligned}\beta_{k+1}^{\mathrm{FR}} &= \frac{\langle \operatorname{grad} F(U_{k+1}, V_{k+1}), \operatorname{grad} F(U_{k+1}, V_{k+1})\rangle_{(U_{k+1}, V_{k+1})}}{\langle \operatorname{grad} F(U_k, V_k), \operatorname{grad} F(U_k, V_k)\rangle_{(U_k, V_k)}} \\ &= \frac{\operatorname{tr}\left(\bar{\xi}_{k+1}^T \bar{\xi}_{k+1}\right) + \operatorname{tr}\left(\bar{\eta}_{k+1}^T \bar{\eta}_{k+1}\right)}{\operatorname{tr}\left(\bar{\xi}_k^T \bar{\xi}_k\right) + \operatorname{tr}\left(\bar{\eta}_k^T \bar{\eta}_k\right)},\end{aligned} \tag{4.4.18}$$

and

$$\begin{aligned}\beta_{k+1}^{\mathrm{PR}} =& \frac{\langle \operatorname{grad} F\left(U_{k+1}, V_{k+1}\right), \operatorname{grad} F\left(U_{k+1}, V_{k+1}\right) - \mathcal{T}_{\alpha_k(\xi_k,\eta_k)}(\operatorname{grad} F\left(U_k, V_k\right))\rangle_{(U_{k+1},V_{k+1})}}{\langle \operatorname{grad} F\left(U_k, V_k\right), \operatorname{grad} F\left(U_k, V_k\right)\rangle_{(U_k,V_k)}} \\ =& \frac{\operatorname{tr}\left(\bar{\xi}_{k+1}^T\left(\bar{\xi}_{k+1} - \zeta_{k+1}\right)\right) + \operatorname{tr}\left(\bar{\eta}_{k+1}^T\left(\bar{\eta}_{k+1} - \chi_{k+1}\right)\right)}{\operatorname{tr}\left(\bar{\xi}_k^T\bar{\xi}_k\right) + \operatorname{tr}\left(\bar{\eta}_k^T\bar{\eta}_k\right)},\end{aligned} \tag{4.4.19}$$

respectively, where we have introduced the notation

$$\left(\zeta_{k+1}, \chi_{k+1}\right) = \mathcal{T}_{\alpha_k(\xi_k,\eta_k)}\left(\operatorname{grad} F\left(U_k, V_k\right)\right). \tag{4.4.20}$$

We perform numerical experiments with various types of Algorithm 4.4.2 for Problem 4.2.2, where the variation of the algorithms depends on the choices of $C_k$, $\beta_k$, and $\mathcal{T}$. For a $500 \times 300$ matrix $A$, we obtain Fig. 4.4.2, whose vertical axis carries the differences between the values $F(U_k, V_k)$ and the minimum value $F_{\min}$ of $F$. In Fig. 4.4.2, the abbre-

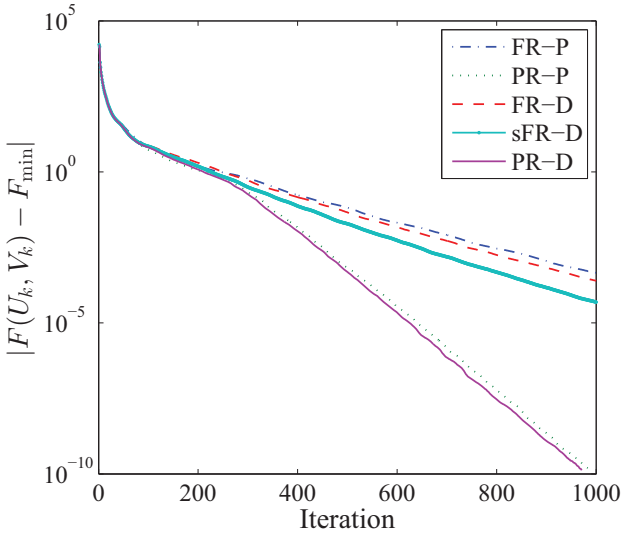



Figure 4.4.2: Comparison among the performances of several conjugate gradient algorithms with respect to the difference between the optimal and the current values of the objective function.

viations FR, sFR, PR in the legend mean the standard Fletcher-Reeves type (with $\beta^{\mathrm{FR}}$ in Eq. (4.4.18) and $C_k = 1$), the scaled Fletcher-Reeves type (with $\beta^{\mathrm{FR}}$ in Eq. (4.4.18) and $C_k$ in Eq. (4.4.17)), the standard Polak-Ribière type (with $\beta^{\mathrm{PR}}$ in Eq. (4.4.19) and $C_k = 1$), respectively. Further, the abbreviations P and D mean the vector transports by the orthogonal projection $\mathcal{T}^P$ and by the differentiated retraction $\mathcal{T}^R$, respectively. We note that sFR-D type is what we proposed in Chapter 3 and its global convergence property is guaranteed (see also Prop. 3.7.2 in the appendix of Chapter 3). We can observe

that the scaling procedure indeed improves the performance of FR-D type algorithm. We also note that the graph of sFR-P type, which does not appear in Fig. 4.4.2, coincide with that of FR-P type. However, as is in the case for Euclidean conjugate gradient method, algorithms with the $\beta$ of Polak-Ribière show better performances than those with the $\beta$ of Fletcher-Reeves, though a global convergence is not guaranteed for the algorithm with $\beta^{\mathrm{PR}}$. While PR-D type algorithm shows a faster convergence than PR-P type, the computational time for computing $\mathcal{T}^R$ in PR-D type may be longer than that for computing $\mathcal{T}^P$ in PR-P type.

From these observations, in what follows in this chapter, we describe the PR-P type algorithm in detail, that is, we use the vector transport (4.4.6) together with the $\beta$ of Polak-Ribière (4.4.19).

**Algorithm 4.4.3** Polak-Ribière type conjugate gradient method with the vector transport by the orthogonal projection for Problem 4.2.2

1: Choose an initial point $(U_0, V_0) \in \mathrm{St}(p,m) \times \mathrm{St}(p,n)$.

2: Set

$$(\xi_0, \eta_0) = \left(AV_0N - U_0 \operatorname{sym}\left(U_0^T A V_0 N\right), A^T U_0 N - V_0 \operatorname{sym}\left(V_0^T A^T U_0 N\right)\right) \tag{4.4.21}$$

and $\left(\bar{\xi}_0, \bar{\eta}_0\right) = -\left(\xi_0, \eta_0\right)$.

3: **for** $k = 0, 1, 2, \ldots$ **do**

4: Compute the Wolfe step size $\alpha_k$ and set

$$(U_{k+1}, V_{k+1}) = \left(\operatorname{qf}\left(U_k + \alpha_k \xi_k\right), \operatorname{qf}\left(V_k + \alpha_k \eta_k\right)\right). \tag{4.4.22}$$

5: Compute

$$\begin{aligned}(\zeta_{k+1}, \chi_{k+1}) = \Big(&\bar{\xi}_k - \operatorname{qf}\left(U_k + \alpha_k \xi_k\right) \operatorname{sym}\left(\operatorname{qf}\left(U_k + \alpha_k \xi_k\right)^T \bar{\xi}_k\right), \\ &\bar{\eta}_k - \operatorname{qf}\left(V_k + \alpha_k \eta_k\right) \operatorname{sym}\left(\operatorname{qf}\left(V_k + \alpha_k \eta_k\right)^T \bar{\eta}_k\right)\Big)\end{aligned} \tag{4.4.23}$$

and

$$\begin{aligned}\left(\bar{\xi}_{k+1}, \bar{\eta}_{k+1}\right) = \Big(&U_{k+1} \operatorname{sym}\left(U_{k+1}^T A V_{k+1} N\right) - A V_{k+1} N, \\ &V_{k+1} \operatorname{sym}\left(V_{k+1}^T A^T U_{k+1} N\right) - A^T U_{k+1} N\Big).\end{aligned} \tag{4.4.24}$$

6: Compute $\beta_{k+1}$ by

$$\beta_{k+1} = \frac{\operatorname{tr}\left(\bar{\xi}_{k+1}^T \left(\bar{\xi}_{k+1} - \zeta_{k+1}\right)\right) + \operatorname{tr}\left(\bar{\eta}_{k+1}^T \left(\bar{\eta}_{k+1} - \chi_{k+1}\right)\right)}{\operatorname{tr}\left(\bar{\xi}_k^T \bar{\xi}_k\right) + \operatorname{tr}\left(\bar{\eta}_k^T \bar{\eta}_k\right)}. \tag{4.4.25}$$

7: Set

$$\begin{aligned}(\xi_{k+1}, \eta_{k+1}) = \Big(&-\bar{\xi}_{k+1} + \beta_{k+1}\left(\xi_k - \operatorname{qf}\left(U_k + \alpha_k \xi_k\right) \operatorname{sym}\left(\operatorname{qf}\left(U_k + \alpha_k \xi_k\right)^T \xi_k\right)\right), \\ &-\bar{\eta}_{k+1} + \beta_{k+1}\left(\eta_k - \operatorname{qf}\left(V_k + \alpha_k \eta_k\right) \operatorname{sym}\left(\operatorname{qf}\left(V_k + \alpha_k \eta_k\right)^T \eta_k\right)\right)\Big).\end{aligned} \tag{4.4.26}$$

8: **end for**

### 4.4.3 Newton's method on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$

We now set up Newton's method for Problem 4.2.2. The only difference between Newton's method and the steepest descent method lies in the choice of the search direction. In Newton's method for Problem 2.2.1, the search direction $\Delta_k \in T_{x_k} M$ at $x_k \in M$ is determined

to be the solution to Newton's equation

$$\operatorname{Hess} f(x_k)[\Delta_k] = -\operatorname{grad} f(x_k), \tag{4.4.27}$$

and we set the step size to be $t_k = 1$ for simplicity.

Since we have already computed in Prop. 4.3.5 the Hessian of our objective function $F$, we can describe Newton's method for Problem 4.2.2 as follows:

**Algorithm 4.4.4** Newton's method for Problem 4.2.2

1: Choose an initial point $(U_0, V_0) \in \mathrm{St}(p,m) \times \mathrm{St}(p,n)$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Solve Newton's equation

$$\begin{cases} \xi_k S_{1,k} - A\eta_k N - U_k \operatorname{sym}\left(U_k^T(\xi_k S_{1,k} - A\eta_k N)\right) = AV_k N - U_k S_{1,k}, \\ \eta_k S_{2,k} - A^T\xi_k N - V_k \operatorname{sym}\left(V_k^T\left(\eta_k S_{2,k} - A^T\xi_k N\right)\right) = A^T U_k N - V_k S_{2,k} \end{cases} \tag{4.4.28}$$

for the unknown $(\xi_k, \eta_k) \in T_{(U_k,V_k)}(\mathrm{St}(p,m) \times \mathrm{St}(p,n))$, where $S_{1,k} = \operatorname{sym}\left(U_k^T A V_k N\right)$ and $S_{2,k} = \operatorname{sym}\left(V_k^T A^T U_k N\right)$.
4: Compute the next iterate $(U_{k+1}, V_{k+1}) = R_{(U_k,V_k)}((\xi_k, \eta_k))$, where $R$ is a retraction on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$.
5: **end for**

The Newton's equation system (4.4.28) is difficult to solve, because two unknown matrices $\xi_k$ and $\eta_k$ are coupled together. However, in the case of $p = 1$, Eqs. (4.4.28) are tractable. Here, the diagonal matrix $N$ becomes a scalar and can be put as $N = 1$ without loss of generality. Since $\mathrm{St}(1,m) \times \mathrm{St}(1,n) = S^{m-1} \times S^{n-1}$, the condition for $\xi_k$ (resp. $\eta_k$) to be a tangent vector to $S^{m-1}$ (resp. $S^{n-1}$) reduces to $U_k^T\xi_k = 0$ (resp. $V_k^T\eta_k = 0$), and thereby Eqs. (4.4.28) reduce to

$$S_k\xi_k - \left(I_m - U_kU_k^T\right)A\eta_k = \left(I_m - U_kU_k^T\right)AV_k, \tag{4.4.29}$$

$$S_k\eta_k - \left(I_n - V_kV_k^T\right)A^T\xi_k = \left(I_n - V_kV_k^T\right)A^TU_k, \tag{4.4.30}$$

where $S_{1,k} = S_{2,k} =: S_k$. Further, $S_k$ are scalars.

If $S_k \neq 0$, it follows from Eq. (4.4.29) that

$$\xi_k = S_k^{-1}\left(I_m - U_kU_k^T\right)A\left(V_k + \eta_k\right). \tag{4.4.31}$$

Substituting Eq. (4.4.31) into Eq. (4.4.30), we obtain

$$S_k\eta_k - S_k^{-1}\left(I_n - V_kV_k^T\right)A^T\left(I_m - U_kU_k^T\right)A\left(V_k + \eta_k\right) = \left(I_n - V_kV_k^T\right)A^TU_k. \tag{4.4.32}$$

If $S_k^2 I_n - \left(I_n - V_k V_k^T\right) A^T \left(I_m - U_k U_k^T\right) A$ is invertible, Eq. (4.4.32) results in

$$\eta_k = \left(S_k^2 I_n - \left(I_n - V_k V_k^T\right) A^T \left(I_m - U_k U_k^T\right) A\right)^{-1} \left(I_n - V_k V_k^T\right) A^T A V_k. \tag{4.4.33}$$

Once $\eta_k$ is computed, $\xi_k$ is also computed by (4.4.31). Alternatively, if $S_k \neq 0$ and if $S_k^2 I_m - \left(I_m - U_k U_k^T\right) A \left(I_n - V_k V_k^T\right) A^T$ is invertible, then

$$\xi_k = \left(S_k^2 I_m - \left(I_m - U_k U_k^T\right) A \left(I_n - V_k V_k^T\right) A^T\right)^{-1} \left(I_m - U_k U_k^T\right) A A^T U_k, \tag{4.4.34}$$

and hence

$$\eta_k = S_k^{-1} \left(I_n - V_k V_k^T\right) A^T \left(U_k + \xi_k\right). \tag{4.4.35}$$

In view of our assumption that $m \geq n$, we are inclined to use Eqs. (4.4.33) and (4.4.31) on account of the size of the matrix for which the inverse is to be calculated.

We obtain the following algorithm of Newton's method for Problem 4.2.2 with $p = 1$ and $N = 1$, where the uppercase letters $U$, $S$, and $V$ are replaced by the lowercase ones $u$, $s$, and $v$, respectively, because $u$ and $v$ are vectors and $s$ is a scalar in the case of $p = 1$, and where the QR-based retraction takes a simple form.

---

**Algorithm 4.4.5** Newton's method for Problem 4.2.2 with $p = 1$ and $N = 1$

---

1: Choose an initial point $(u_0, v_0) \in \mathrm{St}(1, m) \times \mathrm{St}(1, n) = S^{m-1} \times S^{n-1}$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Compute $\eta_k$ and $\xi_k$ by

$$\eta_k = \left(s_k^2 I_n - \left(I_n - v_k v_k^T\right) A^T \left(I_m - u_k u_k^T\right) A\right)^{-1} \left(I_n - v_k v_k^T\right) A^T A v_k, \tag{4.4.36}$$

$$\xi_k = s_k^{-1} \left(I_m - u_k u_k^T\right) A \left(v_k + \eta_k\right), \tag{4.4.37}$$

where $s_k = u_k^T A v_k$.
4: Set $(u_{k+1}, v_{k+1}) = R_{(u_k, v_k)}\left((\xi_k, \eta_k)\right) = \left(\dfrac{u_k + \xi_k}{\|u_k + \xi_k\|}, \dfrac{v_k + \eta_k}{\|v_k + \eta_k\|}\right)$.
5: **end for**

---

If we know a good approximate solution of the problem in advance, Newton's method works effectively. This is because Newton's method generates locally but quadratically convergent sequences in general, as is shown in Thm. 2.2.2. However, if the initial point is not chosen in the neighborhood of a global optimal solution, the target of the sequence may not be the global optimal solution but another critical point in general. We propose a method to settle this issue in the next section.

Another question arises as to what will happen if the Hessian of $f$ is degenerate at a critical point. We will investigate this question in Section 4.6 for our objective function together with numerical experiments.

## 4.5 New algorithms for the singular value decomposition based on optimization methods on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$

In this section, we first develop an algorithm for computing the largest singular value and associated singular vectors of a matrix with $p = 1$. Then, we propose an improved algorithm for the singular value decomposition without the restriction of $p = 1$.

### 4.5.1 An algorithm for computing the largest singular value and associated left and right singular vectors

We consider Problem 4.2.2 with $p = 1$ and $N = 1$. Since $p = 1$, Newton's equation (4.4.28) can be solved as in (4.4.33) and (4.4.31), so that Algorithm 4.4.5 can be applied. However, the sequence generated by Newton's method does not necessarily converge to a global optimal solution, but often to a local one. Taking this into account, we start with the conjugate gradient method for Problem 4.2.2, and then switch the method to Newton's method, if the current iterate is sufficiently close to an optimal solution. The new algorithm is stated as follows:

**Algorithm 4.5.1** Hybrid method for Problem 4.2.2 with $p = 1$ and $N = 1$

1: Choose an initial point $(U_0, V_0) \in S^{m-1} \times S^{n-1}$ and a parameter $\varepsilon > 0$. Set $k := 0$.
2: Set

$$\begin{aligned}(\xi_0, \eta_0) &= -\operatorname{grad} F(U_0, V_0) \\ &= \left(AV_0N - U_0 \operatorname{sym}\left(U_0^T A V_0 N\right), A^T U_0 N - V_0 \operatorname{sym}\left(V_0^T A^T U_0 N\right)\right) \\ &= \left(AV_0 - U_0 U_0^T A V_0, A^T U_0 - V_0 V_0^T A^T U_0\right) \end{aligned} \tag{4.5.1}$$

and $\left(\bar{\xi}_0, \bar{\eta}_0\right) = -(\xi_0, \eta_0)$.
3: **while** $\| \operatorname{grad} F(U_k, V_k) \|_{(U_k, V_k)} > \varepsilon$ **do**
4: Perform Steps 4–7 in Algorithm 4.4.3.
5: $k := k + 1$.
6: **end while**
7: Set $(u_0, v_0) := (U_k, V_k)$ and $k := 0$.
8: Perform Steps 2–5 in Algorithm 4.4.5.

