

# OPTIMIZATION APPROACHES ON SMOOTH MANIFOLDS

**Huibo Ji**

October 2007

*The Australian National University*  
*A thesis submitted for the degree of Doctor of Philosophy of*



Department of Information Engineering  
Research School of Information Sciences and Engineering  
The Australian National University



*To Guiyuan Ji and Cuilan Zhang, my parents  
and Chenxu Li, my wife.*

# Acknowledgements

First of all, I would like to express my deep gratitude to my supervisor, Professor Jonathan Manton for being supportive and motivating throughout this research. His rigorous mathematics knowledge and deep insights help me understand my topic well and complete my research smoothly. His serious attitude towards scientific research greatly influences me on research and life. Most of work in this thesis is attributed to the lively meetings and intense discussions with Jonathan.

I am greatly indebted to Professor John Moore for his valuable suggestions and kind support to my research. I would also like to thank my previous supervisor Danchi Jiang, who helped me build up the basic framework of this research.

I would like to express my special thanks to Dr. Minyi Huang and Dr. Christian Lageman for their advice and support for my PhD research.

I am grateful to Dr. Jochen Trumpf and Dr. Robert Orsi for their instructions on mathematics. Moreover, I would like to thank Dr. Vinayaka Pandit for his supports during my internship in the IBM India Research Lab.

I would like to thank the NICTA SEACS program leader and my advisor, Dr. Knut Hüper for his support of my research. I also thank the administrative and IT staff in both the Department of Information Engineering and NICTA for various support, in particular, Debbie Pioch, Rosemary Shepherd, Rita Murray, Julie Arnold, Lesley Goldberg and Peter Shevchenko.

I am grateful to my fellow students, in particular, Huan Zhang, Kaiyang Yang, Pei Yean Lee, Hendra Nurdin, Arvin Dehghani, Wynita Griggs, Andrew Danker, Chi Li, Hao Shen and Yueshi Shen. Their help and friendship enrich my PhD experiences and broaden my horizons.

I would like to acknowledge the financial support from the Australian National University and National ICT Australia Limited (NICTA) for provision of scholarships.

Last, but definitely not least, I offer my special thanks to my family, especially my wife Chenxu Li for all her love, encouragement and understanding.

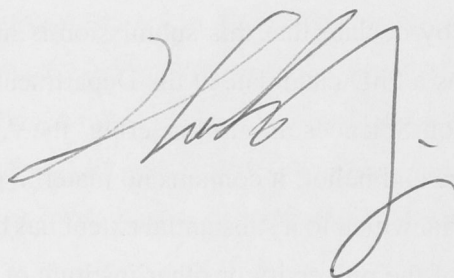
# Statement of Originality

I hereby declare that this submission is my own work, in collaboration with others, while enrolled as a PhD candidate at the Department of Information Engineering, Research School of Information Sciences and Engineering, the Australian National University. To the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Most of the technical discussions in this thesis are based on the following papers published or in preparation:

- Huibo Ji and Danchi Jiang. “A Dynamical System Approach to Semi-definite Programming”. *Proceedings of 2006 CACS Automatic Control Conference*, Taiwan, pp. 105-110, Nov. 2006.
- Huibo Ji, Minyi Huang, John B. Moore and Jonathan H. Manton. “A Globally Convergent Conjugate Gradient Method for Minimizing Self-Concordant Functions with Application to Constrained Optimisation Problems”. *Proceedings of the 2007 American Control Conference*, New York, pp. 540-545, July 2007.
- Danchi Jiang, John B. Moore and Huibo Ji. “Self-Concordant Functions for Optimization on Smooth Manifolds”. *Journal of Global Optimization*, Vol. 38, No. 3, pp. 437-457, July 2007.
- Huibo Ji, Jonathan H. Manton and John B. Moore. “A Globally Convergent Conjugate Gradient Method for Minimizing Self-Concordant Functions on Riemannian Manifolds”. *Proceedings of the 17th World Congress, IFAC*, Seoul, Korea, pp. 14313-14318, July 6-11, 2008.

- Huibo Ji, Jonathan H. Manton and John B. Moore. “A Globally Convergent Conjugate Gradient Method for Minimizing Self-Concordant Functions on Riemannian Manifolds ”. *In preparation*
- Huibo Ji, Jonathan H. Manton and John B. Moore. “An Interior Point Method Based on Conjugate Gradient Method ”. *In preparation*
- Huibo Ji and Jonathan H. Manton. “An Novel Quasi-Newton Method For Optimization on Smooth Manifolds ”. *In preparation*



Huibo Ji

October 2007

Department of Information Engineering  
Research School of Information Sciences and Engineering  
The Australian National University

# Abstract

In recent years, optimization on manifolds has drawn more attention since it can reduce the dimension of optimization problems compared against solving the problems in their ambient Euclidean space. Many traditional optimization methods such as the steepest decent method, conjugate gradient method and Newton method have been extended to Riemannian manifolds or smooth manifolds. In Euclidean space, there exist a special class of convex functions, self-concordant functions introduced by Nesterov and Nemirovskii. They are used in interior point methods, where they play an important role in solving certain constrained optimization problems. Thus, to define self-concordant functions on manifolds will provide the guidance for developing corresponding interior point methods. The aims of this thesis are to

- fully explore properties of the self-concordant function in Euclidean space and develop gradient-based algorithms for optimization of such function;
- define the self-concordant function on Riemannian manifolds, explore its properties and devise corresponding optimization algorithms;
- generalize a quasi-Newton method on smooth manifolds without the Riemannian structure.

Firstly, in Euclidean space, we present a damped gradient method and a damped conjugate gradient method for minimizing self-concordant functions. These two methods are ordinary gradient-based methods but with step-size selection rules chosen to guarantee convergence. As a result, we build up an interior point method based on our proposed damped conjugate gradient method. This method is shown to have lower computational complexity than the Newton-based interior point method.

Secondly, we define the concept of self-concordant functions on Riemannian manifolds and develop the corresponding damped Newton and conjugate gradient methods to minimize such functions on Riemannian manifolds. These methods are proved to guarantee the convergence.

Thirdly, we propose a numerical quasi-Newton method for the optimization on smooth manifolds. This method only requires the local parametrization of smooth manifolds without the need of the Riemannian structure. This method is also shown to have super-linear convergence.

Numerical results show the effectiveness of our proposed algorithms.



# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Overview . . . . .	1
1.2	Literature Review . . . . .	2
1.2.1	Interior Point Method and Self-concordant Functions . . . . .	2
1.2.2	Optimization On Smooth Manifolds . . . . .	3
1.3	Motivation and Research Aims . . . . .	6
1.3.1	Motivation . . . . .	6
1.3.2	Research Aims . . . . .	6
1.4	Outline of This Thesis . . . . .	7
1.4.1	Self-concordant Functions in Euclidean Space . . . . .	7
1.4.2	Self-concordant Functions On Riemannian Manifolds . . . . .	7
1.4.3	The Quasi-Newton Methods . . . . .	8
<b>I</b>	<b>Self-concordant Functions in Euclidean Space</b>	<b>9</b>
<b>2</b>	<b>Introduction to Self-Concordant Functions in Euclidean Space</b>	<b>10</b>
2.1	Introduction . . . . .	10
2.2	Definition and Properties . . . . .	10
2.3	Damped Newton Algorithm . . . . .	13
<b>3</b>	<b>Damped Gradient and Conjugate Gradient Methods</b>	<b>15</b>
3.1	Introduction . . . . .	15
3.2	Review of The Gradient and Conjugate Gradient Methods . . . . .	17
3.3	The Damped Method . . . . .	19
3.3.1	The Damped Gradient Method . . . . .	19
3.3.2	The Damped Conjugate Gradient Method . . . . .	23
3.4	Interior Point Method . . . . .	30

3.5	Numerical Examples . . . . .	33
3.5.1	Example One . . . . .	33
3.5.2	Example Two . . . . .	39
3.6	Conclusion . . . . .	43
<b>II</b>	<b>Self-Concordant Functions On Riemannian Manifolds</b>	<b>44</b>
<b>4</b>	<b>Self-concordant Functions on Riemannian Manifolds</b>	<b>45</b>
4.1	Introduction . . . . .	45
4.2	Concepts of Riemannian Manifolds . . . . .	46
4.3	Self-Concordant Functions . . . . .	50
4.4	Newton Decrement . . . . .	60
4.5	A Damped Newton Algorithm for Self-Concordant Functions . . . . .	61
4.6	Conclusion . . . . .	68
<b>5</b>	<b>Damped Conjugate Gradient Methods on Riemannian Manifolds</b>	<b>70</b>
5.1	Introduction . . . . .	70
5.2	Conjugate Gradient Method On Riemannian manifolds . . . . .	71
5.3	Damped Conjugate Gradient Method . . . . .	73
5.4	Conclusion . . . . .	82
<b>6</b>	<b>Application of Damped Methods on Riemannian Manifolds</b>	<b>83</b>
6.1	Example One . . . . .	84
6.2	Example Two . . . . .	85
6.3	Example Three . . . . .	89
<b>III</b>	<b>The Quasi-Newton Method</b>	<b>104</b>
<b>7</b>	<b>A Quasi-Newton Method On Smooth Manifolds</b>	<b>105</b>
7.1	Introduction . . . . .	105
7.2	Preliminaries . . . . .	106
7.3	Quasi-Newton Method On Smooth Manifolds . . . . .	106
7.4	Numerical Example . . . . .	115
7.5	Conclusions . . . . .	117
	<b>Bibliography</b>	<b>119</b>

# List of Figures

3.1	Conjugate gradient in Euclidean space . . . . .	18
3.2	Error vs. iteration number with good initial guess for QCQOP from the back-tracking gradient method, damped Newton method, line search gradient method and line search conjugate gradient method . . . . .	37
3.3	Error vs. iteration number with good initial guess for QCQOP from the back-tracking gradient method, damped Newton method, damped gradient method and damped conjugate gradient method . . . . .	37
3.4	Error vs. iteration number with bad initial guess for QCQOP from the back-tracking gradient method, damped Newton method, damped gradient method and damped conjugate gradient method . . . . .	38
3.5	The value of the original cost function with $\mu$ approaching 0 for OCQOP . . . . .	38
3.6	Error vs. iteration number for SOCP from the damped Newton method, damped gradient method and damped conjugate gradient method . . . . .	42
3.7	The value of the original cost function with $\mu$ approaching 0 for SOCP . . . . .	42
4.1	The tangent and normal spaces . . . . .	47
4.2	Move a tangent vector parallel to itself to another point on the manifold . . . . .	48
4.3	Parallel transport (infinitesimal space) . . . . .	48
5.1	Conjugate gradient direction on Riemannian Manifolds . . . . .	72
6.1	The result of damped Newton method for the self-concordant function defined on the circle . . . . .	87
6.2	The result of damped Newton method for the self-concordant function defined on high-dimension sphere . . . . .	88
6.3	The result of damped conjugate gradient method for the self-concordant function defined on high-dimension sphere . . . . .	89
6.4	The result of the the damped Newton method for the self-concordant function defined on the Hyperboloid model . . . . .	102

6.5	The result of the damped conjugate gradient method for the self-concordant function defined on the Hyperboloid model . . . . .	103
7.1	The result of Quasi-Newton Method compared against the Newton method and steepest decent method . . . . .	118

# List of Notation

$\mathbb{R}$	The real numbers.
$\mathbb{R}^n$	The set of all real $n \times 1$ vectors.
$\mathbb{R}^{n \times p}$	The set of all real $n \times p$ matrices.
$A^T$	The transpose of a matrix $A$ .
$I_n$	The $n \times n$ identity matrix.
$\det(A)$	The determinant of matrix $A$ .
$\text{diag}(x)$	The $n \times n$ diagonal matrix for $x \in \mathbb{R}^n$ whose $i^{\text{th}}$ diagonal term is $x_i$ .
$\text{vec}(A)$	The column vector consisting of the stacked columns of the matrix $A$ .
$\text{tr}(A)$	The sum of the diagonal elements of a matrix $A$ .
$\ x\ $	The Euclidean norm of a vector $x \in \mathbb{R}^n$ , $\ x\  = \sqrt{x^T x}$ .
$\ A\ _F$	The Frobenius Norm of a matrix $A \in \mathbb{R}^{n \times p}$ , $\ A\ _F = \sqrt{\text{tr}(A^T A)}$ .
$T_p M$	The tangent space of a manifold $M$ at $p \in M$ .
$\langle X, Y \rangle_p$	The inner product defined on $X, Y \in T_p M$ , where $p \in M$ and $M$ is a Riemannian manifold.
$\ X\ _p$	The norm of a tangent vector $X \in T_p M$ , $\ X\ _p = \sqrt{\langle X, X \rangle_p}$ , where $p \in M$ and $M$ is a Riemannian manifold.
$\text{Exp}_p tX$	The geodesic emanating from $p \in M$ in the direction $X \in T_p M$ where $M$ is a Riemannian manifold.
$\tau_{pq}$	The parallel transport from $T_p M$ to $T_q M$ along the given geodesic connecting $p$ and $q$ where $p, q \in M$ and $M$ is a Riemannian manifold.
$S^n$	The $n$ -dimensional unit sphere in $\mathbb{R}^{n+1}$ , $S^n = \{x \in \mathbb{R}^{n+1}   x^T x = 1\}$ .
$I^n$	The $n$ -dimensional hyperboloid model in $\mathbb{R}^{n+1}$ , $I^n = \{x \in \mathbb{R}^{n+1}   -\sum_{i=1}^n x_i^2 + x_{n+1}^2 = 1, x_{n+1} > 0\}$ .
$St(n, p)$	The Stiefel manifold defined by the set of real $n \times p$ orthogonal matrices, $St(n, p) := \{X \in \mathbb{R}^{n \times p}   X^T X = I_p\}$ .
$Gr(n, p)$	The Grassmann manifold defined by the set of all $p$ -dimensional spaces of $\mathbb{R}^n$ .

# Chapter 1

## Introduction

### 1.1 Overview

Optimization plays an important role in both research and applications. The essence of optimization problems is to search the minimum or maximum of cost functions. Methods to solve optimization problems have been widely studied. For example, given a cost function  $f$  defined on the whole  $\mathbb{R}^n$ , one can use conventional methods such as the steepest descent method, conjugate gradient method or Newton method to minimize this function. However, in engineering, many cost functions are subject to constraints. For example, see [53, 72]. Minimizing functions subject to inequality constraints have resulted in new optimization methods. If a constraint is linear or nonlinear convex, the interior point method is commonly used. Assuming that we have an optimization problem defined on a linear or nonlinear convex constraint, the idea of the interior point method is to transform the constrained optimization problem into a parameterized unconstrained one using a barrier penalty function, commonly constructed by a self-concordant function defined by Nesterov and Nemirovskii [58]. The barrier function remains relatively flat in the interior of the constraint while tending to infinity when approaching the boundary. A new cost function including the original cost function and the barrier function is constructed. Then we can use plain methods to minimize the new cost function until we find the optimal value of the original problem.

In recent years, optimization on smooth manifolds has drawn more attention since it can reduce the dimension of optimization problems compared to solving the original problem in their ambient Euclidean space. Its applications appear in medicine [3], signal processing [53], machine learning [60], computer vision [50, 32], and robotics [35, 33]. Optimization approaches

on smooth manifolds can be categorized into Riemannian approaches and non-Riemannian approaches.

1. Methods of solving minimization problems on Riemannian manifolds have been extensively researched. For more details, see [70, 67, 18, 19, 71]. In fact, traditional optimization methods such as the steepest gradient method, conjugate gradient method and Newton method in Euclidean space can be applied to optimization on Riemannian manifolds with slight changes. A typical intrinsic approach for minimization is based on the computation of geodesics and covariant differentials, which may be resource expensive. However, there are many meaningful cases where the computation can be very simple. An example is the hyperboloid space, where the geodesic and parallel transformation can be computed via hyperbolic functions and vector calculations. Another simple but non-trivial case is the sphere, where the geodesic and parallel transformation can be computed via trigonometrical functions and vector calculation. As a consequence, it is natural to ask: can we define self-concordant functions on Riemannian manifolds and what is the related interior point method? Clearly solving such a question will have practical importance and theoretical completeness.
2. As mentioned above, the computational cost of computing geodesics is often relatively high. For this and other reasons, Manton [53] developed a more general framework for optimization on manifolds. This framework does not require a Riemannian structure to be defined on the manifold and its greater generality allows more efficient algorithms to be developed.