### 4.5.2 An algorithm for computing the $p$ largest singular values and associated left and right singular vectors

For Problem 4.2.2 with a generic number $p$, we first apply the conjugate gradient method (Algorithm 4.4.3) to obtain a point $\left(\tilde{U}, \tilde{V}\right)$ on $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ close to a global op-

timal solution. Let $u_1, \ldots, u_p$ and $v_1, \ldots, v_p$ denote columns of $\tilde{U}$ and $\tilde{V}$ from the left, respectively; $\tilde{U} = (u_1, \ldots, u_p)$, $\tilde{V} = (v_1, \ldots, v_p)$. Then, $u_i$ and $v_i$ are close to singular vectors $u_i^*$ and $v_i^*$ associated with the $i$-th largest singular value, respectively. If $u_i$ and $v_i$ are sufficiently close to $u_i^*$ and $v_i^*$, then the function $\mathrm{tr}\left(u_i^T A v_i\right)$ takes values near to the singular value $\sigma_i$, as was seen in the paragraph after the proof of Prop. 4.2.1, so that Newton's method with $p = 1$ and with $i$ fixed (Algorithm 4.4.5) generates a sequence on $S^{m-1} \times S^{n-1}$ which quadratically converges to $(u_i^*, v_i^*)$. If we obtain singular vectors $u_i^*$ and $v_i^*$ by Newton's method for $i = 1, \ldots, p$, we put them together to form $U_* = \left(u_1^*, \ldots, u_p^*\right)$ and $V_* = \left(v_1^*, \ldots, v_p^*\right)$. As the singular vectors are mutually orthogonal if singular values are distinct, we see that $U_* \in \mathrm{St}(p, m)$ and $V_* \in \mathrm{St}(p, n)$. Thus we divide our problem into $p$ subproblems. We now propose the following Algorithm 4.5.2.

**Algorithm 4.5.2** Hybrid method for Problem 4.2.2

1: Choose an initial point $(U_0, V_0) \in \mathrm{St}(p, m) \times \mathrm{St}(p, n)$ and a parameter $\varepsilon > 0$. Set $k := 0$.
2: Set

$$\begin{aligned}(\xi_0, \eta_0) &= -\operatorname{grad} F\left(U_0, V_0\right) \\ &= \left(AV_0N - U_0 \operatorname{sym}\left(U_0^T A V_0 N\right), A^T U_0 N - V_0 \operatorname{sym}\left(V_0^T A^T U_0 N\right)\right)\end{aligned} \tag{4.5.2}$$

and $\left(\bar{\xi}_0, \bar{\eta}_0\right) = -\left(\xi_0, \eta_0\right)$.
3: **while** $\| \operatorname{grad} F\left(U_k, V_k\right) \|_{(U_k, V_k)} > \varepsilon$ **do**
4: Perform Steps 4–7 in Algorithm 4.4.3.
5: $k := k + 1$.
6: **end while**
7: Set $\left(\tilde{U}, \tilde{V}\right) := (U_k, V_k)$.
8: **for** $i = 1, 2, \ldots, p$ **do**
9: Set $(u_0, v_0) := \left(\tilde{U}_i, \tilde{V}_i\right)$ and $k := 0$, where $\tilde{U}_i$ and $\tilde{V}_i$ are the $i$-th column vectors of $\tilde{U}$ and $\tilde{V}$, respectively.
10: Perform Steps 2–5 in Algorithm 4.4.5.
11: **end for**

We here again note that we can use the scaled Fletcher-Reeves type conjugate gradient method with the differentiated retraction vector transport (Algorithm 4.4.2 with $\beta = \beta^{\mathrm{FR}}$, $C_k = 1$, $\mathcal{T} = \mathcal{T}^R$) instead of Algorithm 4.4.3, in order to theoretically ensure the global convergence of the algorithm.

We perform numerical experiments with the proposed three algorithms for Problem 4.2.2 with $m = 500$, $n = 300$, $p = 10$, $N = \mathrm{diag}(10, \ldots, 2, 1)$. Here, matrices in question are generated so as to take the form

$$A = U_{\mathrm{r}} \operatorname{diag}(\sigma_1, \ldots, \sigma_n) V_{\mathrm{r}}^T, \tag{4.5.3}$$

where $U_{\rm r} \in \mathbb{R}^{m\times n}$ and $V_{\rm r} \in \mathbb{R}^{n\times n}$ are orthonormal matrices with randomly chosen elements, and where $\sigma_1 \geq \cdots \geq \sigma_n$ are also randomly chosen out of the interval $[0, 300]$. Figs. 4.5.1 and 4.5.2 show the comparison among the performances of Algorithms 4.4.1, 4.4.3, and 4.5.2, where we set $\varepsilon = 0.5$ in Algorithm 4.5.2, and where the vertical axes of these figures carry different measures. For a given $A$, an initial point is also chosen randomly on $\mathrm{St}(p, m) \times \mathrm{St}(p, n)$.

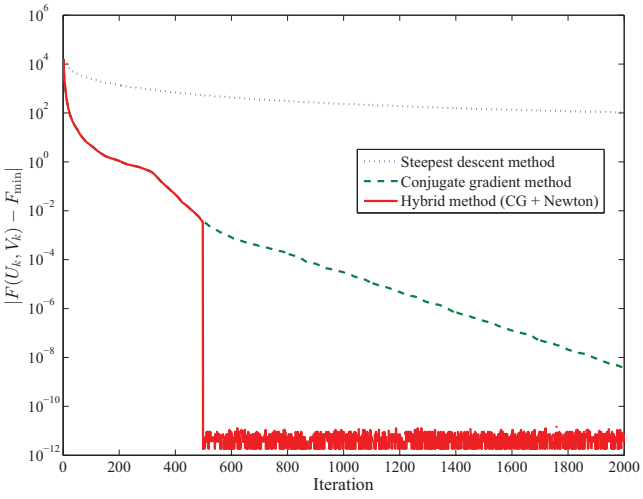



Figure 4.5.1: Comparison among the performances of the three algorithms with reference to the difference between the optimal and the current values of the objective function.

The dotted curves in Figs. 4.5.1 and 4.5.2, which are generated by Algorithm 4.4.1, show that the convergence of the steepest descent method is very slow. The middle-located dashed curves (from the upper left to the lower right), which are generated by Algorithm 4.4.3, show that the convergence in the conjugate gradient method is much faster than that in the steepest descent method. If a point close to a global optimal solution is obtained by the conjugate gradient method, switching to Newton's method makes the convergence drastically faster, as is seen in the vertical segments at the iteration number 498. Put another way, Algorithm 4.5.2 generates a curve composed of three pieces, one of which is an initial part of the curve generated by the conjugate gradient method, the second piece is the vertical line segment, and the last piece is the jagged line segment sitting in the bottom of Figs. 4.5.1 and 4.5.2. The jagged line segment means that our optimal solution is the best within accuracy subject to machine epsilon.

The computational time for 2000 iterations in Algorithm 4.4.1 and 4.4.3 are 209.4431 seconds and 410.8532 seconds, respectively. For Algorithm 4.5.2, it takes 101.2578 seconds for 498 iterations in the conjugate gradient method, which is the required time before switching to Newton's method, and 2.7388 seconds for 1 iteration in Newton's method,

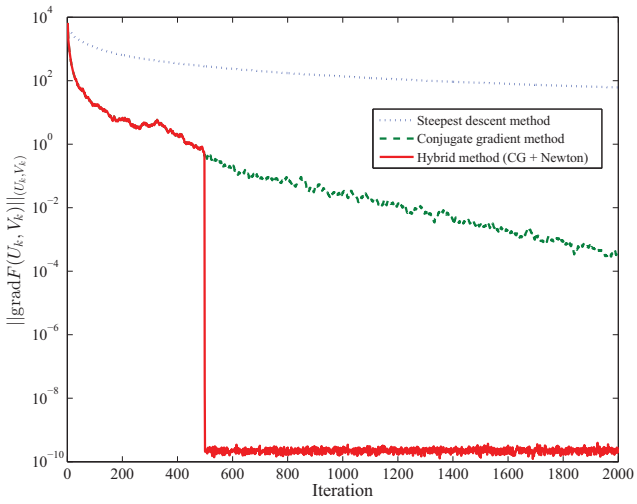



Figure 4.5.2: Comparison among the performances of the three algorithms with reference to the norm of the gradient of the objective function.

which corresponds to the vertical line segments in Figs. 4.5.1 and 4.5.2. Even though Newton's method takes the longest time per iteration among the three, the convergence is very quick as a whole.

We here note that though Algorithm 4.5.2 consists of two stages, the conjugate gradient and Newton's parts, the algorithm is by no means complicated. Though the usual approach to the singular value decomposition needs preconditioning, the present algorithm does not. The conjugate gradient part of Algorithm 4.5.2 seems to be like preconditioning.

### 4.5.3 Accuracy of numerical solutions

If a good approximation to the singular value decomposition $\tilde{U}\tilde{\Sigma}\tilde{V}^T$ of a matrix $A$ is known in advance by using another method, then we have only to perform Newton's method, that is, Steps 8–11 of Algorithm 4.5.2, in order to obtain solutions of higher accuracy.

Suppose we are given matrices $A$ of the form (4.5.3) together with $N = \mathrm{diag}(5, \ldots, 2, 1)$, where $m = 300$, $n = 100$, $p = 5$, and where $U_{\mathrm{r}} \in \mathbb{R}^{m\times n}$ and $V_{\mathrm{r}} \in \mathbb{R}^{n\times n}$ are orthonormal matrices with randomly chosen elements. Singular values $\sigma_1 \geq \cdots \geq \sigma_n$ are also chosen randomly from the interval $[0, 100]$ under the condition that $A$ has distinct singular values among the largest $p$ singular values of each $A$. In this setting, optimal solutions are given by $U_{\mathrm{opt}} := U_{\mathrm{r}} I_{n,p}$ and $V_{\mathrm{opt}} := V_{\mathrm{r}} I_{n,p}$, where $I_{n,p} = \begin{pmatrix} I_p \\ \\ 0 \end{pmatrix} \in \mathbb{R}^{n\times p}$. Suppose we obtain

$\tilde{U}$, $\tilde{\Sigma}$, and $\tilde{V}$ factors of the truncated singular value decomposition of $A$ by applying MATLAB's svd function. We can perform Steps 8–11 of Algorithm 4.5.2 with these $\tilde{U}$ and $\tilde{V}$ as initial data in order to obtain more accurate decomposition $U_{\mathrm{New}}\Sigma_{\mathrm{New}}V_{\mathrm{New}}^T$. To see the degree of accuracy, we compare the Frobenius norms $\|\tilde{U}^T A\tilde{V} - U_{\mathrm{opt}}^T A V_{\mathrm{opt}}\|$ and $\|U_{\mathrm{New}}^T A V_{\mathrm{New}} - U_{\mathrm{opt}}^T A V_{\mathrm{opt}}\|$. If Newton's method gives a more accurate decomposition, the following inequality is expected to hold,

$$\|U_{\mathrm{New}}^T A V_{\mathrm{New}} - U_{\mathrm{opt}}^T A V_{\mathrm{opt}}\| < \|\tilde{U}^T A\tilde{V} - U_{\mathrm{opt}}^T A V_{\mathrm{opt}}\|. \tag{4.5.4}$$

Let $U_{\mathrm{New}}$ and $V_{\mathrm{New}}$ be matrices obtained by performing Steps 8–11 of Algorithm 4.5.2 only once. Our as many as 1000 experiments with randomly chosen matrices $A$ show that Eq. (4.5.4) holds 962 of the time out of 1000. This means that our Newton's method mostly enhances the accuracy of the decomposition obtained by MATLAB's svd function. We may perform 10 iterations of Steps 8–11 to obtain a sequence $(U_1, V_1), \ldots, (U_{10}, V_{10})$. If we are allowed to define $(U_{\mathrm{New}}, V_{\mathrm{New}})$ by

$$(U_{\mathrm{New}}, V_{\mathrm{New}}) := (U_i, V_i), \qquad i = \arg\min_{j=1,\ldots,10} \|U_j^T A V_j - U_{\mathrm{opt}}^T A V_{\mathrm{opt}}\|, \tag{4.5.5}$$

then Eq. (4.5.4) holds for all 1000 matrices of $A$. We conclude that if the singular values of $A$ are not degenerate, our Newton's method always generates singular value decompositions of higher accuracy in the end. If $A$ has degenerate singular values, however, unit column vectors generated by Newton's methods with $p = 1$ are not necessarily mutually orthogonal. We see what will happen if $A$ has degenerate singular values in detail in the next section.

## 4.6 Degenerate optimal solutions

If singular values are degenerate, the global optimal solutions form a continuum. To see this, we study the degeneracy of global optimal solutions, using the Hessian of the objective function $F$. In the proofs of the following propositions, we use the lemma [EAS98]:

**Lemma 4.6.1.** *The tangent space to* $\mathrm{St}(p,n)$ *at* $Y \in \mathrm{St}(p,n)$ *is given by*

$$T_Y\mathrm{St}(p,n) = \left\{\xi = YB + Y_\perp C \,|\, B \in \mathrm{Skew}(p),\, C \in \mathbb{R}^{(n-p)\times p}\right\}, \tag{4.6.1}$$

*where* $\mathrm{Skew}(p)$ *denotes the set of all* $p \times p$ *skew-symmetric matrices, and where* $Y_\perp$ *is an arbitrary* $n \times (n-p)$ *orthonormal matrix such that* $YY^T + Y_\perp Y_\perp^T = I_n$.

We first analyze the case where the $p$-th singular value is non-vanishing, $\sigma_p \neq 0$.

**Proposition 4.6.1.** *Assume that* $A$ *has* $k+1$ *distinct singular values among the largest* $p$ *singular values with multiplicity counted. In other words, the singular values of* $A \in \mathbb{R}^{m\times n}$ *are put in the form* $\sigma_1 = \cdots = \sigma_{n_1} > \cdots > \sigma_{n_1+\cdots+n_{k-1}+1} = \cdots = \sigma_{n_1+\cdots+n_k} > \sigma_{n_1+\cdots+n_k+1} =$

$\cdots = \sigma_p = \cdots = \sigma_{n_1+\cdots+n_{k+1}} > \cdots \geq \sigma_n$, *where* $n_1, \ldots, n_k, n_{k+1}$ *are multiplicities. If* $\sigma_p \neq 0$, *then global optimal solutions* $(U_*, V_*) \in \mathrm{St}(p,m) \times \mathrm{St}(p,n)$ *to Problem 4.2.2 form a submanifold diffeomorphic to* $\mathcal{M} := O(n_1) \times \cdots \times O(n_k) \times \mathrm{St}(p-q, n_{k+1})$, *where* $q := n_1 + \cdots + n_k$. *Further, the Hessian* $\mathrm{Hess}\, F(U,V)$ *is degenerate at* $(U_*, V_*)$ *for the tangent space to this submanifold, which is viewed as a subspace of* $T_{(U_*,V_*)}(\mathrm{St}(p,m) \times \mathrm{St}(p,n))$.

*Proof.* From Prop. 4.2.1 and from the course of its proof, it turns out that $(U,V)$ is a global optimal solution to Problem 4.2.2 if and only if

$$AV = US, \quad A^T U = VS, \tag{4.6.2}$$

where $S = \mathrm{diag}(\sigma_1, \ldots, \sigma_p)$. Let $(U,V)$ be put in the form $(U,V) = ((u_1, \ldots, u_p), (v_1, \ldots, v_p))$. From (4.6.2), one has $A^T A V = V S^2$, which means that $v_i$ is an eigenvector of $A^T A$ associated with the eigenvalue $\sigma_i^2$. Let $\left\{v_{q+1}, \ldots, v_p, \ldots, v_{q+n_{k+1}}\right\}$ be a basis of the eigenspace associated with the eigenvalues $\sigma_p^2$ of $A^T A$. Since eigenvalues of $A^T A$ are degenerate, the associated orthonormal eigenvectors admit orthogonal transformations in each eigenspace. Then, for any global optimal solution $(U_*, V_*)$ to Problem 4.2.2, $V_* = \left(v_1^*, \ldots, v_p^*\right)$ is related with $V = (v_1, \ldots, v_p)$ and $v_{p+1}, \ldots, v_{q+n_{k+1}}$ by

$$\begin{aligned} V_* = &\left(v_1, \ldots, v_{n_1}, \ldots, v_{n_1+\cdots+n_{k-1}+1}, \ldots, v_q, v_{q+1}, \ldots, v_p, \ldots, v_{q+n_{k+1}}\right) \\ &\times \begin{pmatrix} Q_1 & & & \\ & \ddots & & \\ & & Q_k & \\ & & & Q_{k+1} \end{pmatrix}, \end{aligned} \tag{4.6.3}$$

where $Q_i \in O(n_i)$, $i = 1, \ldots, k$, and $Q_{k+1} \in \mathrm{St}(p-q, n_{k+1})$, and where we note that $\left(v_{q+1}, \ldots, v_{q+n_{k+1}}\right) Q_{k+1}$ gives a system of $p-q$ orthonormal eigenvectors from the eigenspace associated with $\sigma_p^2$. Denoting by $\hat{V} \in \mathrm{St}(q+n_{k+1}, n)$ and by $Q \in \mathcal{M}$ the first and second factor matrices in the right-hand side of (4.6.3), respectively, we put (4.6.3) in the form

$$V_* = \hat{V} Q. \tag{4.6.4}$$

Further, once $V_*$ is expressed as above, $U_*$ is determined from (4.6.2) to be

$$U_* = A V_* S^{-1}. \tag{4.6.5}$$

This means that the optimal solution $(U_*, V_*)$ is determined by the second component

$V_*$ only. Eqs.(4.6.4) and (4.6.5) then mean that any optimal solution $(U_*, V_*)$ is related to $(U, V)$ by the transformation defined by $Q$. In other words, Eqs. (4.6.4) and (4.6.5) with $(U, V)$ fixed defines a diffeomorphism of $\mathcal{M}$ to the degeneracy submanifold formed by $(U_*, V_*)$'s.