In the rest of this chapter, we first review the developments in the interior point methods, self-concordant functions and the optimization on smooth manifolds. Then the motivation and research aims are given. Finally, the outline of this thesis is presented.

## 1.2 Literature Review

### 1.2.1 Interior Point Method and Self-concordant Functions

The basic idea of interior-point methods is as follows. If  $f(x)$  is a convex cost function we wish to minimize on a convex set  $Q$  of  $\mathbb{R}^n$ , and if  $g(x)$  is a barrier function meaning that  $g(x)$

approaches infinity on the boundary of  $Q$ , then we solve the sequence of optimization problems  $x_k = \arg \min_x \frac{1}{\mu_k} f(x) + g(x)$  where  $\mu_k \rightarrow 0$  and  $\mu_k > 0$ . As  $\mu_k$  converges to zero,  $x_k$  will converge to the minimal point of the original cost function  $f(x)$  constrained to  $Q$ , under the assumption that  $f(x)$  and  $g(x)$  are convex.

The history of the interior point method can be traced back to Khachiyan's work [45] which first introduced the polynomial-time interior point method in 1979. However, the start of the interior-point revolution was Karmarkar's claim in 1984 of a polynomial-time linear programming method [43]. Then the equivalence between Karmarkar's method and the classical logarithmic barrier method was shown in 1986 [26].

The milestone work of Nesterov and Nemirovskii [58] presented a new special class of barrier methods and developed polynomial-time complexity results for new convex optimization problems. Their proposed self-concordant functions are critically important in powerful interior point polynomial algorithms for convex programming in Euclidean space. The significance of these functions lies in two aspects. Firstly, they provide many of logarithmic barrier functions which are important in interior point methods for solving convex optimization problems. Secondly, the proposed damped Newton method for optimizing self-concordant functions avoids the knowledge of some parameters (e.g., Lipschitz constants, etc.). This is useful for constrained optimization problems. It is also worth noting that using self-concordant barrier functions guarantees the original problem to be solved in a polynomial time for a pre-defined precision.

## 1.2.2 Optimization On Smooth Manifolds

As stated before, traditional optimization techniques in Euclidean space can have their counterparts on smooth manifolds, which have been studied. In this section, we first review the classical Riemannian approaches and then the relatively recent non-Riemannian approaches.

### Riemannian Approach

#### 1. Steepest descent method on manifolds

The steepest descent method is the simplest method for the optimization on Riemannian manifolds and it has good convergence properties but slow linear convergence rate. This method was first introduced to manifolds by Luenberger [48, 49] and Gabay [25]. In the early nineties, this method was carried out to problems in systems and control theory by Brockett [13], Helmke and Moore [34], Smith [66] and Mahony [51].



## 2. Newton method on manifolds

Compared against the steepest descent method, the Newton method has a faster (quadratic) local convergence rate. In 1982, Gabay extended the Newton method to a Riemannian sub-manifold of  $\mathbb{R}^n$  by updating iterations along a geodesic. Other independent work has been developed to extend the Newton method on Riemannian manifolds by Smith [67] and Mahony [51, 52] restricting to the compact Lie group, and by Udriste [70] restricting to convex optimization problems on Riemannian manifolds. Edelman, Arias and Smith [19] also introduced a Newton method for the optimization on orthogonality constraints - the Stiefel and Grassmann manifolds. There is also a recent paper by Dedieu, Priouret and Malajovich [18] which studied the Newton method to find zero of a vector field on general Riemannian manifolds.

## 3. Quasi-Newton method on manifolds

Even though the Newton's method has faster quadratic convergence rate, it requires solving a linear system per iteration which consists of the second order local information of the cost function. Therefore, it increases the computational cost. In order to avoid this problem, the quasi-Newton method in Euclidean space was presented by Davidon [17] in late 1950s. This method uses only the first order information of the cost function to approximate the Hessian inverse and has a super-linear local convergence rate. Since then, various quasi-Newton methods have been introduced. However, among them, the most popular methods are the Davidon-Fletcher-Powell (DFP) [22] method and the Broyden [15, 16] Fletcher [21] Goldfarb [28] Shanno [65] (BFGS) method.

In the early eighties, Gabay [65] firstly generalized the BFGS method to a Riemannian manifold. However, he did not give the complete proof of the convergence of his method. Recently, Brace and Manton [12] developed an improved BFGS method on the Grassmann manifold and achieved a lower computational complexity compared to Gabay's method.

## 4. Conjugate gradient method on manifolds

While considering the large scale optimization problems with sparse Hessian matrices, the quasi-Newton methods encounters difficulties. Due to avoiding computing the inverse of the Hessian, the conjugate gradient method can be used for solving such problems. This method was originally developed by Hestenes and Stiefel [38] in the 1950s to solve large scale systems of linear equations. Then in the mid 1960s, Fletcher and Reeves [24]

popularized this method to solve unconstrained optimization problems. In 1994, Smith [67] extended this method to Riemannian manifolds and later Edelman, Arias and Smith [19] applied his method specifically on the Stiefel and Grassmann manifolds.

### Non-Riemannian Approach

The traditional methods for optimizing a cost function on a manifold all start by endowing the manifold with a metric structure, thus making it into a Riemannian manifold. The reason for doing so is that it allows Euclidean algorithms, such as steepest descent and Newton methods, to be generalised reasonably straightforwardly; the gradient is replaced by the Riemannian gradient, for example. However, as pointed out in [54], the introduction of a metric structure is extraneous to the underlying optimisation problem and thus, in general, is detrimental. (A possible exception is when the cost function is somehow related to the Riemannian geometry, such as if it is defined in terms of the distance function on the Riemannian manifold.)

For an arbitrary smooth manifold, the only structure we know is that around any point, the manifold looks like  $\mathbb{R}^n$ . This is explained as follows. Let  $M$  be a smooth  $n$ -dimensional manifold. For every point  $p \in M$ , there exists a smooth map

$$\psi_p : \mathbb{R}^n \rightarrow M, \psi_p(0) = p \quad (1.1)$$

which is a local diffeomorphism around  $0 \in \mathbb{R}^n$ . Such a  $\psi_p$  is called a local parametrization.

In [53], Manton gave a general framework for developing numerical algorithms for minimizing a cost function defined on a manifold. The framework entailed choosing a particular local parametrization about each point on the manifold. In the same paper, this framework was applied to the Stiefel and Grassmann manifolds, and as an example, the local parametrizations were chosen, in a certain sense, to be projections from the tangent space to the manifold itself. Based on this parametrization, the corresponding steepest descent and Newton methods were shown to reduce the computational complexity compared with the traditional Riemannian methods.

## 1.3 Motivation and Research Aims

### 1.3.1 Motivation

Since from the optimization point of view self-concordant functions enjoy good tractability, it is tempting to extend its definition to manifolds and develop corresponding optimization algorithms. In [57], Nesterov only provided a Newton-based algorithm for optimization of a self-concordant function in Euclidean space. Although this algorithm has quadratic convergence, the requirement of solving a linear system per iteration creates significant computational complexity per iteration. To avoid this problem, some gradient-based methods such as gradient and conjugate gradient methods can be taken first before switching to Newton-based methods. Due to the importance of gradient-based method, we are motivated to develop a damped gradient method and a damped conjugate gradient method for optimization of self-concordant functions. Furthermore, it is expected that these two methods can be applied to the optimization of self-concordant functions on Riemannian manifolds. Consequently, they can provide guidance to develop interior point methods on Riemannian manifolds.

In addition, although the steepest and Newton methods have been developed as the non-Riemannian approach, to our best knowledge, we are not aware of any published papers introducing the non-Riemannian based quasi-Newton methods on smooth manifolds. Since the quasi-Newton method has prominent advantages, we are also motivated to develop a quasi-Newton method on smooth manifolds.

### 1.3.2 Research Aims

The aims of this thesis are to

- fully explore properties of the self-concordant function in Euclidean space and develop gradient-based algorithms for optimization of such function;
- define the self-concordant function on Riemannian manifolds, explore its properties and devise corresponding optimization algorithms;
- generalize a quasi-Newton method on smooth manifolds without the Riemannian structure.

## 1.4 Outline of This Thesis

To achieve our research aims, besides this introduction chapter, this thesis consists of three parts.