We proceed to the degeneracy of the Hessian at a global optimal solution $(U_*, V_*)$. In association with $(U_*, V_*)$, $A$ has a singular value decomposition such that

$$A = (U_*, U_\perp) \begin{pmatrix} \mathrm{diag}(\sigma_1, \ldots, \sigma_n) \\ \\ 0 \end{pmatrix} (V_*, V_\perp)^T , \tag{4.6.6}$$

where $U_\perp \in \mathrm{St}(m-p, m)$ and $V_\perp \in \mathrm{St}(n-p, n)$. Since $(U_*, U_\perp)$ and $(V_*, V_\perp)$ are orthogonal matrices, we have $U_* U_*^T + U_\perp U_\perp^T = I_m$, $V_* V_*^T + V_\perp V_\perp^T = I_n$, and further $U_*^T U_\perp = 0$, $V_*^T V_\perp = 0$. Let $(\xi, \eta)$ be a tangent vector to $\mathrm{St}(p, m) \times \mathrm{St}(p, n)$ at $(U_*, V_*)$. Then, Eq. (4.6.1) shows that $\xi$ and $\eta$ can be written as

$$\xi = U_* B + U_\perp C, \quad \eta = V_* D + V_\perp E, \qquad B, D \in \mathrm{Skew}(p),\ C \in \mathbb{R}^{(m-p)\times p}, E \in \mathbb{R}^{(n-p)\times p}. \tag{4.6.7}$$

Note that $U_*^T A V_* = S = \mathrm{diag}(\sigma_1, \ldots, \sigma_p)$, as is seen from Eq. (4.6.2). Denoting $\mathrm{diag}(\sigma_{p+1}, \ldots, \sigma_n)$ by $S_\perp$, we express $\mathrm{diag}\,(\sigma_1, \ldots, \sigma_n)$ as $\begin{pmatrix} S & \\ & S_\perp \end{pmatrix}$. Then, we obtain from Eq. (4.6.6)

$$U_*^T A V_\perp = U_*^T (U_*, U_\perp) \left( \begin{array}{c} \begin{matrix} S & \\ & S_\perp \end{matrix} \\ \hdashline 0 \end{array} \right) (V_*, V_\perp)^T V_\perp = 0, \tag{4.6.8}$$

where we have used the fact that $U_*^T U_\perp = 0$ and $V_*^T V_\perp = 0$. Similarly, we also have $U_\perp^T A V_* = 0$ and $U_\perp^T A V_\perp = \begin{pmatrix} S_\perp \\ \\ 0 \end{pmatrix}$. We are now in a position to write out the quadratic form $\langle \mathrm{Hess}\, F(U_*, V_*)[(\xi, \eta)], (\xi, \eta) \rangle_{(U_*, V_*)}$ by using Eq. (4.3.22). A calculation is performed

to provide

$$
\begin{aligned}
&\langle \mathrm{Hess}\, F(U_*,V_*)[(\xi,\eta)],(\xi,\eta)\rangle_{(U_*,V_*)}\\
&=\mathrm{tr}\left(\xi^T\xi U_*^TAV_*N+U_*^TAV_*\eta^T\eta N-2\xi^TA\eta N\right)\\
&=\mathrm{tr}\left(\begin{array}{l}\left(B^TB+C^TC\right)SN+S\left(D^TD+E^TE\right)N-2\left(B^TSD+C^T\begin{pmatrix}S_\perp\\0\end{pmatrix}E\right)N\\ \end{array}\right)\\
&=\sum_{i,j=1}^{p}\left(\sigma_j\mu_j\left(b_{ij}^2+d_{ij}^2\right)-2\sigma_i\mu_jb_{ij}d_{ij}\right)+\sum_{i=1}^{m-p}\sum_{j=1}^{p}\sigma_j\mu_jc_{ij}^2+\sum_{i=1}^{n-p}\sum_{j=1}^{p}\left(\sigma_j\mu_je_{ij}^2-2\sigma_{p+i}\mu_jc_{ij}e_{ij}\right)\\
&=\sum_{\substack{i,j=1\\ i<j}}^{p}\left(\left(\sigma_i\mu_i+\sigma_j\mu_j\right)\left(b_{ij}^2+d_{ij}^2\right)-2\left(\sigma_i\mu_j+\sigma_j\mu_i\right)b_{ij}d_{ij}\right)\\
&\quad+\sum_{i=1}^{n-p}\sum_{j=1}^{p}\mu_j\left(\sigma_j\left(c_{ij}^2+e_{ij}^2\right)-2\sigma_{p+i}c_{ij}e_{ij}\right)+\sum_{i=n-p+1}^{m-p}\sum_{j=1}^{p}\sigma_j\mu_jc_{ij}^2\\
&=\sum_{\substack{i,j=1\\ i<j}}^{p}\left(\sigma_i\mu_i+\sigma_j\mu_j\right)^{-1}\left(\left(\left(\sigma_i\mu_j+\sigma_j\mu_i\right)b_{ij}-\left(\sigma_i\mu_i+\sigma_j\mu_j\right)d_{ij}\right)^2+\left(\sigma_i^2-\sigma_j^2\right)\left(\mu_i^2-\mu_j^2\right)b_{ij}^2\right)\\
&\quad+\sum_{i=1}^{n-p}\sum_{j=1}^{p}\mu_j\sigma_j^{-1}\left(\left(\sigma_{p+i}c_{ij}-\sigma_je_{ij}\right)^2+\left(\sigma_j^2-\sigma_{p+i}^2\right)c_{ij}^2\right)+\sum_{i=n-p+1}^{m-p}\sum_{j=1}^{p}\sigma_j\mu_jc_{ij}^2. \qquad (4.6.9)
\end{aligned}
$$

We observe from (4.6.9) that the Hessian quadratic form is positive semi-definite and further that $b_{ij}$ with $i,j\in\{1,\ldots,n_1\},\{n_1+1,\ldots,n_1+n_2\},\ldots,\{q+1,\ldots,p\}$ and $c_{ij}$ with $i=1,\ldots,q+n_{k+1}-p,\ j=q+1,\ldots,p,$ make no contribution to the positivity of the Hessian quadratic form because of the degeneracy of singular values. Hence, the $\langle \mathrm{Hess}\, F(U_*,V_*)[(\xi,\eta)],(\xi,\eta)\rangle_{(U_*,V_*)}$ vanishes if and only if $b_{ij}$ with $i,j\in\{1,\ldots,n_1\}$, $\{n_1+1,\ldots,n_1+n_2\},\ldots,\{q+1,\ldots,p\}$ and $c_{ij}$ with $i=1,\ldots,q+n_{k+1}-p,\ j=q+1,\ldots,p$ are arbitrary but subject to the condition that $b_{ij}=-b_{ji}$, and the other $b_{ij}$ and $c_{ij}$ are 0, and further $d_{ij}$ and $e_{ij}$ satisfy

$$
d_{ij}=\left(\sigma_i\mu_i+\sigma_j\mu_j\right)^{-1}\left(\sigma_i\mu_j+\sigma_j\mu_i\right)b_{ij},\quad e_{ij}=\sigma_j^{-1}\sigma_{p+i}c_{ij}, \qquad (4.6.10)
$$

respectively. The above-mentioned conditions for $B$ and $C$ are put in the form

$$B = \begin{pmatrix} B_1 & & & \\ & \ddots & & \\ & & B_k & \\ & & & B_{k+1} \end{pmatrix}, \quad C = \begin{pmatrix} & C_{k+1} \\ 0 & \end{pmatrix}, \tag{4.6.11}$$

respectively, where $B_i \in \mathrm{Skew}(n_i), i = 1, \ldots, k$, $B_{k+1} \in \mathrm{Skew}(p-q)$, and $C_{k+1} \in \mathbb{R}^{(q+n_{k+1}-p)\times(p-q)}$. It then turns out that the Hessian is degenerate for the subspace of $T_{(U_*,V_*)}(\mathrm{St}(p,m) \times \mathrm{St}(p,n))$ which is isomorphic to

$$\begin{aligned} &\mathrm{Skew}(n_1) \times \cdots \times \mathrm{Skew}(n_k) \times \mathrm{Skew}(p-q) \times \mathbb{R}^{(q+n_{k+1}-p)\times(p-q)} \\ \simeq &T_{Q_1}O(n_1) \times \cdots \times T_{Q_k}O(n_k) \times T_{Q_{k+1}}\mathrm{St}(p-q, n_{k+1}) \simeq T_Q\mathcal{M}. \end{aligned} \tag{4.6.12}$$

This completes the proof. □

A similar reasoning applies to the case of $\sigma_p = 0$ as follows:

**Proposition 4.6.2.** *Assume that the singular values of $A \in \mathbb{R}^{m\times n}$ are given in descending order by $\sigma_1 = \cdots = \sigma_{n_1} > \cdots > \sigma_{n_1+\cdots+n_{k-1}+1} = \cdots = \sigma_{n_1+\cdots+n_k} > \sigma_{n_1+\cdots+n_k+1} = \cdots = \sigma_p = \cdots = \sigma_n = 0$. Let $q := n_1 + \cdots + n_k$ and $n_{k+1} := n - q$, so that $n_1 + \cdots + n_{k+1} = n$. Then, global optimal solutions $(U_*, V_*) \in \mathrm{St}(p,m) \times \mathrm{St}(p,n)$ to Problem 4.2.2 form a submanifold diffeomorphic to $\mathcal{M}_0 := O(n_1) \times \cdots \times O(n_k) \times \mathrm{St}(p-q, n_{k+1}) \times \mathrm{St}(p-q, m-q)$. Further, the Hessian $\mathrm{Hess}\, F(U_*, V_*)$ at $(U_*, V_*)$ is degenerate for the tangent subspace of $T_{(U_*,V_*)}(\mathrm{St}(p,m) \times \mathrm{St}(p,n))$ which is isomorphic, as a vector space, to a tangent space to $\mathcal{M}_0$.*

*Proof.* Let $(U, V) \in \mathrm{St}(p,m) \times \mathrm{St}(p,n)$ be a fixed global optimal solution to Problem 4.2.2 and $(U_*, V_*) \in \mathrm{St}(p,m) \times \mathrm{St}(p,n)$ be any global optimal solution. The same discussion as in the proof of Prop. 4.6.1 is carried out to provide Eq. (4.6.4) with $\hat{V} = (V, v_{p+1}, \ldots, v_n)$. However, since $S$ is not invertible, we do not obtain an equation like (4.6.5). Nevertheless, $S_q := \mathrm{diag}\,(s_1, \ldots, s_q)$ is invertible. Then, we can obtain, in place of (4.6.5),

$$U_* = (U_1, U_2), \quad U_1 = AV_* \begin{pmatrix} I_q \\ 0 \end{pmatrix} S_q^{-1}, \quad A^T U_2 = 0. \tag{4.6.13}$$

This implies that if $V_*$ is expressed as $V_* = \hat{V}Q$, a part of $U_*$ or $U_1$-part is determined but the other part or $U_2$-part is never determined, where the columns of $U_2$ are arbitrary $p-q$ orthonormal vectors in $\mathrm{Ker}\, A^T$. Since $\dim \mathrm{Ker}\, A^T = m - \mathrm{rank}\, A = m - q$, we can take $\{\tilde{u}_{q+1}, \ldots, \tilde{u}_m\}$ as an orthonormal basis of $\mathrm{Ker}\, A^T$. Then, a system of $p-q$ orthonormal vectors from $\mathrm{Ker}\, A^T$ is given by

$$U_2 = (\tilde{u}_{q+1}, \ldots, \tilde{u}_m)\, Q_0, \qquad Q_0 \in \mathrm{St}(p-q, m-q). \tag{4.6.14}$$

Thus, any optimal solution $(U_*, V_*)$ is related to $(U, V)$ by the transformation determined by $Q := (Q_1, \ldots, Q_k, Q_{k+1})$ and $Q_0$. Put another way, if $(U, V)$ is fixed, the set of optimal solutions forms a submanifold diffeomorphic to $\mathcal{M}_0$.

A similar computation to (4.6.9) results in

$$\begin{aligned}
&\langle \mathrm{Hess}\, F(U_*, V_*)[(\xi, \eta)], (\xi, \eta)\rangle_{(U_*, V_*)} \\
&= \sum_{i=1}^{q} \sum_{j>i} (\sigma_i \mu_i + \sigma_j \mu_j)^{-1} \left( ((\sigma_i \mu_j + \sigma_j \mu_i)\, b_{ij} - (\sigma_i \mu_i + \sigma_j \mu_j)\, d_{ij})^2 + \left(\sigma_i^2 - \sigma_j^2\right) \left(\mu_i^2 - \mu_j^2\right) b_{ij}^2 \right) \\
&\quad + \sum_{i=1}^{m-p} \sum_{j=1}^{q} \sigma_j \mu_j c_{ij}^2 + \sum_{i=1}^{n-p} \sum_{j=1}^{q} \sigma_j \mu_j e_{ij}^2,
\end{aligned} \tag{4.6.15}$$

where $B,\ C,\ D$, and $E$ come from Eq. (4.6.7). The condition for $\langle \mathrm{Hess}\, F(U_*, V_*)[(\xi, \eta)], (\xi, \eta)\rangle_{(U_*, V_*)}$ to vanish is that $b_{ij}$ with $i, j \in \{1, \ldots, n_1\}$, $\{n_1 + 1, \ldots, n_1 + n_2\}, \ldots, \{n_1 + \cdots + n_{k-1} + 1, \ldots, q\}, \{q+1, \ldots, p\}$ are arbitrary (under $b_{ij} = -b_{ji}$) and the other $b_{ij}$ equal to 0, and $c_{ij}$ with $i = 1, \ldots, m-p,\ j = 1, \ldots, q$ and $e_{ij}$ with $i = 1, \ldots, n-p,\ j = 1, \ldots, q$ are 0 and the other $c_{ij}$ and $e_{ij}$ arbitrary, and $d_{ij}$ with $i, j \in \{q+1, \ldots, p\}$ are arbitrary (under $d_{ij} = -d_{ji}$) and the other $d_{ij}$ are given by

$$d_{ij} = (\sigma_i \mu_i + \sigma_j \mu_j)^{-1} (\sigma_i \mu_j + \sigma_j \mu_i)\, b_{ij}. \tag{4.6.16}$$

In matrix notation, these constants are put in the form

$$B = \begin{pmatrix} B_1 & & & \\ & \ddots & & \\ & & B_k & \\ & & & B_{k+1} \end{pmatrix}, \quad C = \begin{pmatrix} 0 & C_{k+1} \end{pmatrix}, \quad D = \begin{pmatrix} D_B & \\ & D_{k+1} \end{pmatrix}, \quad E = \begin{pmatrix} 0 & E_{k+1} \end{pmatrix}, \tag{4.6.17}$$

where $B_i \in \mathrm{Skew}(n_i), i = 1, \ldots, k,\ B_{k+1} \in \mathrm{Skew}(p-q),\ C_{k+1} \in \mathbb{R}^{(m-p)\times(p-q)},\ D_{k+1} \in$

$\mathrm{Skew}(p-q)$, $E_{k+1} \in \mathbb{R}^{(n-p)\times(p-q)}$, and where $D_B$ is determined by (4.6.16). Therefore, the subspace on which the Hessian is degenerate is isomorphic with

$$\begin{aligned}&\mathrm{Skew}(n_1) \times \cdots \times \mathrm{Skew}(n_k) \times \mathrm{Skew}(p-q) \times \mathbb{R}^{(n-p)\times(p-q)} \times \mathrm{Skew}(p-q) \times \mathbb{R}^{(m-p)\times(p-q)} \\ \simeq &T_{Q_1}O(n_1) \times \cdots \times T_{Q_k}O(n_k) \times T_{Q_{k+1}}\mathrm{St}(p-q, n_{k+1}) \times T_{Q_0}\mathrm{St}(p-q, m-q) \simeq T_{(Q,Q_0)}\mathcal{M}_0.\end{aligned} \tag{4.6.18}$$

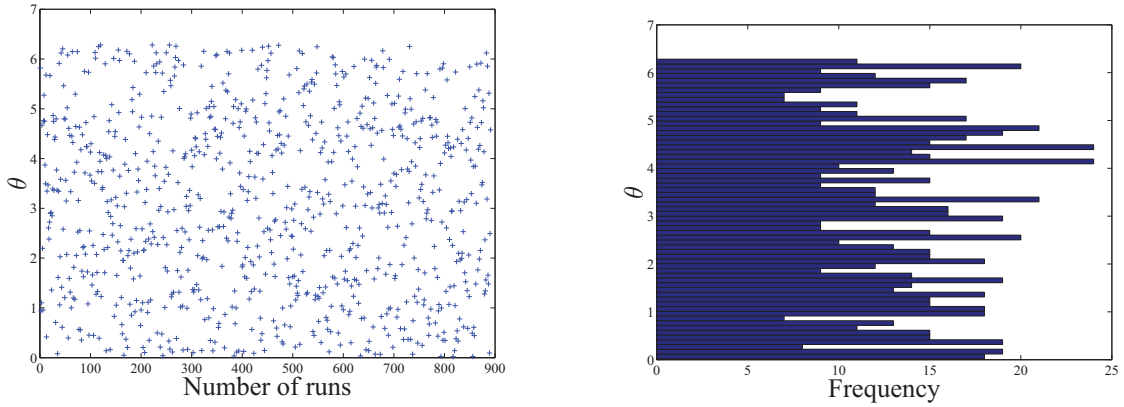
This ends the proof. □

Though we have so far discussed the degeneracy submanifold of global optimal solutions, these propositions hold true even if there is no degeneracy in singular values. If $\sigma_1 > \cdots > \sigma_p > 0$, then the transformation matrix $Q$ given in (4.6.3) takes values in $O(1) \times \cdots \times O(1)$, the product of $p$ copies of $O(1)$. Since $O(1) = \{\pm 1\}$, we have the following corollary.

**Corollary 4.6.1.** *Assume that the largest $p$ singular values $\sigma_1, \ldots, \sigma_p$ of $A \in \mathbb{R}^{m\times n}$ are positive and all distinct, that is, $\sigma_1 > \cdots > \sigma_p > 0$. Then, there are $2^p$ global optimal solutions $(U_*, V_*) \in \mathrm{St}(p,m) \times \mathrm{St}(p,n)$ to Problem 4.2.2 and the Hessian $\mathrm{Hess}\, F(U_*, V_*)$ is positive definite on each $T_{(U_*,V_*)}(\mathrm{St}(p,m) \times \mathrm{St}(p,n))$.*

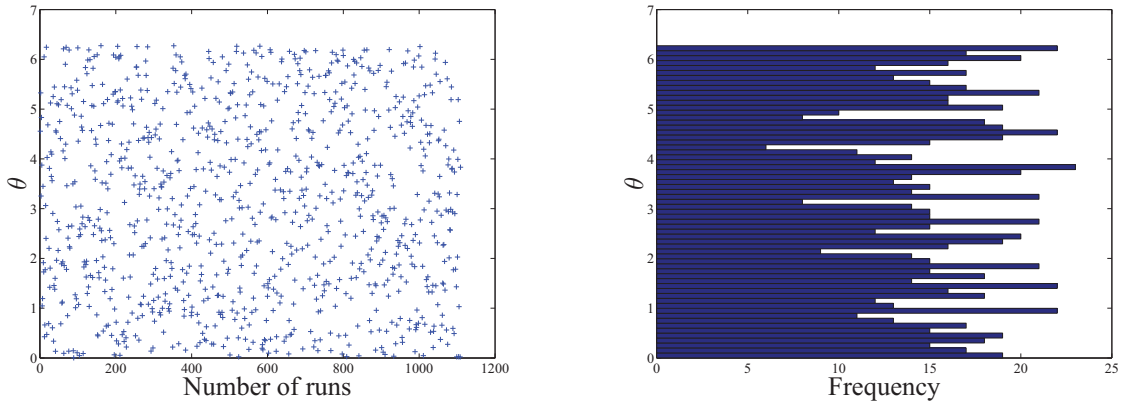
In order to see that sequences generated by Newton's method indeed converge to a set of global optimal solution given by $\mathcal{M}$ of Prop. 4.6.1, we perform Algorithm 4.5.2 for the case of $m = n = 3,\ p = 2,\ A = \mathrm{diag}(1,1,0)$ with various randomly chosen initial points. In this case, Prop. 4.6.1 means that the set of global optimal solutions is diffeomorphic to $\mathcal{M} = O(2)$. One of the global optimal solutions is $(U,V) = \left(\begin{pmatrix} I_2 \\ 0 \end{pmatrix}, \begin{pmatrix} I_2 \\ 0 \end{pmatrix}\right)$. For any global optimal solution $(U_*, V_*)$, there exists $Q \in O(2)$ such that $U_* = UQ,\ V_* = VQ$. The $Q$ is equal to $V^T V_*$ and in one-to-one corresponds to $(U_*, V_*)$.