### 1.4.1 Self-concordant Functions in Euclidean Space

Part I includes three chapters and mainly focuses on the properties of self-concordant functions in Euclidean space and algorithms for the optimization of such functions.

In Chapter 2, we review the definition of self-concordant functions in Euclidean space, introduced by Nesterov and Nemirovskii in [58] and their properties. We also recall the damped Newton algorithm from [58] for the optimization of self-concordant functions and its convergence properties.

In Chapter 3, we propose a damped gradient method and a damped conjugate gradient method for optimization of self-concordant functions. A damped Newton method introduced by Nesterov is an ordinary Newton method but with a step-size selection rule chosen to guarantee convergence. Based on the gradient and conjugate gradient methods, our methods provide novel step-size selection rules which are proved to ensure that algorithms converge to the global minimum. The advantage of our methods over the damped Newton method is that the former have a lower computational complexity. Then, we build up an interior point method based on our proposed damped conjugate gradient method. Finally, our algorithms are applied to second order cone programming and quadratically constrained quadratic optimization problems.

### 1.4.2 Self-concordant Functions On Riemannian Manifolds

Part II includes three chapters and mainly focuses on the properties of self-concordant functions on Riemannian manifolds and algorithms for the optimization of such functions.

In chapter 4, the self-concordant functions are defined on Riemannian manifolds. Then generalizations of the properties of self-concordant functions on Riemannian manifolds are derived. Based on properties, a damped Newton algorithm is proposed for optimization of self-concordant functions, which guarantees that the solution falls in any given small neighborhood of the optimal solution, with its existence and uniqueness also proved in this chapter, in a finite number of steps. It also ensures quadratic convergence within a neighborhood of the minimal point.

In Chapter 5, we present a damped conjugate gradient method for optimization of self-concordant functions defined on smooth Riemannian manifolds. A damped conjugate gradient method is an ordinary conjugate gradient method with an explicit step-size selection rule. It is proved that this method guarantees to converge to the global minimum super-linearly. Compared against the damped Newton method, the damped conjugate gradient method has a lower computational complexity.

In Chapter 6, we introduce three examples in which the cost functions are self-concordant on different Riemannian manifolds. We also applied our damped Newton method and conjugate gradient method into minimizing these three cost functions. Simulation results show the nice performance of our algorithms.

### 1.4.3 The Quasi-Newton Methods

Part III includes one chapter and mainly focuses developing a new quasi-Newton method on smooth manifolds.

In Chapter 7, we propose a new quasi-Newton method on smooth manifolds based on the local parametrization. This method is proved to converge to the minimum of the cost function. To demonstrate its efficiency, we applied this quasi-Newton method into a cost function defined on the Grassmann manifold. The simulation result shows our method has super-linear convergence.

# Part I

## Self-concordant Functions in Euclidean Space

## Chapter 2

# Introduction to Self-Concordant Functions in Euclidean Space

### 2.1 Introduction

In this chapter, we review the self-concordant function in Euclidean space, which is proposed by Nesterov and Nemirovskii in [58]. A damped Newton method was also presented in [58] for optimization of such function. In this chapter, we briefly introduce the damped Newton method and its convergence properties. For more details, refer to [58] and [57].

### 2.2 Definition and Properties

In this section, we recall some concepts and properties related to self-concordant functions defined in Euclidean space.

Throughout this chapter,  $f$  will denote a real-valued function from a convex subset  $Q$  of  $\mathbb{R}^n$ . We consider constrained optimization problems of the following form

$$\min_{x \in Q} f(x) \quad f : Q \subset \mathbb{R}^n \rightarrow \mathbb{R}. \quad (2.1)$$

In general, it is hard to solve (2.1), even numerically. However, if  $f$  has certain nice properties, there exist powerful techniques to solve (2.1). In [57], Nesterov considered the case when  $f$  is self-concordant, defined as follows.

**Definition 1.** Let  $f : Q \rightarrow \mathbb{R}$  be a  $C^3$ -smooth closed convex function defined on an open domain  $Q \subset \mathbb{R}^n$ . Then  $f$  is self-concordant if there exists a constant  $M_f > 0$  such that the inequality

$$|D^3 f(x)[u, u, u]| \leq M_f \{D^2 f(x)[u, u]\}^{3/2} \quad (2.2)$$

holds for any  $x \in Q$  and direction  $u \in \mathbb{R}^n$ .

Recall that  $f$  being  $C^3$ -smooth means  $f$  is three-times continuously differentiable. A convex function  $f : Q \rightarrow \mathbb{R}$  is called closed if its epigraph, defined as  $\text{epi}(f) = \{(x, t) \in Q \times \mathbb{R} \mid t \geq f(x)\}$ , is closed. The reason why  $f$  is required to be closed in Definition 1 is to ensure that  $f$  behaves nicely on the boundary of its domain; see (2.8). Also, the second and third directional derivatives  $D^2$  and  $D^3$  are defined as follows. Given  $x \in Q$  and  $u \in \mathbb{R}^n$ ,

$$D^2 f(x)[u, u] = \left. \frac{d^2}{dt^2} \right|_{t=0} \{f(x + tu)\}, \quad (2.3)$$

$$D^3 f(x)[u, u, u] = \left. \frac{d^3}{dt^3} \right|_{t=0} \{f(x + tu)\}. \quad (2.4)$$

We consider the special case of the optimization problem in (2.1) where  $f$  satisfies the following assumption.

**Assumption 1.** The function  $f$  in (2.1) is self-concordant, has a minimum in  $Q$  and for all  $x \in Q$ ,  $f''(x)$  is nonsingular. By scaling  $f$  if necessary, it is assumed without loss of generality that  $f$  satisfies (2.2) with  $M_f = 2$ .

The need for assuming  $f$  has a minimum in Assumption 1 can be seen from the example  $g(x) = -\ln x$  defined on  $(0, +\infty)$  where  $g$  is self-concordant but  $g$  has no minimum. If  $f$  has a minimum, then it is unique because  $f$  is strictly convex by Assumption 1. Note that by Theorem 4.1.3 in [57],  $f''(x)$  is nonsingular for all  $x \in Q$  if the domain  $Q$  contains no straight line.

Functions satisfying Assumption 1 have interesting properties which facilitate our further analysis. For details, see [57]. For a given  $x \in Q$ , we introduce a local norm on  $\mathbb{R}^n$

$$\|u\|_x = \langle f''(x)u, u \rangle^{1/2}, \quad u \in \mathbb{R}^n \quad (2.5)$$

and the Dikin ellipsoid of unit length

$$W^0(x) = \{y \in \mathbb{R}^n \mid \|y - x\|_x < 1\}. \quad (2.6)$$



Then the following properties hold.

1. For any point  $\bar{x} \in \partial(Q)$  and any sequence

$$\{x_k\} \subset Q : x_k \rightarrow \bar{x} \tag{2.7}$$

we have

$$f(x_k) \rightarrow +\infty \tag{2.8}$$

where  $\partial(Q)$  denotes the boundary of  $Q$ . In other words,  $f$  is a barrier function going to infinity on the boundary of  $Q$ .

2. For any  $x \in Q$ , we have

$$W^0(x) \subset Q. \tag{2.9}$$

3. Let  $x \in Q$ . Then for any  $y \in W^0(x)$  we have

$$(1 - \|y - x\|_x)^2 f''(x) \preceq f''(y) \preceq \frac{1}{(1 - \|y - x\|_x)^2} f''(x) \tag{2.10}$$

where  $A \preceq B$  means the matrix  $B - A$  is positive semidefinite. In other words, a self-concordant function looks approximately quadratic in a small enough region around any point in  $Q$ .

4. Let  $x \in Q$  and  $r = \|y - x\|_x < 1$ . Then we can estimate the matrix

$$M = \int_0^1 f''(x + \tau(y - x)) d\tau \tag{2.11}$$

by

$$(1 - r + \frac{r^2}{3}) f''(x) \preceq M \preceq \frac{1}{1 - r} f''(x). \tag{2.12}$$

5. For any  $x, y \in Q$  we have

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \omega(\|y - x\|_x) \quad (2.13)$$

where  $\omega(t) = t - \ln(1 + t)$ . Note that if  $y \neq x$ , then  $\omega(\|y - x\|_x)$  is positive and (2.13) gives a useful lower bound on  $f(y)$ .