For $Q \in O(2)$ with $\det Q = 1$, there exists a unique $\theta \in [0, 2\pi)$ such that $Q = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$. If $\det Q = -1$ then $Q = \begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{pmatrix}$. We perform the algorithm with 2000 randomly chosen initial points to observe that the resultant $Q$'s lie all over $O(2)$. Since the target manifold is disconnected, we first compute $\det Q$ and then calculate $\theta$. In our experiment, the 891 of 2000 resultant $Q$'s have the determinant 1 and the others $-1$. Plotting the values of $\theta$ for $Q$ with $\det Q = 1$ results in the left figure (a) of Fig. 4.6.1. The right figure 4.1(b) is a histogram obtained by counting dots in 4.1(a) in each subinterval of $\theta$. Fig. 4.6.2 is for $Q$ with $\det Q = -1$. It turns out that sequences generated by the present algorithm can converge to any point of the set $\mathcal{M}$ of global optimal solutions given in Prop. 4.6.1.




(a) (b)

Figure 4.6.1: Values of $\theta$ corresponding to $Q$ with $\det Q = 1$.




(a) (b)

Figure 4.6.2: Values of $\theta$ corresponding to $Q$ with $\det Q = -1$.

## 4.7 Summary

We have dealt with the problem of the singular value decomposition as an optimization problem on the product manifold $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$ (see Problem 4.2.2). In order to develop optimization algorithms for this problem, we have set up materials such as retractions and the gradient and the Hessian of the objective function $F$ on $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$.

The steepest descent method needs the gradient of $F$, but not the Hessian. Though this algorithm is simple, the convergence is slow. Alternatively, we have set up the conjugate gradient method for the problem, which converges faster. However, the convergence is still slower than that of Newton's method. Newton's method, however, has the drawback that it is likely to converge to a solution not globally but locally. Moreover, Newton's equation (4.4.28) with $p \neq 1$ is difficult to solve.

We have resolved these difficulties as in Algorithm 4.5.2. For speeding up the convergence, we combine Newton's method with the conjugate gradient method, and for diminishing the difficulty in solving (4.4.28), we divide the problem into subproblems in which we use Newton's method with $p = 1$ (Algorithm 4.4.5).

Furthermore, if we know a good approximation of a global optimal solution of Problem 4.2.2 in advance, we need not perform the conjugate gradient method, but only Newton's method. We have compared the singular value decomposition obtained by MATLAB's svd function with that obtained by Newton's method to show that Newton's method generates a better solution.

Our present algorithm deals with the singular value decomposition of a real matrix. For a complex matrix $A \in \mathbb{C}^{m\times n}$, an expected objective function is $-\operatorname{tr}(U^H AVN)$, which is no longer real-valued, where the superscript $H$ means the Hermitian conjugation. A way to generalize the proposed algorithm to a complex case is to define the objective function $F$ by $F(U,V) = -|\operatorname{tr}(U^* AVN)|^2$, but the computation of the gradient and the Hessian of $F$ is not so straightforward. More effective way is to use the objective function of the form $-\operatorname{Re}\left(U^H AVN\right)$. We will further discuss the problem in Chapter 5.

The use of our algorithm depends on the compactness of $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$. Because of compactness, the steepest descent, the conjugate gradient, and Newton's methods can generate converging sequences for an arbitrarily given initial point, though a target point is not necessarily a global optimal point if Newton's method is adopted. To the contrary, if the manifold in question is not compact, we have to guess a domain in which we put an initial point to get a converging sequence.

# Chapter 5

# A Complex Singular Value Decomposition Algorithm Based on the Riemannian Newton Method

## 5.1 Introduction

In Chapter 4, an algorithm for the singular value decomposition of a real matrix has been proposed on the basis of Riemannian optimization methods on the product of two real Stiefel manifolds. The singular value decomposition of a complex matrix, which is also used frequently [AFG91, HC92], has not been formulated as a Riemannian optimization problem. In this chapter, a complex singular value decomposition algorithm based on the Riemannian Newton method is proposed as a generalization of the real singular value decomposition algorithm discussed in Chapter 4.

As is expected, the complex singular value decomposition problem is described as a Riemannian optimization problem on the product of two complex Stiefel manifolds. However, optimization problems with complex variables are somewhat difficult to solve directly. Methods to solve such problems have been still developing. See [SBL12] for a recent research for Euclidean unconstrained optimization of general real-valued objective functions with complex variables. An example of approaches to problems on the complex Stiefel manifold is found in [Man02]. In this chapter, the complex matrix variables are decomposed into the real and imaginary parts in order to reformulate the problem into a real one, so that standard Riemannian Newton's method can be applied. For a general optimization problem with complex variables, such an approach may sometimes disguise the inherent structure of the problem in its original complex form. However, for the problem which is dealt with in this chapter, rewriting into a real one still keeps the problem having the structure of the (real) Stiefel manifold. Newton's method for the resulting real problem is

obtained in a similar manner to the approach in Chapter 4. Further, the proposed algorithm is in turn put into an algorithm on the complex problem. Numerical experiments show that a sequence generated by the final algorithm, if it converges, exhibits a quadratic convergence.

The organization of the chapter is as follows: For feasibility purpose, the problem is equivalently rewritten as a problem on the product of two real manifolds, each of which is an intersection of the real Stiefel manifold and the "quasi-symplectic set" to be defined in Section 5.2. The Riemannian geometry of the real product manifold in question is investigated after Chapter 4 in Section 5.3. In particular, the gradient and the Hessian of the objective function are given together with a retraction map. In this setting, Newton's method on the real product manifold is developed in Section 5.4, which is in turn converted to that on the complex product manifold, and followed by a new complex singular value decomposition algorithm. A numerical experiment is also performed to show that the present algorithm may improve the accuracy of a complex singular value decomposition obtained by an existing method. Moreover, like the algorithm given in Chapter 4, the proposed algorithm divides the problem into easier subproblems, which can be solved in parallel. This chapter concludes with some remarks in Section 5.5.

## 5.2 Complex singular value decomposition and the corresponding Riemannian optimization problem

### 5.2.1 Setting up

Let $m$ and $n$ be positive integers with $m \geq n$. As is discussed in Chapter 4, the singular value decomposition of an $m \times n$ real matrix $A$ is wholly or partly realized by solving an optimization (minimization) problem on $\mathrm{St}(p, m, \mathbb{R}) \times \mathrm{St}(p, n, \mathbb{R})$, where $p$ is an arbitrary positive integer not greater than $n$, $\mathrm{St}(p, n, \mathbb{R})$ is the real Stiefel manifold defined by $\mathrm{St}(p, n, \mathbb{R}) = \{Y \in \mathbb{R}^{n \times p} \,|\, Y^T Y = I_p\}$. The objective function $F_{\mathbb{R}}$ of the problem is

$$F_{\mathbb{R}}(U, V) = -\operatorname{tr}(U^T A V N), \qquad (U, V) \in \mathrm{St}(p, m, \mathbb{R}) \times \mathrm{St}(p, n, \mathbb{R}), \tag{5.2.1}$$

where $N = \operatorname{diag}(\mu_1, \ldots, \mu_p),\ \mu_1 > \cdots > \mu_p > 0$. The solution to the present optimization problem is a pair of matrices whose columns are left and right singular vectors associated with the $p$ largest singular values of $A$, respectively.

We turn to the singular value decomposition of an $m \times n$ complex matrix $A$, which is expressed as

$$A = U \Sigma V^H, \quad U \in U(m),\ V \in U(n),\ \Sigma = \begin{pmatrix} \Sigma_1 \\ \\ 0 \end{pmatrix}, \tag{5.2.2}$$

where $\Sigma_1 = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$, $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$, and where the superscript $H$ denotes the Hermitian conjugation of a matrix. The non-negative real numbers $\sigma_1, \ldots, \sigma_n$ are called the singular values of $A$ and the $j$-th columns of $U$ and $V$ are the left and right singular vectors associated with $\sigma_j$, respectively. In a similar manner to the real case discussed in [AMS08,SI13] (see Chapter 4), we consider a truncated complex singular value decomposition of $A$. The problem is to find the left and right singular vectors associated with $p$ largest singular values of $A$, where $p$ is an arbitrarily fixed integer not greater than $n$.

The truncated complex singular value decomposition is naturally formulated to be an optimization problem on $\mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$, where $\mathrm{St}(p, n, \mathbb{C})$ is the complex Stiefel manifold defined by $\mathrm{St}(p, n, \mathbb{C}) = \left\{ Y \in \mathbb{C}^{n \times p} \,|\, Y^H Y = I_p \right\}$. With the same matrix $N$ as described above, the objective function (5.2.1) would be replaced by $-\mathrm{tr}(U^H A V N)$ with $(U, V) \in \mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$. However, this function is no longer real-valued in general, and not appropriate as an objective function. An alternative objective function is $f(U, V) = -|\mathrm{tr}(U^H A V N)|$. However, as will be shown in Thm. 5.2.1, another real-valued function $F(U, V) = -\,\mathrm{Re}(\mathrm{tr}(U^H A V N))$ is an appropriate objective function, where $\mathrm{Re}(\cdot)$ denotes the real part of the quantity in the parentheses. Further, the function $F$ is better than $f$ from both theoretical and numerical viewpoints. Indeed, if we try to use $f$ as an objective function, we end up with choosing the square of $f$ in computing the gradient and the Hessian with respect to $U$ and $V$. In contrast with this, if $F$ is chosen as an objective function, its gradient and Hessian can be calculated without squaring $F$, and further $F$ consists only of the real part of $\mathrm{tr}(U^H A V N)$ while $f$ includes both the real and imaginary parts. For this reason, $F$ is a better choice for computing requisites. See also the remark to be made below Problem 5.2.3.

Now, we deal with the following optimization problem on $\mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$.

**Problem 5.2.1.**

$$\text{minimize} \quad F(U, V) = -\,\mathrm{Re}(\mathrm{tr}(U^H A V N)), \tag{5.2.3}$$

$$\text{subject to} \quad (U, V) \in \mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C}), \tag{5.2.4}$$

where $N = \mathrm{diag}(\mu_1, \ldots, \mu_p)$, $\mu_1 > \cdots > \mu_p > 0$.

Although the objective function $F$ consists of the real part of $\mathrm{tr}(U^H A V N)$, it works well for finding the truncated singular value decomposition of $A$, as is shown in the following theorem.

**Theorem 5.2.1.** *Let $(U_*, V_*)$ be an optimal solution to Problem 5.2.1. Then, the $j$-th columns of $U_*$ and $V_*$ are the left and right singular vectors of $A$ associated with the $j$-th dominant singular value, respectively. In addition, the $p$ largest singular values $\sigma_1 \geq \cdots \geq \sigma_p$ can be calculated through the formula $U_*^H A V = \mathrm{diag}(\sigma_1, \ldots, \sigma_p)$, i.e., Eq. (5.2.22) with $s_j = \sigma_j$.*

To prove the theorem, we put Problem 5.2.1 into the form of a real problem. We denote $A = B + iC$, $U = X + iY$, and $V = Z + iW$, where $B,\ C \in \mathbb{R}^{m\times n}$ are the real and imaginary parts of $A \in \mathbb{C}^{m\times n}$, respectively, and where $X,\ Y \in \mathbb{R}^{m\times p}$ and $Z,\ W \in \mathbb{R}^{n\times p}$ are those of $U \in \mathbb{C}^{m\times p}$ and $V \in \mathbb{C}^{n\times p}$, respectively. The conditions $U^HU = V^HV = I_p$ for $(U, V) \in \mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$ are written out in terms of $X, Y, Z, W$, and the objective function $-\mathrm{Re}(\mathrm{tr}(U^HAVN))$ are expressed in terms of $B, C, X, Y, Z, W$ as well. Hence, Problem 5.2.1 can be put equivalently in the real form as follows:

**Problem 5.2.2.**

$$\begin{aligned} \text{maximize} \quad & G(X,Y,Z,W) = \mathrm{tr}((X^TBZ - X^TCW + Y^TBW + Y^TCZ)N), && (5.2.5)\\ \text{subject to} \quad & X^TX + Y^TY = Z^TZ + W^TW = I_p, \\ & X^TY - Y^TX = Z^TW - W^TZ = 0. && (5.2.6) \end{aligned}$$

We here introduce the Lagrangian of Problem 5.2.2 by

$$\begin{aligned} & L(X,Y,Z,W,\Lambda,\Omega,\Gamma,\Delta) \\ =& G(X,Y,Z,W) + \mathrm{tr}(\Lambda(X^TX + Y^TY - I_p)) \\ & + \mathrm{tr}(\Omega(Z^TZ + W^TW - I_p)) + \mathrm{tr}(\Gamma(X^TY - Y^TX)) + \mathrm{tr}(\Delta(Z^TW - W^TZ)), \end{aligned} \tag{5.2.7}$$

where $\Lambda, \Omega \in \mathrm{Sym}(p)$ and $\Gamma, \Delta \in \mathrm{Skew}(p)$ are the Lagrange multiplier matrices, and where $\mathrm{Sym}(p)$ and $\mathrm{Skew}(p)$ are the sets of all $p \times p$ real symmetric and skew-symmetric matrices, respectively. Note here that $X^TX + Y^TY - I_p, Z^TZ + W^TW - I_p \in \mathrm{Sym}(p)$ and $X^TY - Y^TX, Z^TW - W^TZ \in \mathrm{Skew}(p)$.

Let $L_X$ denote the partial derivative of $L$ with respect to $X$, and so on. Since $L_X = L_Y = 0$, $L_Z = L_W = 0$, and $L_\Lambda = L_\Omega = L_\Gamma = L_\Delta = 0$ at an optimal solution $(X_*, Y_*, Z_*, W_*, \Lambda_*, \Omega_*, \Gamma_*, \Delta_*)$, we have

$$(BZ_* - CW_*)N + 2X_*\Lambda_* + 2Y_*\Gamma_* = 0, \tag{5.2.8}$$
$$(BW_* + CZ_*)N + 2Y_*\Lambda_* - 2X_*\Gamma_* = 0, \tag{5.2.9}$$
$$(B^TX_* + C^TY_*)N + 2Z_*\Omega_* + 2W_*\Delta_* = 0, \tag{5.2.10}$$
$$(B^TY_* - C^TX_*)N + 2W_*\Omega_* - 2Z_*\Omega_* = 0, \tag{5.2.11}$$
$$X_*^TX_* + Y_*^TY_* = Z_*^TZ_* + W_*^TW_* = I_p, \tag{5.2.12}$$
$$X_*^TY_* - Y_*^TX_* = Z_*^TW_* - W_*^TZ_* = 0. \tag{5.2.13}$$

We return to the proof of the theorem in the complex form. Let $U_* = X_* + iY_*$ and $V_* = Z_* + iW_*$. Note that rewriting Eqs. (5.2.12) and (5.2.13) into the complex forms results in $U_*^HU_* = V_*^HV_* = I_p$. Adding (5.2.8) to (5.2.9) multiplied by $i$, we obtain

$$AV_*N + 2U_*(\Lambda_* - i\Gamma_*) = 0. \tag{5.2.14}$$

Since $U_*^H U_* = I_p$, it follows from (5.2.14) that

$$\Lambda_* - i\Gamma_* = -\frac{1}{2} U_*^H A V_* N. \tag{5.2.15}$$

Substituting (5.2.15) into (5.2.14) yields

$$AV_* = U_* U_*^H A V_*. \tag{5.2.16}$$

Also, since $\Lambda_* \in \mathrm{Sym}(p)$ and $\Gamma_* \in \mathrm{Skew}(p)$, we obtain

$$(\Lambda_* - i\Gamma_*)^H = \Lambda_*^T + i\Gamma_*^T = \Lambda_* - i\Gamma_*, \tag{5.2.17}$$

which implies that $\Lambda_* - i\Gamma_*$ is a Hermitian matrix. Therefore, the right-hand side of (5.2.15) is also Hermitian, so that we have

$$U_*^H A V_* N = N V_*^H A^H U_*. \tag{5.2.18}$$

In a similar manner, Eqs. (5.2.10) and (5.2.11) are put together to eventually give rise to

$$A^H U_* = V_* V_*^H A^H U_*, \tag{5.2.19}$$

$$V_*^H A^H U_* N = N U_*^H A V_*. \tag{5.2.20}$$

From (5.2.18) and (5.2.20), it follows that

$$U_*^H A V_* N^2 = N^2 U_*^H A V_*. \tag{5.2.21}$$

Since $N^2$ is a diagonal matrix, Eq. (5.2.21) implies that $U_*^H A V_*$ is a diagonal matrix as well, which we express as

$$U_*^H A V_* = \mathrm{diag}(s_1, \ldots, s_p). \tag{5.2.22}$$

From (5.2.22) and its Hermitian conjugate, Eq. (5.2.18) is found to take the form

$$\mathrm{diag}(s_1\mu_1, \ldots, s_p\mu_p) = \mathrm{diag}(\bar{s}_1\mu_1, \ldots \bar{s}_p\mu_p), \tag{5.2.23}$$

which implies that $s_j$'s are real numbers. In addition, Eqs. (5.2.16) and (5.2.19) are put together to imply that

$$\begin{aligned} A^H A V_* =& (A^H U_*)(U_*^H A V_*) \\ =& V_* (V_*^H A^H U_*)(U_*^H A V_*) \\ =& V_* \,\mathrm{diag}(s_1, \ldots, s_p)^2 \\ =& V_* \,\mathrm{diag}(s_1^2, \ldots, s_p^2), \end{aligned} \tag{5.2.24}$$

which means that $s_j^2$ are eigenvalues of $A^H A$ and that the $j$-th column of $V_*$ is the corresponding eigenvector. Then, the objective function $G$ regarded as the function of $(U, V) \in \mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$ is evaluated at an optimal solution $(U_*, V_*)$ as

$$G(U_*, V_*) = \mathrm{tr}(U_*^H A V_* N) = \sum_{j=1}^{p} s_j \mu_j. \tag{5.2.25}$$

Since $\mu_1 > \cdots > \mu_p > 0$ and since $(U, V) = (U_*, V_*)$ are supposed to maximize $G(U, V)$, $s_j$'s should be the $p$ largest singular values among all the singular values of $A$ and be ordered as $s_1 \geq \cdots \geq s_p \geq 0$. Therefore, $s_j$ is the $j$-th largest singular value of $A$ and the $j$-th column of $V_*$ is the corresponding right singular vector. Similarly, the $j$-th column of $U_*$ is the left singular vector associated with $s_j$. This completes the proof.

### 5.2.2 Realization of $\mathrm{St}(p, n, \mathbb{C})$ as the intersection of the real Stiefel manifold and the quasi-symplectic set

An $n \times n$ complex matrix $D = E + iF \in \mathbb{C}^{n \times n}$, $E, F \in \mathbb{R}^{n \times n}$, can be expressed as a $2n \times 2n$ real matrix

$$\tilde{D} = \begin{pmatrix} E & F \\ -F & E \end{pmatrix}, \tag{5.2.26}$$

and vice versa. A $2n \times 2n$ matrix $\hat{D}$ has the form (5.2.26) if and only if

$$J_n \hat{D} = \hat{D} J_n, \quad J_n := \begin{pmatrix} 0 & I_n \\ -I_n & 0 \end{pmatrix}. \tag{5.2.27}$$

Further, if $D = E + iF$ is unitary, then the corresponding real matrix $\tilde{D}$ given in (5.2.26) becomes orthogonal, and the condition $J_n \tilde{D} = \tilde{D} J_n$ is equivalently written as $\tilde{D}^T J_n \tilde{D} = J_n$, which implies that $\tilde{D}$ is a symplectic matrix. Let $\mathrm{Sp}(n, \mathbb{R})$ denote the real symplectic group defined by

$$\mathrm{Sp}(n, \mathbb{R}) = \left\{ \tilde{D} \in \mathbb{R}^{2n \times 2n} \mid \tilde{D}^T J_n \tilde{D} = J_n \right\}. \tag{5.2.28}$$

Then, the map

$$\psi^{(n)} : U(n) \to O(2n) \cap \mathrm{Sp}(n, \mathbb{R}); \tag{5.2.29}$$

$$\psi^{(n)}(E+iF) = \begin{pmatrix} E & F \\ -F & E \end{pmatrix}, \tag{5.2.30}$$

gives an isomorphism between $U(n)$ and $O(2n) \cap \mathrm{Sp}(n, \mathbb{R})$.

We generalize the mapping $\psi^{(n)}$ into the rectangular matrix case. We first define $\mathcal{SP}(p, q)$ for integers $p, q$ as

$$\mathcal{SP}(p, q) = \left\{ \tilde{D} \in \mathbb{R}^{2q \times 2p} \,|\, \tilde{D} J_p = J_q \tilde{D} \right\}, \tag{5.2.31}$$

which we call the quasi-symplectic set. Note that if $p = q = n$, then $U(n) \simeq O(2n) \cap \mathrm{Sp}(n, \mathbb{R}) = O(2n) \cap \mathcal{SP}(n, n)$, though $\mathcal{SP}(n, n)$ itself is not identical to $\mathrm{Sp}(n, \mathbb{R})$. The set $\mathbb{C}^{n \times p}$ of all $n \times p$ complex matrices is isomorphic to $\mathcal{SP}(p, n)$ with the isomorphism

$$\phi^{(p,n)} : \mathbb{C}^{n \times p} \to \mathcal{SP}(p, n);\ \phi^{(p,n)}(E+iF) = \begin{pmatrix} E & F \\ -F & E \end{pmatrix}, \tag{5.2.32}$$

where $E, F \in \mathbb{R}^{n \times p}$. Then, the map $\phi^{(p,n)}|_{\mathrm{St}(p,n,\mathbb{C})}$, which is the restriction of $\phi^{(p,n)}$ to the complex Stiefel manifold $\mathrm{St}(p, n, \mathbb{C})$, gives a real expression of $\mathrm{St}(p, n, \mathbb{C})$, which we denote by

$$\mathrm{Stp}(p, n) := \mathrm{St}(2p, 2n, \mathbb{R}) \cap \mathcal{SP}(p, n). \tag{5.2.33}$$

We introduce the set $\mathcal{SP}(n)$ as the collection of $\mathcal{SP}(p, q)$ over all positive integers $p, q \leq n$:

$$\mathcal{SP}(n) = \cup_{\substack{0<p\leq n \\ 0<q\leq n}} \mathcal{SP}(p, q). \tag{5.2.34}$$

Also, we define the map $\phi$ as the collection of $\phi^{(p,q)}$:

$$\phi(B) = \phi^{(p,q)}(B), \qquad B \in \mathbb{C}^{q \times p}. \tag{5.2.35}$$

In what follows, for a square or rectangular complex matrix $B$, we denote the matrix $\phi(B) \in \mathcal{SP}(n)$ by $\tilde{B} = \phi(B)$. Then, matrix operations on matrices without and with tilde are related as follows:

$$B + C \longleftrightarrow \tilde{B} + \tilde{C}, \quad BD \longleftrightarrow \tilde{B}\tilde{D}, \quad B^H \longleftrightarrow \tilde{B}^T, \tag{5.2.36}$$

and the traces of matrices with and without tilde are related by

$$2\,\mathrm{Re}(\mathrm{tr}(E)) = \mathrm{tr}(\tilde{E}), \tag{5.2.37}$$

where $B, C, D$ are complex matrices of appropriate size for addition and multiplication, and where $E$ is a square complex matrix. Note that the set $\mathcal{SP}(n)$ is closed under the operations in the right-hand sides of (5.2.36).

We are now in a position to deal with the complex Stiefel manifold $\mathrm{St}(p, n, \mathbb{C})$ in the real form $\mathrm{Stp}(p, n)$ given in (5.2.33). On account of Eq. (5.2.37), the objective function $F(U, V) = -\mathrm{Re}(\mathrm{tr}(U^H AVN))$ in Problem 5.2.1 is now rewritten as

$$-\mathrm{Re}(\mathrm{tr}(U^H AVN)) = -\frac{1}{2}\mathrm{tr}(\tilde{U}^T \tilde{A}\tilde{V}\tilde{N}). \tag{5.2.38}$$

We remain to use the same symbol $F$ to denote the function of $(\tilde{U}, \tilde{V}) \in \mathrm{Stp}(p, m) \times \mathrm{Stp}(p, n)$ in the right-hand side of (5.2.38). Thus, we are led to the following optimization problem on $\mathrm{Stp}(p, m) \times \mathrm{Stp}(p, n)$.

**Problem 5.2.3.**

$$\text{minimize} \quad F(\tilde{U}, \tilde{V}) = -\frac{1}{2}\mathrm{tr}(\tilde{U}^T \tilde{A}\tilde{V}\tilde{N}), \tag{5.2.39}$$

$$\text{subject to} \quad (\tilde{U}, \tilde{V}) \in \mathrm{Stp}(p, m) \times \mathrm{Stp}(p, n), \tag{5.2.40}$$

where $\tilde{N} = \begin{pmatrix} N & 0 \\ 0 & N \end{pmatrix}$.

We note that the fact that Problem 1 is naturally put into Problem 3 as a real expression shows another merit in choosing $F(U, V) = -\mathrm{Re}(\mathrm{tr}(U^H AVN))$ as an objective function rather than $f(U, V) = -|\mathrm{tr}(U^H AVN)|$.

## 5.3 Riemannian geometry of $\mathrm{Stp}(p, m) \times \mathrm{Stp}(p, n)$

In order to apply Newton's method to Problem 5.2.3, we need the gradient and the Hessian of the objective function $F$ together with a retraction [AMS08] on the product manifold $\mathrm{Stp}(p, m) \times \mathrm{Stp}(p, n)$. In this section, we deal with the Riemannian geometry of $\mathrm{Stp}(p, m) \times \mathrm{Stp}(p, n)$ by employing the results in Chapter 4.

### 5.3.1 Tangent spaces and the orthogonal projection

The tangent space to $\mathrm{Stp}(p, m) \times \mathrm{Stp}(p, n)$ at $(\tilde{U}, \tilde{V})$ is given by

$$\begin{aligned} &T_{(\tilde{U},\tilde{V})}\left(\mathrm{Stp}(p, m) \times \mathrm{Stp}(p, n)\right) \\ &= \left\{ (\tilde{\xi}, \tilde{\eta}) \in \mathcal{SP}(p, m) \times \mathcal{SP}(p, n) \,|\, \tilde{\xi}^T \tilde{U} + \tilde{U}^T \tilde{\xi} = \tilde{\eta}^T \tilde{V} + \tilde{V}^T \tilde{\eta} = 0 \right\}. \end{aligned} \tag{5.3.1}$$

We proceed to endow $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ with a Riemannian metric. The Euclidean space $\mathbb{R}^{2n\times 2p}$ is endowed with the standard inner product

$$(M_1,M_2)=\mathrm{tr}(M_1^TM_2),\qquad M_1,M_2\in\mathbb{R}^{2n\times 2p}. \tag{5.3.2}$$

When restricted to the subspace $\mathcal{SP}(p,n)$ of $\mathbb{R}^{2n\times 2p}$, the inner product takes the form

$$(\tilde{B},\tilde{C})=\mathrm{tr}(\tilde{B}^T\tilde{C})=2\,\mathrm{tr}(B_1^TC_1+B_2^TC_2), \tag{5.3.3}$$

for $\tilde{B}=\begin{pmatrix} B_1 & B_2\\ -B_2 & B_1\end{pmatrix},\tilde{C}=\begin{pmatrix} C_1 & C_2\\ -C_2 & C_1\end{pmatrix}\in\mathcal{SP}(p,n)$. Getting rid of the factor 2 in the right-hand side of (5.3.3), we define the inner product on $\mathcal{SP}(p,n)$ to be

$$\langle\tilde{B},\tilde{C}\rangle=\frac{1}{2}\,\mathrm{tr}(\tilde{B}^T\tilde{C}),\qquad \tilde{B},\tilde{C}\in\mathcal{SP}(p,n). \tag{5.3.4}$$

Then, the manifold $\mathrm{Stp}(p,n)$ as a submanifold of $\mathcal{SP}(p,n)$ is endowed with the induced metric. Further, the product manifold $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ is endowed with the product metric, which is expressed as

$$\langle(\tilde{\xi}_1,\tilde{\eta}_1),(\tilde{\xi}_2,\tilde{\eta}_2)\rangle_{(\tilde{U},\tilde{V})}=\frac{1}{2}\left(\mathrm{tr}(\tilde{\xi}_1^T\tilde{\xi}_2)+\mathrm{tr}(\tilde{\eta}_1^T\tilde{\eta}_2)\right), \tag{5.3.5}$$

for $(\tilde{\xi}_1,\tilde{\eta}_1),(\tilde{\xi}_2,\tilde{\eta}_2)\in T_{(\tilde{U},\tilde{V})}\left(\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)\right)$.

If we regard $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ as a Riemannian submanifold of $\mathcal{SP}(p,m)\times\mathcal{SP}(p,n)$, we can exploit a previous result in Chapter 4 to obtain the following proposition.

**Proposition 5.3.1.** *For any $(\tilde{B},\tilde{C})\in\mathcal{SP}(p,m)\times\mathcal{SP}(p,n)$, the orthogonal projection operator $P_{(\tilde{U},\tilde{V})}$ onto the tangent space $T_{(\tilde{U},\tilde{V})}(\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n))$ at $(\tilde{U},\tilde{V})$ is given by*

$$P_{(\tilde{U},\tilde{V})}(\tilde{B},\tilde{C})=\left(P_{\tilde{U}}(\tilde{B}),P_{\tilde{V}}(\tilde{C})\right), \tag{5.3.6}$$

*where*

$$P_{\tilde{U}}(\tilde{B})=\tilde{B}-\tilde{U}\,\mathrm{sym}\left(\tilde{U}^T\tilde{B}\right), \tag{5.3.7}$$

$$P_{\tilde{V}}(\tilde{C})=\tilde{C}-\tilde{V}\,\mathrm{sym}\left(\tilde{V}^T\tilde{C}\right), \tag{5.3.8}$$

*and where $\mathrm{sym}(\tilde{B}):=(\tilde{B}+\tilde{B}^T)/2$ denotes the symmetric part of $\tilde{B}$.*

*Proof.* On account of the right-hand sides of (5.3.7) and (5.3.8), it is easy to verify that

$P_{(\tilde{U},\tilde{V})}(\tilde{B},\tilde{C}) \in \mathcal{SP}(p,m) \times \mathcal{SP}(p,n)$. The remaining task is to show that

$$P_{\tilde{U}}(\tilde{B})^T\tilde{U} + \tilde{U}^T P_{\tilde{U}}(\tilde{B}) = P_{\tilde{V}}(\tilde{C})^T\tilde{V} + \tilde{V}^T P_{\tilde{V}}(\tilde{C}) = 0 \tag{5.3.9}$$

and

$$\langle (\tilde{B},\tilde{C}) - P_{(\tilde{U},\tilde{V})}(\tilde{B},\tilde{C}), (\tilde{\xi},\tilde{\eta})\rangle_{(\tilde{U},\tilde{V})} = 0 \tag{5.3.10}$$

for any $(\tilde{\xi},\tilde{\eta}) \in T_{(\tilde{U},\tilde{V})}(\mathrm{Stp}(p,m) \times \mathrm{Stp}(p,n))$. Eq. (5.3.9) is an easy consequence of $\tilde{U}^T\tilde{U} = \tilde{V}^T\tilde{V} = I_{2p}$, and Eq. (5.3.10) results from the fact that the trace of the product of symmetric and skew-symmetric matrices is zero. □

### 5.3.2 Retraction

In each iteration of a Riemannian optimization method on a manifold $M$, for a given search direction $\eta \in T_xM$ at a current point $x \in M$, a search should be performed on a curve emanating from $x$ in the direction of $\eta$. For this purpose, it is necessary to find a retraction on the manifold $M$ in question, which is a map from $TM$ to $M$ (see Subsection 2.2.1).

On the real Stiefel manifold $\mathrm{St}(p,n,\mathbb{R})$, there exists a retraction based on the QR decomposition, which is denoted by $R^{\mathbb{R}}$ and defined to be

$$R^{\mathbb{R}}_Y(\xi) = \mathrm{qf}(Y+\xi), \quad Y \in \mathrm{St}(p,n,\mathbb{R}),\ \xi \in T_Y\mathrm{St}(p,n,\mathbb{R}), \tag{5.3.11}$$

where $R^{\mathbb{R}}_Y$ is the restriction of $R^{\mathbb{R}}$ to $T_Y\mathrm{St}(p,n,\mathbb{R})$, and where $\mathrm{qf}(\cdot)$ denotes the Q-factor of the QR decomposition of the matrix in the parentheses (see Section 2.3). However, this $R^{\mathbb{R}}$ cannot apply for the case of $\mathrm{Stp}(p,n)$. This is because even if $\tilde{B} \in \mathcal{SP}(p,n)$, $\mathrm{qf}(\tilde{B})$ no longer belongs to $\mathcal{SP}(p,n)$ in general. An alternative approach is to start with the QR-based retraction $R^{\mathbb{C}}$ on $\mathrm{St}(p,n,\mathbb{C})$, and then to return to $\mathcal{SP}(p,n)$. Here, $R^{\mathbb{C}}$ is defined by

$$R^{\mathbb{C}}_U(\xi) = \mathrm{qf}(U+\xi), \quad U \in \mathrm{St}(p,n,\mathbb{C}),\ \xi \in T_U\mathrm{St}(p,n,\mathbb{C}), \tag{5.3.12}$$

where the qf in (5.3.12) denotes the Q-factor of the complex QR decomposition. That is, if a full-rank $n \times p$ complex matrix $M$ is decomposed into

$$M = QR, \qquad Q \in \mathrm{St}(p,n,\mathbb{C}), R \in S^+_{\mathrm{upp}}(p), \tag{5.3.13}$$

then $\mathrm{qf}(M) = Q$, where $S^+_{\mathrm{upp}}(p)$ denotes the set of all $p \times p$ upper triangular matrices with strictly positive diagonal entries. We then define the QR-based retraction $R^{\mathrm{Stp}}$ on $\mathrm{Stp}(p,n)$ as follows:

$$R^{\mathrm{Stp}}_{\tilde{U}}(\tilde{\xi}) = \phi\left(R^{\mathbb{C}}_{\phi^{-1}(\tilde{U})}(\phi^{-1}(\tilde{\xi}))\right) = \phi\left(R^{\mathbb{C}}_U(\xi)\right), \qquad \tilde{U} \in \mathrm{Stp}(p,n),\ \tilde{\xi} \in T_{\tilde{U}}\mathrm{Stp}(p,n), \tag{5.3.14}$$

where $\phi$ is defined in (5.2.32).

A retraction $\tilde{R}$ on $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ is immediately defined as

$$\tilde{R}_{(\tilde{U},\tilde{V})}(\tilde{\xi},\tilde{\eta}) = \left(R^{\mathrm{Stp}}_{\tilde{U}}(\tilde{\xi}), R^{\mathrm{Stp}}_{\tilde{V}}(\tilde{\eta})\right), \tag{5.3.15}$$
$$(\tilde{U},\tilde{V})\in\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n),\ (\tilde{\xi},\tilde{\eta})\in T_{(\tilde{U},\tilde{V})}(\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)).$$

### 5.3.3 The gradient and the Hessian

The objective function (5.2.39) in Problem 5.2.3 is quite similar to the function (5.2.1) which is investigated in Chapter 4. The only difference is the factor 1/2 in (5.2.39). However, because of the factor 1/2 in the metric (5.3.5) on $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$, the gradient and the Hessian of the current objective function $F$ on $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ have the same forms as those given in Chapter 4.

**Proposition 5.3.2.** *For $(\tilde{U},\tilde{V})\in\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$, let $\tilde{S}_1$ and $\tilde{S}_2$ be defined to be $\tilde{S}_1 = \mathrm{sym}\left(\tilde{U}^T\tilde{A}\tilde{V}\tilde{N}\right)$ and $\tilde{S}_2 = \mathrm{sym}\left(\tilde{V}^T\tilde{A}^T\tilde{U}\tilde{N}\right)$, respectively. Then, the gradient of (5.2.39) at $(\tilde{U},\tilde{V})\in\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ is expressed as*

$$\mathrm{grad}\, F(\tilde{U},\tilde{V}) = \left(\tilde{U}\tilde{S}_1 - \tilde{A}\tilde{V}\tilde{N}, \tilde{V}\tilde{S}_2 - \tilde{A}^T\tilde{U}\tilde{N}\right). \tag{5.3.16}$$

*Further, let $(\tilde{\xi},\tilde{\eta})$ be a tangent vector at $(\tilde{U},\tilde{V})\in\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$. The Hessian of (5.2.39) at $(\tilde{U},\tilde{V})$ is viewed as a linear transformation of the tangent space and given by*

$$\begin{aligned}&\mathrm{Hess}\, F(\tilde{U},\tilde{V})[(\tilde{\xi},\tilde{\eta})]\\ =&\Big(\tilde{\xi}\tilde{S}_1 - \tilde{A}\tilde{\eta}\tilde{N} - \tilde{U}\,\mathrm{sym}\left(\tilde{U}^T(\tilde{\xi}\tilde{S}_1 - \tilde{A}\tilde{\eta}\tilde{N})\right), \tilde{\eta}\tilde{S}_2 - \tilde{A}^T\tilde{\xi}\tilde{N} - \tilde{V}\,\mathrm{sym}\left(\tilde{V}^T\left(\tilde{\eta}\tilde{S}_2 - \tilde{A}^T\tilde{\xi}\tilde{N}\right)\right)\Big).\end{aligned} \tag{5.3.17}$$

## 5.4 Newton's method and a new complex singular value decomposition algorithm

In this section, we develop a new complex singular value decomposition algorithm based on the Riemannian Newton method.