6. Let  $x \in Q$  and  $\|y - x\|_x < 1$ . Then

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \omega_*(\|y - x\|_x) \quad (2.14)$$

where  $\omega_*(t) = -t - \ln(1 - t)$ . Similar to above, if  $y \neq x$ , then  $\omega_*(\|y - x\|_x)$  is also positive and (2.14) gives a useful upper bound on  $f(y)$ .

**Lemma 1.** *Let  $f : Q \rightarrow \mathbb{R}$  be a  $C^3$ -smooth closed convex function defined on an open domain  $Q \subset \mathbb{R}^n$ . Then  $f$  is self-concordant if and only if there exists a constant  $M_f > 0$  such that for all  $x \in Q$  and any directions  $u_1, u_2, u_3 \in R^n$ , we have*

$$|D^3 f(x)[u_1, u_2, u_3]| \leq M_f \prod_{i=1}^3 \|u_i\|_x.$$

Lemma 1 gives an alternative definition of a self-concordant function.

## 2.3 Damped Newton Algorithm

In this section, we review the damped Newton algorithm [57] for optimization of self-concordant functions. This algorithm is a Newton-based method but with an explicit step-size rule.

To illustrate the algorithm, the Newton decrement  $\lambda(x)$  is introduced, defined in terms of the gradient and the Hessian as follows

$$\lambda(x) = \langle [f''(x)]^{-1} \nabla f(x), \nabla f(x) \rangle^{\frac{1}{2}}. \quad (2.15)$$

The Newton decrement  $\lambda(x)$  plays an important role in the optimization of self-concordant functions based on the Newton method.

Nesterov constructed a damped Newton method as follows to minimize  $f$  in (2.1) when  $f$  satisfies Assumption 1.

**Algorithm 1. (Damped Newton Algorithm)** [57]

step 0: Select an initial point  $x_0 \in Q$  and set  $k=0$ ;

step k: Set

$$\lambda(x_k) = \langle [f''(x_k)]^{-1} \nabla f(x_k), \nabla f(x_k) \rangle^{\frac{1}{2}}$$

$$x_{k+1} = x_k - \frac{1}{1 + \lambda(x_k)} [f''(x_k)]^{-1} \nabla f(x_k).$$

Increment  $k$  and repeat until convergence.

It is worth noting that Algorithm 1 guarantees in every step that  $x_{k+1}$  lies in the Dikin ellipsoid around  $x_k$ . In other words, the sequence  $\{x_k\}$  given by Algorithm 1 lies in the domain  $Q$ . The following proposition shows that the damped Newton method decreases the value of  $f(x)$  significantly.

**Proposition 2.3.1.** [57] Let  $\{x_k\}$  be a sequence generated by Algorithm 1, where the cost function  $f : Q \rightarrow R$  in 2.1 satisfies Assumption 1. Then,  $\forall k$ , we have

$$f(x_{k+1}) \leq f(x_k) - \omega(\lambda(x_k)) \quad (2.16)$$

where  $\omega(t) = t - \ln(1 + t)$ .

The following proposition illustrates that the local convergence of Algorithm 1 is quadratic.

**Proposition 2.3.2.** [57] Let  $\{x_k\}$  be a sequence generated by Algorithm 1, where the cost function  $f : Q \rightarrow R$  in (2.1) satisfies Assumption 1. Then  $\forall k$

$$\lambda(x_{k+1}) \leq 2\lambda^2(x_k). \quad (2.17)$$

# Chapter 3

## Damped Gradient and Conjugate Gradient Methods

### 3.1 Introduction

**Background** In [57], Nesterov developed a damped Newton method to compute the minimum of a self-concordant function. The key feature of this method is that it provides an explicit step-size choice based on the Newton method and the value of the function strictly decreases in each iteration of this method. In addition, this method guarantees that the iteration sequence remains in the domain of the cost function. It was proved that this method always converges to the minimum of the self-concordant function. On the other hand, since the damped Newton method is Newton-based, it possesses certain inherent disadvantages of the Newton method. One of them is the expensive computational cost involved in solving a linear system per iteration which consists of the Hessian matrix of the cost function. Therefore, we are motivated to build up gradient-based methods with explicit step-size rules for the optimization of self-concordant functions.

Concerning finding an appropriate step-size, several rules have been developed for gradient-based methods. For instance, the gradient-based step-size can be determined via various line search methods for constrained optimization problems. However, the resulting disadvantage of these methods is that they may increase the cost of additional iterations for a suitable step-size. In [68], Sun and Zhang presented an explicit formula for step-size selection to find the minimum of an unconstrained function based on the conjugate gradient direction. This method is shown to ensure convergence to the local minimum. However, it does not generalize immediately to

the constrained case since it can not guarantee that the iterations remain inside the constrained domain.

**Our work** In this chapter, we present a damped gradient method and a damped conjugate gradient method to minimize self-concordant functions on Euclidean space. Our methods are shown to converge to the optimal solution of a self-concordant function if it exists. One of the advantages of our methods is that they only consist of computing the gradient and Hessian matrix of the cost function without solving a linear system in the Newton method. In every step, the complexity of these two methods is  $O(n^2)$  instead of  $O(n^3)$  in the damped Newton method, where  $n$  is the dimension of the variable.

**Chapter outline** The rest of this chapter is organized as follows. We review the traditional gradient and conjugate gradient methods in Section 3.2. In Section 3.3, the damped gradient and conjugate gradient methods are derived and it is shown that these two algorithms converge to the minimum of the cost function provided the cost function is self-concordant. In the last section, two examples are included to illustrate the convergence properties of these algorithms.

Notation: The symbol  $S^n$  denotes the set of  $n$  dimensional real symmetric matrices. An inner product on  $S^n$  is:

$$\langle X, Y \rangle_F = \text{trace}(XY). \quad (3.1)$$

It induces the Frobenius norm  $\|X\|_F = \langle X, X \rangle_F^{\frac{1}{2}}$ . The notation  $0 \preceq X$  means that  $X$  is positive semidefinite. For any  $X, Y \in S^n$ ,  $Y \preceq X$  means that  $X - Y$  is positive semidefinite. Throughout, the Euclidean inner product on  $\mathbb{R}^n$  is used, namely  $\langle x, y \rangle = x^T y$ . It induces the Euclidean norm  $\|x\| = \langle x, x \rangle^{\frac{1}{2}}$ . The symbol  $\partial(Q)$  denotes the boundary of the set  $Q$ . Throughout,  $f$  will denote a real-valued function defined on a convex subset  $Q$  of  $\mathbb{R}^n$ . We consider constrained optimization problems of the form

$$\min_{x \in Q} f(x), \quad f : Q \subset \mathbb{R}^n \rightarrow \mathbb{R}. \quad (3.2)$$

## 3.2 Review of The Gradient and Conjugate Gradient Methods

In this section, we review descent methods such as the gradient method and the conjugate gradient method for solving the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \quad f: \mathbb{R}^n \rightarrow \mathbb{R}. \quad (3.3)$$

Motivation for introducing our novel damped methods in Section 4 for optimization of self-concordant functions is also given.

Descent algorithms for solving (3.3), such as the gradient and conjugate gradient methods, are of the following form. Initially, a point  $x_0 \in \mathbb{R}^n$  is chosen at random. Then the sequence  $\{x_k\}$  is generated according to the rule  $x_{k+1} = x_k + h_k H_k$  where  $H_k$  is called the descent direction and  $h_k$  the step-size. There are several step-size rules commonly used:

1. The sequence  $\{h_k\}_{k=0}^{\infty}$  is chosen in advance [57]. Two examples are a constant step-size  $h_k = h > 0$ , or  $h_k = \frac{h}{\sqrt{k+1}}$ ,  $h > 0$ .
2. Line search [57]:  $h_k = \arg \min_{h \geq 0} f(x_k + hH_k)$ .
3. Backtracking [72]: Start with unit step-size  $h_k = 1$ , then reduce it by the multiplicative factor  $\beta$  until the stopping condition  $f(x_k + h_k H_k) \leq f(x_k) + \alpha h_k \langle f'(x_k), H_k \rangle$  holds. The parameter  $\alpha$  is typically chosen between 0.01 and 0.3, and  $\beta$  between 0.1 and 0.8.