### 5.4.1 Newton's method for Problem 5.2.3

We apply the Riemannian Newton method [AMS08] to Problem 5.2.3. For a tangent vector $(\tilde{\xi},\tilde{\eta})$ to $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ at $(\tilde{U}_k,\tilde{V}_k)\in\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$, Newton's equation takes the form

$$\mathrm{Hess}\, F(\tilde{U}_k,\tilde{V}_k)[(\tilde{\xi},\tilde{\eta})] = -\,\mathrm{grad}\, F(\tilde{U}_k,\tilde{V}_k). \tag{5.4.1}$$

On substituting (5.3.16) and (5.3.17) into (5.4.1), Newton's equation for (5.2.39) can be easily written out. Further, the QR-based retraction $\tilde{R}$ on $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ has been given in (5.3.15). On the basis of these arrangements, Newton's method for Problem 5.2.3 is described as Algorithm 5.4.1.

**Algorithm 5.4.1** Newton's method for Problem 5.2.3

1: Choose an initial point $(\tilde{U}_0,\tilde{V}_0)\in\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$.
2: **for** $k=0,1,2,\ldots$ **do**
3: Compute the search direction $(\tilde{\xi}_k,\tilde{\eta}_k)\in T_{(\tilde{U}_k,\tilde{V}_k)}(\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n))$ by solving Newton's equations

$$\begin{cases} \tilde{\xi}_k\tilde{S}_{1,k}-\tilde{A}\tilde{\eta}_k\tilde{N}-\tilde{U}_k\,\mathrm{sym}\left(\tilde{U}_k^T(\tilde{\xi}_k\tilde{S}_{1,k}-\tilde{A}\tilde{\eta}_k\tilde{N})\right)=\tilde{A}\tilde{V}_k\tilde{N}-\tilde{U}_k\tilde{S}_{1,k}, \\ \tilde{\eta}_k\tilde{S}_{2,k}-\tilde{A}^T\tilde{\xi}_k\tilde{N}-\tilde{V}_k\,\mathrm{sym}\left(\tilde{V}_k^T\left(\tilde{\eta}_k\tilde{S}_{2,k}-\tilde{A}^T\tilde{\xi}_k\tilde{N}\right)\right)=\tilde{A}^T\tilde{U}_k\tilde{N}-\tilde{V}_k\tilde{S}_{2,k}, \end{cases} \tag{5.4.2}$$

where $\tilde{S}_{1,k}=\mathrm{sym}(\tilde{U}_k^T\tilde{A}\tilde{V}_k\tilde{N})$ and $\tilde{S}_{2,k}=\mathrm{sym}(\tilde{V}_k^T\tilde{A}^T\tilde{U}_k\tilde{N})$.
4: Compute the next iterate

$$(\tilde{U}_{k+1},\tilde{V}_{k+1}):=\tilde{R}_{(\tilde{U}_k,\tilde{V}_k)}(\tilde{\xi}_k,\tilde{\eta}_k), \tag{5.4.3}$$

where $\tilde{R}$ is the QR-based retraction on $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ defined in (5.3.15).
5: **end for**

Algorithm 5.4.1 is quite similar to Algorithm 4.4.4 in Chapter 4. In Chapter 4, Newton's equation are divided into a collection of sub-equations by putting $p=1$ and treating the equation on $\mathrm{St}(1,m,\mathbb{R})\times\mathrm{St}(1,n,\mathbb{R})=S^{m-1}\times S^{n-1}$. This makes Newton's equation into a vector equation which is easy to solve. However, this division method does not result in an easy-to-perform algorithm for the present Newton's equation. This is because even for $p=1$, Newton's equations in Algorithm 2.2.1 are still matrix equations for $2m\times 2$ and $2n\times 2$ matrices, $\tilde{\xi}_k$ and $\tilde{\eta}_k$, which are still difficult to solve. Furthermore, as we can observe from (5.2.30), treating matrices on $\mathrm{Stp}(p,n)$ needs twice as much computer memory as those on $\mathrm{St}(p,n,\mathbb{C})$. Also, addition and multiplication of matrices on $\mathrm{Stp}(p,n)$ need about twice as much computation time as those on $\mathrm{St}(p,n,\mathbb{C})$. To avoid these difficulties, we shall put Algorithm 2.2.1 in the complex form in the next subsection.

### 5.4.2 Newton's method for Problem 5.2.1

Through the map $\phi^{-1}$, Newton's method for Problem 5.2.3 can be translated into Newton's method for Problem 5.2.1 on $\mathrm{St}(p,m,\mathbb{C})\times\mathrm{St}(p,n,\mathbb{C})$. In the process of translation, the relations (5.2.36) are used together with the relation for $B\in\mathbb{C}^{p\times p}$ and $\tilde{B}=\phi(B)\in$

$\mathcal{SP}(p,p)$,

$$\mathrm{sym}(\tilde{B}) = \frac{\tilde{B}+\tilde{B}^T}{2} \longleftrightarrow \frac{B+B^H}{2} = \mathrm{her}(B), \tag{5.4.4}$$

where $\mathrm{her}(\cdot)$ denotes the Hermitian part of the matrix in the parentheses. Further, the retraction $\tilde{R}$ given in (5.3.15) on $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ corresponds to the retraction $R$ on $\mathrm{St}(p,m,\mathbb{C})\times\mathrm{St}(p,n,\mathbb{C})$ defined by

$$R_{(U,V)}(\xi,\eta) = \left(R_U^{\mathbb{C}}(\xi), R_V^{\mathbb{C}}(\eta)\right) = (\mathrm{qf}(U+\xi), \mathrm{qf}(V+\eta)), \tag{5.4.5}$$

$$(U,V)\in \mathrm{St}(p,m,\mathbb{C})\times\mathrm{St}(p,n,\mathbb{C}),\ (\xi,\eta)\in T_{(U,V)}(\mathrm{St}(p,m,\mathbb{C})\times\mathrm{St}(p,n,\mathbb{C})). \tag{5.4.6}$$

Thus, Algorithm 2.2.1 is translated to Algorithm 5.4.2 for Problem 5.2.1, which provides Newton's method for Problem 1.

**Algorithm 5.4.2** Newton's method for Problem 5.2.1

1: Choose an initial point $(U_0,V_0)\in\mathrm{St}(p,m,\mathbb{C})\times\mathrm{St}(p,n,\mathbb{C})$.
2: **for** $k=0,1,2,\ldots$ **do**
3: Compute the search direction $(\xi_k,\eta_k)\in T_{(U_k,V_k)}\left(\mathrm{St}(p,m,\mathbb{C})\times\mathrm{St}(p,n,\mathbb{C})\right)$ by solving Newton's equations

$$\begin{cases} \xi_k S_{1,k} - A\eta_k N - U_k\,\mathrm{her}\left(U_k^H(\xi_k S_{1,k} - A\eta_k N)\right) = AV_kN - U_kS_{1,k}, \\ \eta_k S_{2,k} - A^H\xi_k N - V_k\,\mathrm{her}\left(V_k^H\left(\eta_k S_{2,k} - A^H\xi_k N\right)\right) = A^HU_kN - V_kS_{2,k}, \end{cases} \tag{5.4.7}$$

where $S_{1,k}=\mathrm{her}(U_k^HAV_kN)$ and $S_{2,k}=\mathrm{her}(V_k^HA^HU_kN)$.
4: Compute the next iterate

$$(U_{k+1},V_{k+1}) := R_{(U_k,V_k)}(\xi_k,\eta_k), \tag{5.4.8}$$

where $R$ is the QR-based retraction on $\mathrm{St}(p,m,\mathbb{C})\times\mathrm{St}(p,n,\mathbb{C})$ defined in (5.4.6).
5: **end for**

Though Newton's equations in Algorithm 5.4.2 are not easy to solve, the problem can be divided into $p$ subproblems which are easy to solve, as is done in Chapter 4. To this end, we treat Newton's equations with $p=1$ at first. If $p=1$, then $N$ is a positive real number, and hence we may put $N=1$ without loss of generality. Furthermore, one has $U_k^H\xi_k = V_k^H\eta_k = 0$, and $S_{1,k}=S_{2,k}=\mathrm{Re}(U_k^HAV_kN)=\mathrm{Re}(U_k^HAV_k)$, where $U_k,V_k,\xi_k,\eta_k$ are column vectors and $S_k$ is a scalar. In what follows, we replace $U_k,V_k,S_k$ with the lower case symbols $u_k,v_k,s_k$, respectively, since they are no longer matrices. Then, Newton's equations with $p=1$ are written out as

$$s_k\xi_k - A\eta_k + u_k\,\mathrm{Re}(u_k^HA\eta_k) = Av_k - s_ku_k, \tag{5.4.9}$$

$$s_k\eta_k - A^H\xi_k + v_k\,\mathrm{Re}(v_k^HA^H\xi_k) = A^Hu_k - s_kv_k. \tag{5.4.10}$$

If $s_k \neq 0$, (5.4.9) yields

$$\xi_k = s_k^{-1}\left(A(\eta_k + v_k) - u_k \operatorname{Re}(u_k^H A \eta_k)\right) - u_k. \tag{5.4.11}$$

Substituting (5.4.11) into (5.4.10) and simplifying the resulting equation, we obtain the equation for $\eta_k$ without $\xi_k$:

$$\begin{aligned}&(s_k^2 I_n - A^H A)\eta_k + (A^H u_k - s_k v_k)\operatorname{Re}(u_k^H A \eta_k) + v_k \operatorname{Re}(v_k^H A^H A \eta_k)\\ =&A^H A v_k - v_k \operatorname{Re}(v_k^H A^H A v_k).\end{aligned} \tag{5.4.12}$$

Let $B_k := s_k^2 I_n - A^H A \in \mathbb{C}^{n\times n}$ and $a_k := A^H u_k - s_k v_k,\ b_k := A^H u_k,\ c_k := A^H A v_k,\ d_k := A^H A v_k - v_k \operatorname{Re}(v_k^H A^H A v_k) \in \mathbb{C}^n$. In terms of these matrices and vectors, (5.4.12) is rewritten as

$$B_k \eta_k + a_k \operatorname{Re}(b_k^H \eta_k) + v_k \operatorname{Re}(c_k^H \eta_k) = d_k. \tag{5.4.13}$$

We decompose (5.4.13) into its real and imaginary parts by introducing real vectors such as $\eta_k = \eta_k^1 + i\eta_k^2,\ \eta_k^1, \eta_k^2 \in \mathbb{R}^n$. The resultant equation is expressed as

$$\begin{aligned}&B_k^1\eta_k^1 - B_k^2\eta_k^2 + a_k^1(b_k^1)^T\eta_k^1 + a_k^1(b_k^2)^T\eta_k^2 + v_k^1(c_k^1)^T\eta_k^1 + v_k^1(c_k^2)^T\eta_k^2\\ +&i\left(B_k^1\eta_k^2 + B_k^2\eta_k^1 + a_k^2(b_k^1)^T\eta_k^1 + a_k^2(b_k^2)^T\eta_k^2 + v_k^2(c_k^1)^T\eta_k^1 + v_k^2(c_k^2)^T\eta_k^2\right) = d_k^1 + id_k^2.\end{aligned} \tag{5.4.14}$$

We can write out equations for the real and imaginary parts as

$$\left(B_k^1 + a_k^1(b_k^1)^T + v_k^1(c_k^1)^T\right)\eta_k^1 + \left(-B_k^2 + a_k^1(b_k^2)^T + v_k^1(c_k^2)^T\right)\eta_k^2 = d_k^1, \tag{5.4.15}$$

$$\left(B_k^2 + a_k^2(b_k^1)^T + v_k^2(c_k^1)^T\right)\eta_k^1 + \left(B_k^1 + a_k^2(b_k^2)^T + v_k^2(c_k^2)^T\right)\eta_k^2 = d_k^2, \tag{5.4.16}$$

respectively. These equations are put in the form

$$\boldsymbol{A}_k \boldsymbol{\eta}_k = \boldsymbol{d}_k, \tag{5.4.17}$$

where the bold symbols are

$$\boldsymbol{\eta}_k = \begin{pmatrix}\eta_k^1\\ \\ \eta_k^2\end{pmatrix},\ \boldsymbol{d}_k = \begin{pmatrix}d_k^1\\ \\ d_k^2\end{pmatrix},\ \boldsymbol{A}_k = \begin{pmatrix}B_k^1 + a_k^1(b_k^1)^T + v_k^1(c_k^1)^T & -B_k^2 + a_k^1(b_k^2)^T + v_k^1(c_k^2)^T\\ \\ B_k^2 + a_k^2(b_k^1)^T + v_k^2(c_k^1)^T & B_k^1 + a_k^2(b_k^2)^T + v_k^2(c_k^2)^T\end{pmatrix}. \tag{5.4.18}$$

If $\boldsymbol{A}_k$ is invertible, one has

$$\begin{pmatrix}\eta_k^1\\ \\ \eta_k^2\end{pmatrix} = \boldsymbol{\eta}_k = \boldsymbol{A}_k^{-1}\boldsymbol{d}_k, \tag{5.4.19}$$

so that $\eta_k = \eta_k^1 + i\eta_k^2$ is found as well. Once $\eta_k$ is computed, $\xi_k$ is given by (5.4.11). By introducing $\boldsymbol{a}_k = \begin{pmatrix} a_k^1 \\ \\ a_k^2 \end{pmatrix}$, $\boldsymbol{b}_k = \begin{pmatrix} b_k^1 \\ \\ b_k^2 \end{pmatrix}$, $\boldsymbol{c}_k = \begin{pmatrix} c_k^1 \\ \\ c_k^2 \end{pmatrix}$, $\boldsymbol{v}_k = \begin{pmatrix} v_k^1 \\ \\ v_k^2 \end{pmatrix}$, these equations take a simpler form. Now we are led to Algorithm 5.4.3.

---

**Algorithm 5.4.3** Newton's method for Problem 5.2.1 with $p = 1$

---

1: Choose an initial point $(u_0, v_0) \in \mathrm{St}(1, m, \mathbb{C}) \times \mathrm{St}(1, n, \mathbb{C})$.

2: **for** $k = 0, 1, 2, \ldots$ **do**

3: Compute the search direction $(\xi_k, \eta_k) \in T_{(u_k, v_k)} \left(\mathrm{St}(1, m, \mathbb{C}) \times \mathrm{St}(1, n, \mathbb{C})\right)$ by

$$\eta_k = \begin{pmatrix} I_n & iI_n \end{pmatrix} \left( \boldsymbol{B}_k + \begin{pmatrix} \boldsymbol{a}_k & \boldsymbol{v}_k \end{pmatrix} \begin{pmatrix} \boldsymbol{b}_k & \boldsymbol{c}_k \end{pmatrix}^T \right)^{-1} \boldsymbol{d}_k, \tag{5.4.20}$$

$$\xi_k = s_k^{-1} \left( A(\eta_k + v_k) - u_k \,\mathrm{Re}(u_k^H A \eta_k) \right) - u_k, \tag{5.4.21}$$

where $s_k = \mathrm{Re}(u_k^H A v_k)$, $\boldsymbol{b}_k = \begin{pmatrix} \mathrm{Re}(A^H u_k) \\ \\ \mathrm{Im}(A^H u_k) \end{pmatrix}$, $\boldsymbol{a}_k = \boldsymbol{b}_k - s_k \begin{pmatrix} \mathrm{Re}(v_k) \\ \\ \mathrm{Im}(v_k) \end{pmatrix}$, $\boldsymbol{c}_k = \begin{pmatrix} \mathrm{Re}(A^H A v_k) \\ \\ \mathrm{Im}(A^H A v_k) \end{pmatrix}$, $\boldsymbol{d}_k = \boldsymbol{c}_k - \mathrm{Re}(v_k^H A^H A v_k) \begin{pmatrix} \mathrm{Re}(v_k) \\ \\ \mathrm{Im}(v_k) \end{pmatrix}$, $\boldsymbol{B}_k = s_k^2 I_{2n} - \begin{pmatrix} \mathrm{Re}(A^H A) & -\mathrm{Im}(A^H A) \\ \\ \mathrm{Im}(A^H A) & \mathrm{Re}(A^H A) \end{pmatrix}$, and where $\mathrm{Im}(\cdot)$ denotes the imaginary part of the quantity in the parentheses.

4: Compute the next iterate

$$(u_{k+1}, v_{k+1}) := R_{(u_k, v_k)}(\xi_k, \eta_k) = \left( \frac{u_k + \xi_k}{\|u_k + \xi_k\|}, \frac{v_k + \eta_k}{\|v_k + \eta_k\|} \right), \tag{5.4.22}$$

where $\|\cdot\|$ denotes the standard norm on $\mathbb{C}^m$ and $\mathbb{C}^n$.

5: **end for**

---

### 5.4.3 Complex singular value decomposition algorithm based on the Riemannian Newton method

Let $(U_{\mathrm{app}}, V_{\mathrm{app}})$ be a sufficiently accurate approximate solution to Problem 5.2.1 with a general $p$. We denote by $(\cdot)_j$ the $j$-th column of the matrix in the parentheses. Then, for each $j \in \{1, \ldots, p\}$, the pair $((U_{\mathrm{app}})_j, (V_{\mathrm{app}})_j)$ can be considered to be in the convergence region of $((U_*)_j, (V_*)_j)$ for Algorithm 5.4.3, where $(U_*, V_*)$ is an optimal solution to Problem 5.2.1. Then, we can solve each of these $p$ subproblems by applying Algorithm 5.4.3, and eventually solve Problem 5.2.1 after collecting the solutions to the subproblems. We now propose a new complex singular value decomposition algorithm as Algorithm 5.4.4.

---

**Algorithm 5.4.4** Complex singular value decomposition algorithm based on Newton's method for Problem 5.2.1

---

**Require:** A sufficiently accurate approximate solution $(U_{\mathrm{app}}, V_{\mathrm{app}}) \in \mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$ for Problem 5.2.1.

1: **for** $j = 1, 2, \ldots, p$ **do**
2: Set $(u_0, v_0) := ((U_{\mathrm{app}})_j, (V_{\mathrm{app}})_j)$,
3: Perform Steps 2–5 in Algorithm 5.4.3.
4: **end for**
5: Stack the vectors $u_1, \ldots, u_p$ and $v_1, \ldots, v_p$ to form $U$ and $V$, respectively:

$$U = (u_1, \ldots, u_p),\ V = (v_1, \ldots, v_p), \tag{5.4.23}$$

where each $(u_j, v_j)$ is obtained by Step 3.

---

Since the problem is divided into $p$ subproblems, Algorithm 5.4.4 can be performed by parallel $p$ iterations of Algorithm 5.4.3.

A way to obtain an initial approximate solution is to use the MATLAB's svd function. Another method to obtain an approximate solution is to apply the conjugate gradient method for Problem 5.2.1 as in Chapter 4, which we omit to discuss in this chapter. Our method for obtaining initial approximate solutions is as follows: We first make up several test matrices $A$ of which the exact singular value decompositions, hence an optimal solution $(U_*, V_*)$, are available in advance. Then, approximate initial solutions $(U_{\mathrm{app}}, V_{\mathrm{app}})$ are made by adding a pair of matrices with small random elements to the exact solution $(U_*, V_*)$.

We set $m = 300$, $n = 10$, and $p = 5$, and then form unitary matrices $U_{\mathrm{SVD}} \in U(m)$ and $V_{\mathrm{SVD}} \in U(n)$ with randomly chosen elements and fix them in what follows. We proceed to compute $A_j = U_{\mathrm{SVD}} \Sigma_j V_{\mathrm{SVD}}^H$, $\Sigma_j = \begin{pmatrix} D_j \\ \\ 0 \end{pmatrix}$, for the $n \times n$ diagonal matrices $D_j$ given below:

$$D_1 = \mathrm{diag}(10, 9, \ldots, 1), \tag{5.4.24}$$

$$D_2 = \mathrm{diag}(100, 99, \ldots, 92, 1), \tag{5.4.25}$$

$$D_3 = \mathrm{diag}(100, 99, \ldots, 96, 5, 4, \ldots, 1), \tag{5.4.26}$$

$$D_4 = \mathrm{diag}(1000, 999, \ldots, 992, 1), \tag{5.4.27}$$

$$D_5 = \mathrm{diag}(9.64, 8.97, 8.19, 7.77, 5.55, 5.02, 4.23, 4.10, 3.60, 0.29), \tag{5.4.28}$$

where the singular values of $A_5$ (or the diagonal elements of $D_5$) are randomly chosen out of the interval $[0, 10]$. The condition numbers of the matrices $A_j$ are $\mathrm{cond}(A_1) = 10$, $\mathrm{cond}(A_2) = \mathrm{cond}(A_3) = 100$, $\mathrm{cond}(A_4) = 1000$, $\mathrm{cond}(A_5) = 32.92$, respectively. From the very definition of $A_j$, the columns of $U_{\mathrm{SVD}}$ and $V_{\mathrm{SVD}}$ are exactly the left and right singular vectors of $A_j$, $j = 1, \ldots, 5$. Therefore, the $(U_{\mathrm{opt}}, V_{\mathrm{opt}})$ defined by

$$U_{\mathrm{opt}} = U_{\mathrm{SVD}} I_{m,p}, \ V_{\mathrm{opt}} = V_{\mathrm{SVD}} I_{n,p} \tag{5.4.29}$$

is an optimal solution to Problem 5.2.1 with $A = A_j$, where $I_{n,p}$ is defined to be $I_{n,p} = \begin{pmatrix} I_p \\ 0 \end{pmatrix} \in \mathbb{R}^{n \times p}$. An approximate initial solution $(U_{\mathrm{app}}, V_{\mathrm{app}}) \in \mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$ is then formed by

$$U_{\mathrm{app}} = \mathrm{qf}(U_{\mathrm{opt}} + U_{\mathrm{rand}}), \ V_{\mathrm{app}} = \mathrm{qf}(V_{\mathrm{opt}} + V_{\mathrm{rand}}), \tag{5.4.30}$$

where $U_{\mathrm{rand}} \in \mathbb{C}^{m \times p}$ and $V_{\mathrm{rand}} \in \mathbb{C}^{n \times p}$ are randomly chosen matrices with elements less than 0.05 in absolute values. For example, the difference in the values of the objective function is $F(U_{\mathrm{app}}, V_{\mathrm{app}}) - F(U_{\mathrm{opt}}, V_{\mathrm{opt}}) = 12.30$ for the matrix $A_1$. We apply Algorithm 5.4.4 with $(U_{\mathrm{app}}, V_{\mathrm{app}})$ as initial data to obtain Fig. 5.4.1, which shows that the differences between the values $F(U_k, V_k)$ and the minimum values $F(U_{\mathrm{opt}}, V_{\mathrm{opt}})$ of $F$ decrease rapidly against the iteration number $k$ for any test matrices $A_j$. For $A = A_1$, the computer decides that $F(U_6, V_6) - F(U_{\mathrm{opt}}, V_{\mathrm{opt}}) = 0$ within a machine epsilon at $k = 6$. For $A = A_2, A_3, A_4, A_5$, the computer decides that the current iterate at $k = 4, 3, 5, 4$ is an optimal solution within a machine epsilon, respectively. We here note that the fact that each graph in Fig. 5.4.1 ends at some iteration number means that the difference reaches 0 within computer accuracy at the next iterate.

## 5.5 Summary

We have formulated the complex singular value decomposition problem as an optimization problem on $\mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$. After defining the quasi-symplectic set and the manifold $\mathrm{Stp}(p, n)$ as a real realization of the complex Stiefel manifold $\mathrm{St}(p, n, \mathbb{C})$, we have reformulated the optimization problem on $\mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$ as that on the real form $\mathrm{Stp}(p, m) \times \mathrm{Stp}(p, n)$ of $\mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$. In developing Newton's method for the

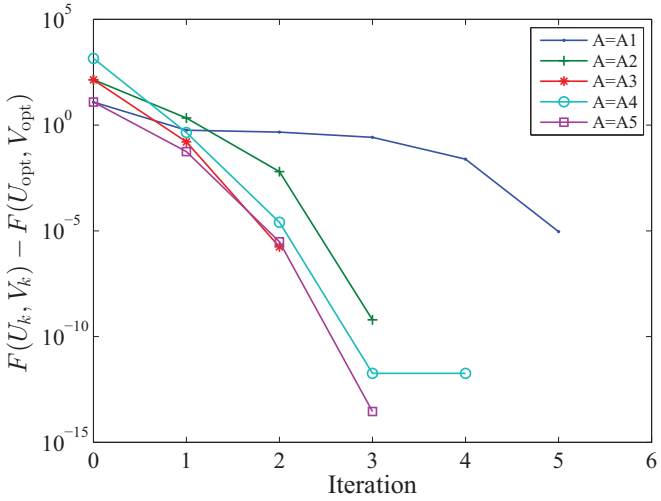



Figure 5.4.1: The differences between the optimal and the current values of the objective functions with $A = A_1, A_2, \ldots, A_5$.

problem on $\mathrm{Stp}(p,m) \times \mathrm{Stp}(p,n)$, the results obtained in Chapter 4 for the real singular value decomposition case have been extensively used. Pulling Newton's method on $\mathrm{Stp}(p,m) \times \mathrm{Stp}(p,n)$ back to that on $\mathrm{St}(p,m,\mathbb{C}) \times \mathrm{St}(p,n,\mathbb{C})$, we have obtained Newton's method for the problem on $\mathrm{St}(p,m,\mathbb{C}) \times \mathrm{St}(p,n,\mathbb{C})$. Though Newton's equation in the algorithm is difficult to solve, the division of the problem into $p$ subproblems and the decomposition of the resultant $p$ equations into the real and imaginary parts make Newton's equations easy to solve.

We have performed numerical experiments with the presented algorithm for several test matrices $A$. The results show that the proposed algorithm can improve a given approximate singular value decomposition within computer accuracy, independently of the condition numbers of the test matrices $A$.

# Chapter 6

# Concluding Remarks

In this chapter, we make some concluding remarks on the topics treated in the thesis together with a further discussion on Riemannian optimization.

After reviewing Euclidean and Riemannian optimization methods in Chapter 2, we have studied Riemannian optimization from both theoretical and application points of view. From the theoretical point of view, the Riemannian conjugate gradient method is studied in Chapter 3, and the matrix singular value decomposition problem is addressed in Chapters 4 and 5 from the application point of view.

In Chapter 3, we have proposed a new Riemannian conjugate gradient method together with the notion of a scaled vector transport. Though the research in Riemannian optimization has been currently developing, a global convergence property of the standard Fletcher-Reeves type Riemannian conjugate gradient method had not been discussed before [RW12]. In view of the fact that in order for the method in [RW12] to have the global convergence property, the vector transport in question needs to be assumed not to increase the norm of the previous search direction vector, we have introduced the notion of scaled vector transport and proposed a scaled Fletcher-Reeves type Riemannian conjugate gradient method (Algorithm 3.3.2) which possesses a global convergence property. The assumption made in [RW12] becomes unnecessary, since the scaling of the tangent vector is performed only when the differentiated-retraction vector transport increases the norm of the search direction. The scaled vector transport is no longer a vector transport in its original sense because of the lack of the linearity property. Our algorithm is nevertheless well defined and needs only a very mild computational overhead per iteration, since we have only to compute the norm of a tangent vector at each iterate in addition to the procedure of the standard algorithm. We have shown the global convergence property of the algorithm by the use of the property that the scaled vector transport no longer increases the norm of the transported tangent vector. Further, we have performed some numerical experiments to verify the practical utility of the present algorithm.

In Chapter 4, we have developed a new real singular value decomposition algorithm on the basis of Riemannian optimization. We have proved that performing the (truncated)

singular value decomposition of a real matrix is equivalent to minimizing a certain function on the product of two Stiefel manifolds. Then, we have calculated several requisites for Riemannian optimization, such as a retraction, and the gradient and the Hessian of the objective function. Using these materials, we have developed the steepest descent, the conjugate gradient, and Newton's methods for the optimization problem in question. If a sequence generated by Newton's method converges, the speed of convergence is very fast. However, Newton's method does not have a global convergence property. If an algorithm with a global convergence property is combined with Newton's method, the combination will exhibit a global convergence property with fast convergence speed. As is well known, the speed of convergence of sequences generated by the steepest descent method is too slow especially in a vicinity of a target point, though the method has a global convergence property. A substitute for the steepest descent method is the conjugate gradient method, which is expected to have a global convergence property better than the steepest descent method. We have proposed a hybrid algorithm composed of Newton's and the conjugate gradient methods. In particular, the scaled Fletcher-Reeves type conjugate gradient method and Newton's method have been put together, which we have proposed in Chapter 3, to show that the hybrid algorithm has a global convergence property with fast convergence speed. The difficulty in the computation of the search direction in Newton's part of the algorithm consists in solving the too complicated Newton's equation. Instead of solving Newton's equation directly, we have proposed an approach in which the complicated Newton's equations are divided into equations to be easily solved iteratively. Further, we have discussed the case where the target matrix has degenerate singular values among the $p$ smallest ones. Though optimal solutions are not isolated in such a case, we have intensively studied the solution set and verified that the proposed algorithm works well as far as numerical experiments suggest.

In Chapter 5, we have generalized both the real optimization problem and the Newton's method proposed in Chapter 4 to those in the complex case. We have formulated the complex singular value decomposition problem as a Riemannian optimization problem on the product of two complex Stiefel manifolds. We have shown that the optimization problem is indeed equivalent to the complex singular value decomposition problem. However, such a problem cannot be solved by using standard optimization algorithms, including Newton's method, since they have been developed on real manifolds. In order to perform the algorithm, we have put the problem into an equivalent problem posed on a corresponding real product manifold. This enables us to get around the difficulty and to propose Newton's method for the problem on the real product manifold. Pulling back the proposed Newton's method onto the initial complex product manifold, we have also presented the algorithm based on Newton's method, which is directly applicable to the problem on the complex product manifold. As is expected, the difficulty with performing the algorithm consists in solving Newton's equation. In a similar manner to that in Chapter 4, we have divided the problem into subproblems, in which corresponding Newton's equation is easy to solve.

We point out that both Newton's methods in Chapters 4 and 5 can be parallel computed. Numerical experiments for several target matrices with different condition numbers have shown that the proposed algorithm can refine a given approximate optimal solution to be a more accurate solution.

In concluding the last section, it is worth pointing out that geometric concepts make the proposed algorithms accurate. In particular, we want to bring attention back to the definition of Hessian.

Accordingly, we note that Riemannian Newton's equation is not an equation obtained by just projecting Euclidean Newton's equation to the tangent space at the current iterate to the search manifold in question. For the sake of simplicity, we turn back to the Rayleigh quotient minimization problem on the sphere (see Problems 1.1.3 and 1.1.4). We first derive the correct Riemannian Newton's equation for the objective function $f(x) := x^T Ax$ on $S^{n-1}$. The $S^{n-1}$ is endowed with the induced metric $\langle \cdot, \cdot \rangle$ from the natural inner product on $\mathbb{R}^n$ as in (3.5.8). Since the sphere $S^{n-1}$ is a special case of the Stiefel manifold $\mathrm{St}(p, n)$ with $p = 1$, it follows from Prop. 2.3.3 that the orthogonal projection operator $P_x$ is given, for any vector $y \in \mathbb{R}^n$, by

$$P_x(y) = (I_n - xx^T)y. \tag{6.0.1}$$

We can regard the operator $P_x$ as the matrix $I_n - xx^T$. Since $S^{n-1}$ is viewed as a submanifold of $\mathbb{R}^n$ with the standard induced metric, the gradient $\mathrm{grad}\, f(x)$ of $f$ at $x$ is simply the projected Euclidean gradient $f_x(x)$ onto the tangent space $T_x S^{n-1}$, that is,

$$\mathrm{grad}\, f(x) = P_x(f_x(x)) = 2(I_n - xx^T)Ax. \tag{6.0.2}$$

In contrast with this, for any tangent vector $\xi \in T_x S^{n-1}$, the tangent vector $\mathrm{Hess}\, f(x)[\xi]$, which is obtained by operating $\xi$ with the Hessian $\mathrm{Hess}\, f(x)$, is not equal to the projection $P_x(f_{xx}(x)\xi)$ of the vector $f_{xx}(x)\xi$ obtained by operating $\xi$ with the Euclidean Hessian. In a similar manner to that in the course of the proof of Prop. 4.3.5, the Hessian $\mathrm{Hess}\, f(x)$ proves to act on $\xi \in T_x S^{n-1}$ as

$$\mathrm{Hess}\, f(x)[\xi] = 2\left(I_n - xx^T\right)\left(A - (x^T Ax)I_n\right)\xi. \tag{6.0.3}$$

To see the difference between $\mathrm{Hess}\, f(x)[\xi]$ and $P_x(f_{xx}(x)\xi)$, we take $f_{xx}(x) = 2A$ into account and rewrite Eq. (6.0.3) as

$$\mathrm{Hess}\, f(x) = P_x\left(f_{xx}(x)\xi\right) - 2(x^T Ax)\left(I_n - xx^T\right)\xi. \tag{6.0.4}$$

The correction term $-2(x^T Ax)\left(I_n - xx^T\right)$ to $P_x\left(f_{xx}(x)\xi\right)$ is necessary for the quadratic convergence property of Newton's method. The proper Newton's equation is expressed as

$$2\left(I_n - xx^T\right)\left(A - (x^T Ax)I_n\right)\xi = -2\left(I_n - xx^T\right)Ax, \tag{6.0.5}$$

which is put in the form

$$P_x\left(\left(A-(x^TAx)I_n\right)(\xi+x)\right)=0. \tag{6.0.6}$$

This means that the vector $\left(A-(x^TAx)I_n\right)(\xi+x)$ is in the normal space $N_xS^{n-1}=\operatorname{span}\{x\}_{\mathbb{R}}$, so that there exists a real number $\alpha$ such that

$$\left(A-(x^TAx)I_n\right)(\xi+x)=\alpha x. \tag{6.0.7}$$

If $A-(x^TAx)I_n$ is invertible, $\alpha$ turns out to be $\alpha=\left(x^T\left(A-(x^TAx)I_n\right)^{-1}x\right)^{-1}$. Thus, Newton's equation has the unique solution

$$\xi=-x+\frac{1}{\left(x^T\left(A-(x^TAx)I_n\right)^{-1}x\right)}\left(A-(x^TAx)I_n\right)^{-1}x, \tag{6.0.8}$$

if and only if the matrix $A-(x^TAx)I_n$ is invertible. Therefore, in terms of the retraction

$$R_x(\xi)=\frac{x+\xi}{\|x+\xi\|}, \tag{6.0.9}$$

the updating formula in Newton's method is given by

$$x_{k+1}=R_{x_k}(\xi_k)=\frac{x_k+\xi_k}{\|x_k+\xi_k\|}=\frac{\left(A-(x_k^TAx_k)I_n\right)^{-1}x_k}{\|\left(A-(x_k^TAx_k)I_n\right)^{-1}x_k\|}. \tag{6.0.10}$$

We perform a numerical experiment with an example matrix of the form $A=P\operatorname{diag}(1,2,\dots,10)P^T$, where $P$ is a $10\times 10$ orthogonal matrix with randomly chosen elements. Let $x_+$ and $x_-$ denote the first column of $P$ and its negative, respectively. We note that they are the optimal solutions to the optimization problem. An initial point $x_0$ is chosen by $x_0=x_++x_{\text{rand}}$ with $\|x_++x_{\text{rand}}\|=1$, where $x_{\text{rand}}$ is a small vector with randomly chosen elements. For the sequence $\{x_k\}$ generated by (6.0.10), we have $\|x_0-x_+\|=0.025$, $\|x_1-x_-\|=3.1\times10^{-8}$, $\|x_2-x_+\|=5.0\times10^{-16}$, which show that the sequence $\{x_k\}$ quickly approaches the optimal solutions, though the target points are alternating between $x_+$ and $x_-$ (see Fig. 6.0.1). This alternation is explained from the following observation: We put $A$ in the form $A=P\operatorname{diag}(\lambda_1,\dots,\lambda_n)P^T$ with $\lambda_1\le\cdots\le\lambda_n$, where $P\in O(n)$. Suppose that the current iterate $x_k$ is close to $x_+$. Then, the Rayleigh quotient $x_k^TAx_k$ is just a little larger than the minimum $\lambda_1$ of the objective function $f$, so that it can be expressed as $x_k^TAx_k=\lambda_1+\epsilon$ with a small number $\epsilon>0$. If $\lambda_2-\lambda_1\gg\epsilon$, then the numerator of the right-hand side of (6.0.10) is estimated as

$$\begin{aligned}\left(A-(x_k^TAx_k)I_n\right)^{-1}x_k&\approx\left(P\operatorname{diag}(-\epsilon,\lambda_2-\lambda_1,\dots,\lambda_n-\lambda_1)P^T\right)^{-1}x_k\\&=P\operatorname{diag}(-\epsilon^{-1},(\lambda_2-\lambda_1)^{-1},\dots,(\lambda_n-\lambda_1)^{-1})P^Tx_k\end{aligned}$$

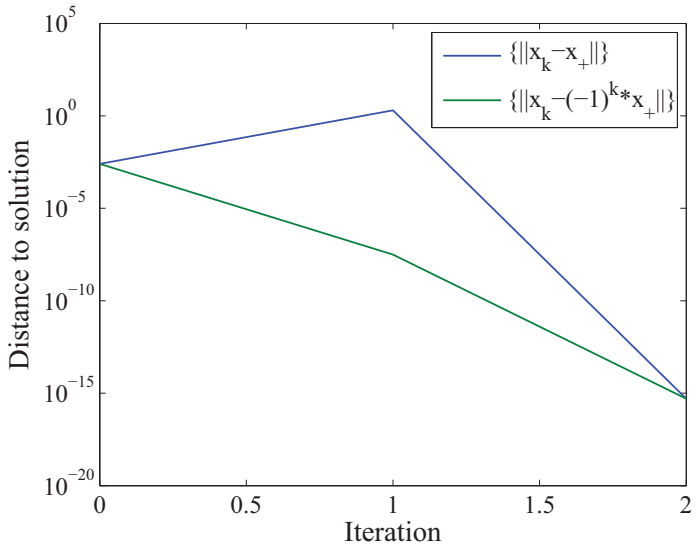



Figure 6.0.1: The sequences of the distances between optimal solutions and the sequence $\{x_k\}$ generated by Riemannian Newton's iteration (6.0.10). One graph shows the sequence $\{\|x_k - x_+\|\}$, which shows that the sequence $\{x_k\}$ approaches $x_+$ and $x_-$ alternately. The other graph describing the sequence $\{\|x_k - (-1)^k x_+\|\}$ shows that if $k$ ranges over even numbers, the subsequence $\{x_k\}$ converges to $x_+$, and if $k$ ranges over odd numbers, the subsequence $\{x_k\}$ converges to $x_-$.

$$\approx P\,\mathrm{diag}(-\epsilon^{-1}, (\lambda_2-\lambda_1)^{-1}, \ldots, (\lambda_n-\lambda_1)^{-1}) \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$= -\epsilon^{-1} x_+, \tag{6.0.11}$$

and the next point is given by and approximated as

$$x_{k+1} = \frac{\left(A-(x^TAx)I_n\right)^{-1} x_k}{\|(A-(x^TAx)I_n)^{-1} x_k\|} \approx \frac{-\epsilon^{-1}x_+}{\epsilon^{-1}} = -x_+ = x_-. \tag{6.0.12}$$

Beside the correct Riemannian Newton's method, we in turn try to use $P_x(f_{xx}(\xi))$ instead of $\mathrm{Hess}\, f(x)[\xi]$, where $\xi \in T_xS^{n-1}$. The improper Newton's equation in place of (6.0.5) is described as

$$2(I_n - xx^T)A\xi = -2(I_n - xx^T)Ax. \tag{6.0.13}$$

Suppose that $A$ is invertible. Then, Eq. (6.0.13) can be solved to give

$$\xi = -x + \frac{A^{-1}x}{x^TA^{-1}x}. \tag{6.0.14}$$

The updating formula with $P_x(f_{xx}(\xi))$ is then given by

$$x_{k+1} = \frac{x_k+\xi_k}{x_k+\xi_k} = \frac{A^{-1}x_k}{\|A^{-1}x_k\|}. \tag{6.0.15}$$

We perform a numerical experiment with the same matrix $A$ as before and obtain Fig. 6.0.2, which shows that the sequence $\{x_k\}$ generated by (6.0.15) converges to an optimal solution $x_+$, but the speed of convergence is much slower than that of (6.0.10).