Although these step-sizes can work well in practice, they each have their limitations. The first rule cannot always guarantee that the descent algorithm converges to the solution of (3.3). Indeed, the values  $f(x_k)$  need not even form a decreasing sequence. In general the second rule is only acceptable in theory because the line search is often too hard to compute in practice. The choice of  $\alpha$  and  $\beta$  in the third rule is somewhat arbitrary.

Different algorithms use different rules for determining the descent direction  $H_k$ , as now reviewed.

The gradient method chooses  $H_k$  to be  $-f'(x_k)$ . Since it requires only first order information, the gradient method is relatively cheap to implement. Often, a few steps of the gradient method are taken before switching to a higher order method.

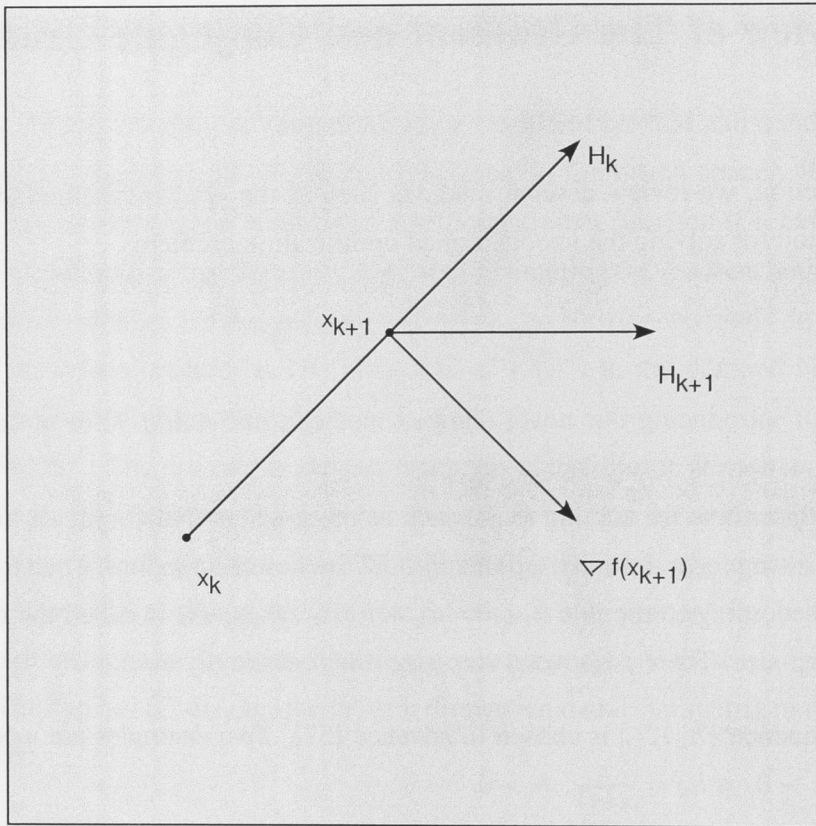


Figure 3.1: Conjugate gradient in Euclidean space

The conjugate gradient method is a popular method because it is easy to implement, has low storage requirements, and provides superlinear convergence in the limit. The primary idea of the conjugate gradient method is to use the conjugacy to find the search direction.

**Algorithm 2. (Conjugate gradient algorithm)**

step 0: Select an initial point  $x_0$ , compute  $H_0 = -f'(x_0)$ , and set  $k = 0$ .

step k: If  $f'(x_k) = 0$  then terminate. Otherwise, compute  $h_k$  with the exact line search method.

Set  $x_{k+1} = x_k + h_k H_k$ .

Set

$$\gamma_{k+1} = -\frac{\|f'(x_{k+1})\|^2}{\langle f'(x_k), H_k \rangle}, \tag{3.4}$$

$$H_{k+1} = -f'(x_{k+1}) + \gamma_{k+1} H_k. \tag{3.5}$$

If  $k + 1 \bmod n \equiv 0$ , set  $H_{k+1} = -f'(x_{k+1})$ . Increment  $k$  and repeat until convergence.

Figure 3.1 illustrates the conjugate gradient direction in Euclidean space. Whereas  $H_{k+1}$  in the gradient method depends only on  $f'(x_{k+1})$ , in the conjugate gradient method,  $H_{k+1}$  relies also on the past history of  $H_k$  via a weighting factor  $\gamma_{k+1}$ . In (3.4),  $\gamma_{k+1}$  is given by the conjugate descent method in [23]; other choices are possible.

Consider applying a gradient or conjugate gradient method to the constrained optimization problem (3.2). None of the three step-size rules presented earlier are directly suitable; the first and third rules cannot ensure  $x_k$  remains in  $Q$ , while the second rule is difficult to implement in practice. This motivates the introduction of novel step-size rules which guarantee the convergence of the algorithm to the minimum of the cost function.

### 3.3 The Damped Method

In this section, we derive explicit step-size rules for use in gradient and conjugate gradient methods for solving (3.2) when  $f$  is self-concordant. These step-size rules guarantee convergence to the global minimum.

#### 3.3.1 The Damped Gradient Method

Let  $f$  in (3.2) satisfy Assumption 1. Suppose we have a point  $x_k$  in  $Q$  at time  $k$ . Given an appropriate step-size  $h_k$ , the gradient method sets  $x_{k+1} = x_k - h_k f'(x_k)$ . We propose choosing  $h_k$  to maximize the bound in (2.14). Later, in Theorem 3.3.1, it is proved such a strategy guarantees convergence to the minimum of  $f$ .

From (2.14), provided  $x_{k+1} \in W(x_k)$  and  $h_k \geq 0$ , we have

$$f(x_k) - f(x_{k+1}) \geq h_k \|f'(x_k)\|^2 + h_k \|f'(x_k)\|_{x_k} + \ln(1 - h_k \|f'(x_k)\|_{x_k}). \quad (3.6)$$

The right hand side is of the form  $\psi(h_k)$  where  $\psi(h) = \alpha h + \ln(1 - \beta h)$  with  $\alpha = \|f'(x_k)\|^2 + \|f'(x_k)\|_{x_k}$  and  $\beta = \|f'(x_k)\|_{x_k}$ . Since  $h$  is a descent step, it is required to be positive. Moreover, if we are not at the minimum of  $f$ , then  $\beta$  will be strictly positive. Therefore,  $\psi$  is defined on the interval  $[0, 1/\beta)$ . Note that if  $h_k \in [0, 1/\beta)$ , then  $x_{k+1} \in W(x_k)$  as required for (3.6) to be a valid bound.



Differentiating  $\psi(h)$  yields

$$\psi'(h) = \alpha - \frac{\beta}{1 - \beta h}, \quad (3.7)$$

$$\psi''(h) = -\frac{\beta^2}{(1 - \beta h)^2} < 0, \quad (3.8)$$

showing that  $\psi(h)$  is concave on its domain  $[0, 1/\beta)$ . It achieves its maximum at

$$h = \frac{\alpha - \beta}{\alpha\beta}. \quad (3.9)$$

Let  $\lambda(x) = \frac{\|f'(x)\|^2}{\|f'(x)\|_x}$ . Substituting  $\alpha$  and  $\beta$  into (3.9), we obtain the rule

$$h_k = \frac{\lambda_k}{(1 + \lambda_k)\|f'(x_k)\|_{x_k}} \quad (3.10)$$

where  $\lambda_k = \lambda(x_k) = \frac{\|f'(x_k)\|^2}{\|f'(x_k)\|_{x_k}}$ .

This motivates us to define the following damped gradient method.

**Algorithm 3. (Damped Gradient Algorithm)**

step 0: Select an initial point  $x_0 \in Q$  and set  $k = 0$ .

step k: If  $f'(x_k) = 0$  then terminate. Otherwise, set

$$\begin{aligned} \lambda_k &= \frac{\|f'(x_k)\|^2}{\|f'(x_k)\|_{x_k}}, \\ h_k &= \frac{\lambda_k}{(1 + \lambda_k)\|f'(x_k)\|_{x_k}}, \\ x_{k+1} &= x_k - h_k f'(x_k). \end{aligned}$$

Increment  $k$  and repeat until convergence.

The convergence of Algorithm 3 is proved in Theorem 3.3.1 with the help of Lemma 2.

**Lemma 2.** Let  $\{x_k\}$  be a sequence generated by Algorithm 3, where the cost function  $f : Q \rightarrow R$  in (3.2) satisfies Assumption 1 in Chapter 1. Then:

1.  $\forall k, x_k \in Q$ .