It is worth pointing out that the updating formulas (6.0.10) and (6.0.15) are already known as the Rayleigh quotient iteration for $A$ and the power method for $A^{-1}$, respectively. We here note that the power method is a method for finding the largest eigenvalue in absolute value of the target matrix. Among eigenvalue decomposition methods in numerical linear algebra, the Rayleigh quotient iteration is known to have a better convergence property than the power method. The quadratic convergence property of the Rayleigh quotient iteration is shown in [Ost57].

We further examine another method for the eigenvalue problem. In [Ber99], a Newton-like method is introduced for a constrained Euclidean optimization problem. In this

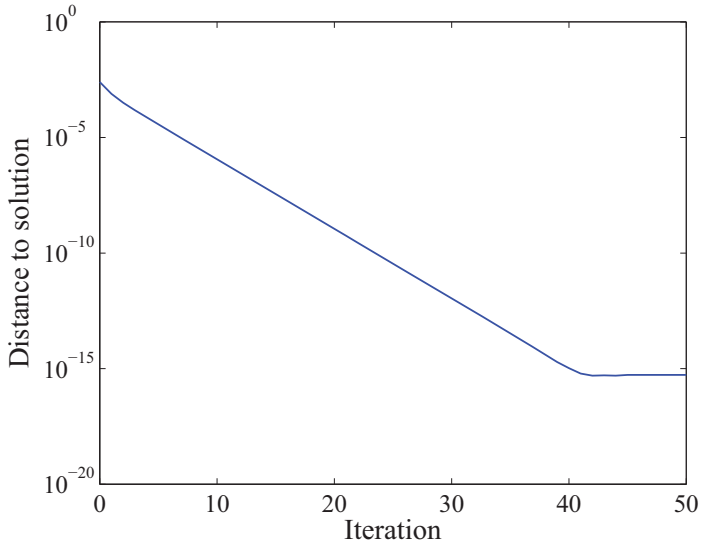



Figure 6.0.2: The sequence of the distances between $x_k$ and $x_+$ with respect to the sequence $\{x_k\}$ generated by (6.0.15).

method, Euclidean Newton's iteration is performed for a Lagrangian function with the constraint taken into account. For the sake of comparison, we apply that method to Problem 1.1.3, in which the Rayleigh quotient objective function $f$ is to be minimized under the constraint $x^T x = 1$. The Lagrangian $L$ is defined to be

$$L(x, \lambda) := x^T A x + \lambda(x^T x - 1), \tag{6.0.16}$$

where $\lambda$ is the Lagrange multiplier. Let $y$ denote $y = \begin{pmatrix} x \\ \\ \lambda \end{pmatrix}$. The Euclidean gradient $L_y$ and the Hessian matrix $L_{yy}$ of $L$ are calculated as

$$L_y(y) = \begin{pmatrix} 2(A + \lambda I_n)x \\ \\ x^T x - 1 \end{pmatrix}, \qquad L_{yy}(y) = \begin{pmatrix} 2(A + \lambda I_n) & 2x \\ \\ 2x^T & 0 \end{pmatrix}, \tag{6.0.17}$$

respectively. Newton's equation $L_{yy}(y)\begin{pmatrix}\xi\\ \\ \mu\end{pmatrix} = -L_y(y)$ then takes the form

$$\begin{pmatrix}(A+\lambda I_n) & x\\ \\ x^T & 0\end{pmatrix}\begin{pmatrix}\xi\\ \\ \mu\end{pmatrix} = -\begin{pmatrix}(A+\lambda I_n)x\\ \\ \frac{1}{2}(x^Tx-1)\end{pmatrix}, \tag{6.0.18}$$

where $\begin{pmatrix}\xi\\ \\ \mu\end{pmatrix} \in \mathbb{R}^n \times \mathbb{R}$ is a Newton vector at $y = \begin{pmatrix}x\\ \\ \lambda\end{pmatrix} \in \mathbb{R}^n \times \mathbb{R}$. If the coefficient matrix $\begin{pmatrix}(A+\lambda I_n) & x\\ \\ x^T & 0\end{pmatrix}$ is invertible, then Eq. (6.0.18) is solved to give rise to the updating formula

$$\begin{pmatrix}x_{k+1}\\ \\ \lambda_{k+1}\end{pmatrix} = \begin{pmatrix}x_k+\xi_k\\ \\ \lambda_k+\mu_k\end{pmatrix} = \begin{pmatrix}x_k\\ \\ \lambda_k\end{pmatrix} - \begin{pmatrix}(A+\lambda_k I_n) & x_k\\ \\ x_k^T & 0\end{pmatrix}^{-1}\begin{pmatrix}(A+\lambda_k I_n)x_k\\ \\ \frac{1}{2}(x_k^Tx_k-1)\end{pmatrix}. \tag{6.0.19}$$

A numerical experiment with the same matrix as before is performed with this updating formula to obtain Fig. 6.0.3.

The Figs. 6.0.1, 6.0.2, and 6.0.3 are put together into Fig. 6.0.4 for the sake of comparison among the performances of the three methods. We can observe that Riemannian Newton's method generates the fastest sequence of the three sequences.

What we have discussed so far tells us that Riemannian optimization has a great potential to solve various problems effectively, and the theory of the differential geometry plays a core role in it. It turns out that geometric methods are intrinsic to constrained problems, and provide accurate algorithms in comparison with those methods devised extrinsically for incorporating constraints. The studies in this thesis are contributions to geometric methods. The field of Riemannian optimization will be still developing from both theoretical and application sides.

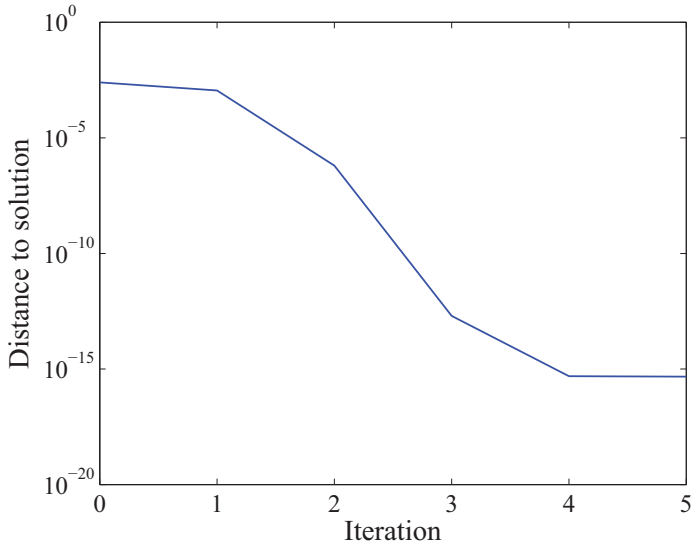



Figure 6.0.3: The sequence of the distances between $x_k$ and $x_+$ with respect to the sequence $\{x_k\}$ generated by (6.0.19).

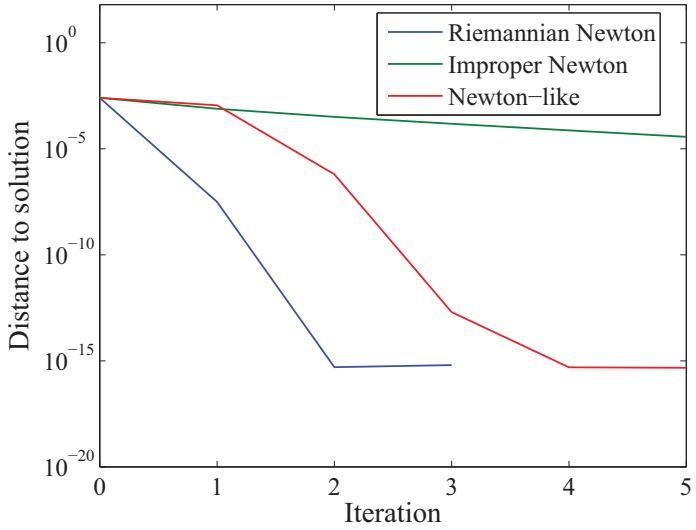



Figure 6.0.4: The sequences of the distances between $x_k$ and $x_+$ with respect to the sequences $\{x_k\}$ generated by (6.0.10), (6.0.15), and (6.0.19).

# Bibliography


[AB85] M. Al-Baali. Descent property and global convergence of the Fletcher-Reeves method with inexact line search. *IMA Journal of Numerical Analysis*, **5**(1):121–124, 1985.

[ABM08] F. Alvarez, J. Bolte, and J. Munier. A unifying local convergence result for Newton's method in Riemannian manifolds. *Foundations of Computational Mathematics*, **8**(2):197–226, 2008.

[AFG91] G. Adams, A. Finn, and M. Griffin. A fast implementation of the complex singular value decomposition on the Connection Machine. In *Proceedings of the Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference*, pages 1129–1132. IEEE, 1991.

[AG03] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical Programming*, **95**(1):3–51, 2003.

[All07] G. Allaire. *Numerical Analysis and Optimization*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, 2007.

[AM12] P.-A. Absil and J. Malick. Projection-like retractions on matrix manifolds. *SIAM Journal on Optimization*, **22**(1):135–158, 2012.

[AMS08] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008.

[Ber99] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1999.

[Bro70] C. G. Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, **6**(1):76–90, 1970.

[Bro93] R. W. Brockett. Differential geometry and the design of gradient algorithms. In *Proceedings of Symposia in Pure Mathematics*, vol. **54**, pages 69–92, 1993.

[CA02] A. Cichocki and S. Amari. *Adaptive Blind Signal and Image Processing*. John Wiley Chichester, 2002.

[Car99] J.-F. Cardoso. High-order contrasts for independent component analysis. *Neural computation*, **11**(1):157–192, 1999.

[CS93] J.-F. Cardoso and A. Souloumiac. Blind beamforming for non-gaussian signals. In *Radar and Signal Processing, IEE Proceedings F*, vol. **140**, pages 362–370. IET, 1993.

[Diw08] U. Diwekar. *Introduction to Applied Optimization*. Springer, 2008.

[DS83] J. J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, 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, 1998.

[FGW02] A. Forsgren, P. E. Gill, and M. H. Wright. Interior methods for nonlinear optimization. *SIAM Review*, **44**(4):525–597, 2002.

[FH10] W. Forst and D. Hoffmann. *Optimization: Theory and Practice*. Springer, 2010.

[Fle70] R. Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, **13**(3):317–322, 1970.

[Fle13] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, 2013.

[FR64] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *The Computer Journal*, **7**(2):149–154, 1964.

[GGT04] G. Giorgi, A. Guerraggio, and J. Thierfelder. *Mathematics of Optimization: Smooth and Nonsmooth Case: Smooth and Nonsmooth Case*. Elsevier, 2004.

[GHN01] N. I. Gould, M. E. Hribar, and J. Nocedal. On the solution of equality constrained quadratic programming problems arising in optimization. *SIAM Journal on Scientific Computing*, **23**(4):1376–1395, 2001.

[Gol70] D. Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of Computation*, **24**(109):23–26, 1970.

[GVL12] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 2012.

[HC92] N. D. Hemkumar and J. R. Cavallaro. A systolic VLSI architecture for complex SVD. In *Proceedings of IEEE International Symposium on Circuits and Systems, 1992. ISCAS'92., 1992*, pages 1061–1064. IEEE, 1992.

[Hes69] M. R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, **4**(5):303–320, 1969.

[HHT07] U. Helmke, K. Hüper, and J. Trumpf. Newton's method on graßmann manifolds. *arXiv preprint arXiv:0709.2205*, 2007.

[HM94] U. Helmke and J. B. Moore. *Optimization and Dynamical Systems*. Communications and Control Engineering Series, Springer-Verlag (London and New York), 1994.

[HS52] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, **49**(6), 1952.

[Kar84] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311. ACM, 1984.

[Kel99] C. T. Kelley. *Iterative Methods for Optimization*. SIAM, 1999.

[KPP04] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.

[LLKS85] E. L. Lawler, J. K. Lenstra, A. R. Kan, and D. B. Shmoys. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley Chichester, 1985.

[LY08] D. G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. Springer, 2008.

[Man02] J. H. Manton. Optimization algorithms exploiting unitary constraints. *IEEE Transactions on Signal Processing*, **50**(3):635–650, 2002.

[NW88] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley New York, 1988.

[NW06] J. Nocedal and S. Wright. *Numerical Optimization, Series in Operations Research and Financial Engineering*. 2006.

[Ost57] A. M. Ostrowski. On the convergence of the Rayleigh quotient iteration for the computation of the characteristic roots and vectors. I. *Archive for Rational Mechanics and Analysis*, **1**(1):233–241, 1957.

[PM76] U. G. Palomares and O. L. Mangasarian. Superlinearly convergent quasi-Newton algorithms for nonlinearly constrained optimization problems. *Mathematical Programming*, **11**(1):1–13, 1976.

[Pow73] M. J. Powell. On search directions for minimization algorithms. *Mathematical Programming*, **4**(1):193–201, 1973.

[Rus06] A. P. Ruszczyński. *Nonlinear optimization*. Princeton university press, 2006.

[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, 2012.

[SBL12] L. Sorber, M. V. Barel, and L. D. Lathauwer. Unconstrained optimization of real functions in complex variables. *SIAM Journal on Optimization*, **22**(3):879–898, 2012.

[Sch03] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003.

[Sha70] D. F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, **24**(111):647–656, 1970.

[SI13] H. Sato and T. Iwai. A Riemannian optimization approach to the matrix singular value decomposition. *SIAM Journal on Optimization*, **23**(1):188–212, 2013.

[SIara] H. Sato and T. Iwai. A complex singular value decomposition algorithm based on the riemannian newton method. In *Proceedings of the 52nd IEEE Conference on Decision and Control*. IEEE, to appear.

[SIarb] H. Sato and T. Iwai. A new, globally convergent Riemannian conjugate gradient method. *Optimization*, to appear.

[Smi94] S. T. Smith. Optimization techniques on Riemannian manifolds. In *Hamiltonian and Gradient Flows, Algorithms and Control*, pages 113–135, 1994.

[Sny05] J. A. Snyman. *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. Springer, 2005.

[TBI97] L. N. Trefethen and D. Bau III. *Numerical Linear Algebra*. SIAM, 1997.

[TCA09] F. Theis, T. P. Cason, and P.-A. Absil. Soft dimension reduction for ica by joint diagonalization on the Stiefel manifold. In *Proceedings of the 8th International Conference on Independent Component Analysis and Signal Separation*, ICA '09, pages 354–361, Berlin, Heidelberg, 2009. Springer-Verlag.

[VB96] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, **38**(1):49–95, 1996.

[WB12] S.-G. Wang and L. Bai. Flexible robust sliding mode control for uncertain stochastic systems with time-varying delay and structural uncertainties. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 1536–1541. IEEE, 2012.

[WS97] M. Wax and J. Sheinvald. A least-squares approach to joint diagonalization. *IEEE Signal Processing Letters*, **4**(2):52–53, 1997.

[Yer02] A. Yeredor. Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation. *IEEE Transactions on Signal Processing*, **50**(7):1545–1553, 2002.

[ZCW12] C.-Z. Zhan, Y.-L. Chen, and A.-Y. Wu. Iterative superlinear-convergence svd beamforming algorithm and vlsi architecture for mimo-ofdm systems. *IEEE Transactions on Signal Processing*, **60**(6):3264–3277, 2012.

[Zou70] G. Zoutendijk. Nonlinear programming, computational methods. *Integer and Nonlinear Programming*, **143**(1):37–86, 1970.

# List of author's papers related to this thesis

1. Hiroyuki Sato and Toshihiro Iwai,
   A Riemannian optimization approach to the matrix singular value decomposition,
   *SIAM Journal on Optimization* **23**(1), 188–212, 2013.

2. Hiroyuki Sato and Toshihiro Iwai,
   A new, globally convergent Riemannian conjugate gradient method,
   *Optimization*, to appear.

3. Hiroyuki Sato and Toshihiro Iwai,
   A complex singular value decomposition algorithm based on the Riemannian Newton method,
   *Proceedings of the 52nd IEEE Conference on Decision and Control*, to appear.

- Chapters 3, 4, and 5 in this thesis are based on the papers 2, 1, and 3, respectively.
- Subsection 4.4.2 of the thesis is an enlargement of Subsection 4.2 of the second paper, which includes numerical experiments with several Riemannian conjugate gradient methods in addition to the scaled Fletcher-Reeves type method proposed in Chapter 3.