2. If  $f'(x_k) \neq 0$ , then  $f(x_{k+1}) \leq f(x_k) - \omega(\lambda_k) < f(x_k)$ , where  $\omega(t) = t - \ln(1 + t)$ .
3. Let  $x^* \in Q$  be the solution of (3.2). If  $x \in Q$  satisfies  $x \neq x^*$ , then  $\lambda(x) > 0$ . Moreover,  $\lim_{x \rightarrow x^*} \lambda(x) = 0$ .

**Proof:**

1. From the earlier derivation, it was already proved that

$$x_{k+1} \in W(x_k). \quad (3.11)$$

Therefore, from (2.9) and the fact that  $x_0 \in Q$ , it follows that  $x_k \in Q$ .

2. Substituting  $h_k$  into (3.6), we obtain

$$f(x_{k+1}) \leq f(x_k) - \omega(\lambda_k) \quad (3.12)$$

where  $\omega(t) = t - \ln(1 + t)$ . Since  $f'(x_k) \neq 0$  implies  $\lambda_k > 0$  and, from (2.13),  $w(t) > 0$  for  $t > 0$ , the result follows.

3. Recall that Assumption 1 implies  $x^*$  is unique. Therefore  $f'(x) = 0$  only and only if  $x = x^*$ . Suppose  $x \in Q$  but  $x \neq x^*$ . Since  $f''(x)$  is positive definite by Assumption 1 and  $f'(x) \neq 0$  we have  $\lambda(x) > 0$ .

Let  $K = \{x \mid \|x - x^*\| \leq \rho\}$  where  $\rho > 0$  is sufficiently small such that  $K \subset Q$ . Let  $\nu_{\min}(f''(x))$  denote the minimum eigenvalue of the Hessian matrix  $f''(x)$ . Then  $\theta = \min_{x \in K} \nu_{\min}(f''(x))$  exists since  $K$  is compact and  $\nu_{\min}(f''(x))$  is continuous. Since  $f''(x)$  is positive definite on  $Q$  by Assumption 1,  $\theta > 0$ . Moreover, for any  $x \in K$  and any direction  $u \in \mathbb{R}^n$ , we have

$$u^T f''(x) u = \|u\|_x^2 \geq \theta \|u\|^2. \quad (3.13)$$

Therefore, for  $x \in K$ , we obtain

$$\begin{aligned} \lambda(x) &= \frac{\|f'(x)\|^2}{\|f'(x)\|_x} \\ &\leq \frac{1}{\sqrt{\theta}} \|f'(x)\|. \end{aligned} \quad (3.14)$$

Therefore, at  $x^*$ , we get

$$0 \leq \lim_{x \rightarrow x^*} \lambda(x) \leq \lim_{x \rightarrow x^*} \frac{1}{\sqrt{\theta}} \|f'(x)\|. \quad (3.15)$$

Since  $f'(x)$  is continuous and  $f'(x^*) = 0$ ,  $\lim_{x \rightarrow x^*} \lambda(x) = 0$ .

□

**Theorem 3.3.1.** *Consider the constrained optimization problem in (3.2). If the cost function  $f : Q \rightarrow R$  in (3.2) satisfies Assumption 1 in Chapter 1, then Algorithm 3 converges to the unique minimum of  $f$ .*

**Proof:** Let  $K = \{y \mid f(y) \leq f(x_0)\}$  where  $x_0$  denotes the initial point. Let  $x^*$  be the solution of (3.2). Then for any  $y \in K$  in view of (2.13), we have

$$f(y) \geq f(x^*) + \omega(\|y - x^*\|_{x^*}). \quad (3.16)$$

It follows from (3.16) that

$$\omega(\|y - x^*\|_{x^*}) \leq f(y) - f(x^*) \leq f(x_0) - f(x^*); \quad (3.17)$$

Note that  $\omega(t)$  is strictly increasing in  $t$ . Therefore,  $\|y - x^*\|_{x^*} \leq \bar{t}$  where  $\bar{t}$  is the unique positive root of the following equation

$$\omega(t) = f(x_0) - f(x^*). \quad (3.18)$$

Thus,  $K$  is closed bounded and hence compact.

From Lemma 2, we have

$$f(x_{k+1}) \leq f(x_k) - \omega(\lambda_k). \quad (3.19)$$

Summing up the inequalities (3.19) for  $k = 0 \dots N$ , we obtain

$$\sum_{k=0}^N \omega(\lambda_k) \leq f(x_0) - f(x_{N+1}) \leq f(x_0) - f(x^*), \quad (3.20)$$

where  $x^*$  is the solution of (3.2). As a consequence of (3.20), we have

$$\omega(\lambda_k) \rightarrow 0 \text{ as } k \rightarrow \infty \quad (3.21)$$

and therefore

$$\lambda_k \rightarrow 0 \text{ as } k \rightarrow \infty. \quad (3.22)$$

Since  $f(x_k)$  decreases as  $k$  increases,  $x_k \in K$  for all  $k$ . Assume that  $x_k \rightarrow x^o$  as  $k \rightarrow \infty$  where  $x^o$  is not the solution of (3.2). Then it follows from Lemma 2 that  $\lambda_k \rightarrow \lambda(x^o) > 0$ , which is contradictory to (3.22). Therefore, the theorem follows.  $\square$

### 3.3.2 The Damped Conjugate Gradient Method

Let  $f$  in (3.2) satisfy Assumption 1 in Chapter 1. We construct a damped conjugate gradient method to solve (3.2).

Suppose we have a point  $x_k$  at time  $k$ . Given an appropriate step-size  $h_k$  and conjugate gradient direction  $H_k$  defined in (3.5), the conjugate gradient method sets  $x_{k+1} = x_k + h_k H_k$ . Similar to the derivation of the damped gradient method, we propose choosing  $h_k$  to maximize the bound in (2.14). In Theorem 3.3.2, it is proved that such a strategy guarantees convergence to the minimum of the cost function.

From (2.14), provided  $x_{k+1} \in W(x_k)$ , we have

$$f(x_k) - f(x_{k+1}) \geq -h_k \langle f'(x_k), H_k \rangle + \|h_k H_k\|_{x_k} + \ln(1 - \|h_k H_k\|_{x_k}). \quad (3.23)$$

Initially we assume  $\langle f'(x_k), H_k \rangle < 0$ . Later, in Lemma 3, it is proved that this assumption is correct. Hence,  $h_k$  is required to be positive.

The right hand side of (3.23) is of the form  $\psi(h_k)$  where  $\psi(h) = \alpha h + \ln(1 - \beta h)$  with  $\alpha = -\langle f'(x_k), H_k \rangle + \|H_k\|_{x_k}$  and  $\beta = \|H_k\|_{x_k}$ .

As before,  $\psi(h)$  is defined on the interval  $[0, 1/\beta)$ , and if  $h_k \in [0, 1/\beta)$ , then  $x_{k+1} \in W(x_k)$  as required for (3.23) to be a valid bound. Recall that  $\psi(h)$  achieves its maximum at  $h = \frac{\alpha - \beta}{\alpha\beta}$ .

Let  $\lambda_k = \frac{|\langle f'(x_k), H_k \rangle|}{\|H_k\|_{x_k}}$  and  $\lambda(x) = \frac{\langle f'(x), H \rangle}{\|H\|_x}$ . Substituting  $\alpha$  and  $\beta$  into  $h$ , we obtain the rule

$$h_k = \frac{\lambda_k}{(1 + \lambda_k)\|H_k\|_{x_k}}. \quad (3.24)$$

Therefore, the proposed damped conjugate gradient algorithm for (3.2) is as follows.

**Algorithm 4. (Damped Conjugate Gradient Algorithm)**

step 0: Select an initial point  $x_0$ , compute  $H_0 = -f'(x_0)$ , and set  $k = 0$ .

step k: If  $f'(x_k) = 0$ , then terminate. Otherwise, set

$$\lambda_k = \frac{|\langle f'(x_k), H_k \rangle|}{\|H_k\|_{x_k}}, \quad (3.25)$$

$$h_k = \frac{\lambda_k}{(1 + \lambda_k)\|H_k\|_{x_k}}, \quad (3.26)$$

$$x_{k+1} = x_k + h_k H_k, \quad (3.27)$$

$$\gamma_{k+1} = \frac{\|f'(x_{k+1})\|^2}{-\langle f'(x_k), H_k \rangle}, \quad (3.28)$$

$$H_{k+1} = -f'(x_{k+1}) + \gamma_{k+1} H_k. \quad (3.29)$$

*Increment  $k$  and repeat until convergence.*

The convergence of Algorithm 4 is proved in Theorem 3.3.2 with the help of Lemma 3, 4 and 5.

**Lemma 3.** *Let the cost function  $f : Q \rightarrow R$  in (3.2) satisfy Assumption 1 in Chapter 1. Assume  $x_0 \in Q$  is such that  $f'(x_0) \neq 0$ . Then Algorithm 4 generates an infinite sequence  $\{x_k\}$  (that is, there are no divisions by zeros). Moreover,  $\forall k, \langle f'(x_k), H_k \rangle < 0$  provided  $f'(x_k) \neq 0$ .*

**Proof:** This proof is by induction. When  $k = 0$ ,  $H_0 = -f'(x_0)$ . Then we obtain

$$\langle f'(x_0), H_0 \rangle = -\|f'(x_0)\|^2 < 0 \quad (3.30)$$

where the inequality follows from the fact that  $x_0$  is not the solution of (3.2).

Assume that  $\langle f'(x_k), H_k \rangle < 0$  for some  $k$ . It follows that  $x_{k+1}$  is well defined.

$$h_k = \frac{\lambda_k}{(1 + \lambda_k)\|H_k\|_{x_k}} = -\frac{\langle f'(x_k), H_k \rangle}{(|\langle f'(x_k), H_k \rangle| + \|H_k\|_{x_k})\|H_k\|_{x_k}}. \quad (3.31)$$

Moreover we have

$$\begin{aligned} \langle f'(x_{k+1}), H_k \rangle &= \langle f'(x_k), H_k \rangle + \frac{\langle f'(x_{k+1}) - f'(x_k), H_k \rangle}{\langle f'(x_k), H_k \rangle} \langle f'(x_k), H_k \rangle \\ &= \rho_k \langle f'(x_k), H_k \rangle. \end{aligned} \quad (3.32)$$

where  $\rho_k = 1 + \frac{\langle f'(x_{k+1}) - f'(x_k), H_k \rangle}{\langle f'(x_k), H_k \rangle}$ .

Furthermore,

$$\begin{aligned} f'(x_{k+1}) - f'(x_k) &= \int_0^1 f''(x_k + \tau(x_{k+1} - x_k))(x_{k+1} - x_k) d\tau \\ &= h_k M_k H_k \end{aligned} \quad (3.33)$$

where  $M_k = \int_0^1 f''(x_k + \tau(x_{k+1} - x_k)) d\tau$  and  $M_k$  is positive definite because  $f''(x)$  is positive definite.

In view of (2.12), we have

$$M_k \preceq (1 + \lambda_k) f''(x_k). \quad (3.34)$$

Substituting (3.33) into the second part of  $\rho_k$ , we obtain

$$\frac{\langle f'(x_{k+1}) - f'(x_k), H_k \rangle}{\langle f'(x_k), H_k \rangle} = \frac{h_k H_k^T M_k H_k}{\langle f'(x_k), H_k \rangle} \quad (3.35)$$

$$\begin{aligned} &= \frac{H_k^T M_k H_k}{(|\langle f'(x_k), H_k \rangle| + \|H_k\|_{x_k})\|H_k\|_{x_k}} \\ &\geq \frac{(1 + \lambda_k)\|H_k\|_{x_k}^2}{(|\langle f'(x_k), H_k \rangle| + \|H_k\|_{x_k})\|H_k\|_{x_k}} \\ &= \frac{1 + \lambda_k}{1 + \lambda_k} \\ &= -1 \end{aligned} \quad (3.36)$$

where (3.31) was substituted for  $h_k$  to obtain the second equality.

Because  $M_k$  is positive definite and  $\langle f'(x_k), H_k \rangle < 0$  by assumption, it follows from (3.35) that

$$\frac{\langle f'(x_{k+1}) - f'(x_k), H_k \rangle}{\langle f'(x_k), H_k \rangle} < 0. \quad (3.37)$$

Therefore, we obtain

$$0 \leq \rho_k < 1. \quad (3.38)$$

For the conjugate gradient method, since  $\gamma_{k+1} = \frac{\|f'(x_{k+1})\|^2}{-\langle f'(x_k), H_k \rangle}$ , we have

$$\begin{aligned} \langle f'(x_{k+1}), H_{k+1} \rangle &= \langle f'(x_{k+1}), -f'(x_{k+1}) + \frac{\|f'(x_{k+1})\|^2}{-\langle f'(x_k), H_k \rangle} H_k \rangle \\ &= -\|f'(x_{k+1})\|^2 - \|f'(x_{k+1})\|^2 \frac{\langle f'(x_{k+1}), H_k \rangle}{\langle f'(x_k), H_k \rangle} \\ &= -\|f'(x_{k+1})\|^2 - \|f'(x_{k+1})\|^2 \frac{\rho_k \langle f'(x_k), H_k \rangle}{\langle f'(x_k), H_k \rangle} \\ &= -(1 + \rho_k) \|f'(x_{k+1})\|^2 < 0. \end{aligned} \quad (3.39)$$

Consequently, this lemma follows. □

**Lemma 4.** *Let  $\{x_k\}$  be a sequence generated by Algorithm 4 where the cost function  $f : Q \rightarrow R$  in (3.2) satisfies Assumption 1 in Chapter 1. Then:*

1.  $\forall k, x_k \in Q$ .
2. If  $f'(x_k) \neq 0$ , then  $\lambda_k > 0$ .
3. If  $f'(x_k) \neq 0$ , then  $f(x_{k+1}) \leq f(x_k) - \omega(\lambda_k) < f(x_k)$ , where  $\omega(t) = t - \ln(1 + t)$ .

**Proof:**

1. From the earlier derivation, it was already proved that

$$x_{k+1} \in W(x_k). \quad (3.40)$$

Therefore, from (2.9) and the fact that  $x_0 \in Q$ , it follows that  $x_k \in Q$ .

2. Since  $f'(x_k) \neq 0$ , it follows from Lemma 3 that  $\langle f'(x_k), H_k \rangle < 0$ . Then it implies that  $\lambda_k > 0$  by the definition of  $\lambda_k$ .
3. Substituting  $h_k$  into (3.23), we obtain that

$$f(x_{k+1}) \leq f(x_k) - \omega(\lambda_k) < f(x_k) \quad (3.41)$$

where  $\omega(t) = t - \ln(1 + t) > 0$  since  $\lambda_k > 0$  by the above proof. □

**Lemma 5.** *Let  $\{x_k\}$  and  $\{H_k\}$  be sequences generated by Algorithm 4 where the cost function  $f : Q \rightarrow R$  in (3.2) satisfies Assumption 1 in Section 2. If  $f'(x_k) \neq 0$ , then for all  $k$*

$$\frac{\|H_{k+1}\|^2}{\|f'(x_{k+1})\|^4} \leq \frac{\|H_k\|^2}{\|f'(x_k)\|^4} + \frac{3}{\|f'(x_{k+1})\|^2}. \quad (3.42)$$

**Proof:** Note that from (3.39) and the inequality (3.38), we have

$$(\langle f(x_{k+1}), H_{k+1} \rangle)^2 = (1 + \rho_k)^2 \|f'(x_{k+1})\|^4 \geq \|f'(x_{k+1})\|^4. \quad (3.43)$$

In view of (3.29), (3.28), (3.32) and (3.43), we obtain

$$\begin{aligned} \|H_{k+1}\|^2 &= \| -f'(x_{k+1}) + \gamma_k H_k \|^2 \\ &= \| -f'(x_{k+1}) + \frac{\|f'(x_{k+1})\|^2}{-\langle f'(x_k), H_k \rangle} H_k \|^2 \\ &= \|f'(x_{k+1})\|^2 + \frac{\|f'(x_{k+1})\|^4}{(\langle f'(x_k), H_k \rangle)^2} \|H_k\|^2 + 2 \frac{\|f'(x_{k+1})\|^2}{\langle f'(x_k), H_k \rangle} \langle f'(x_{k+1}), H_k \rangle \\ &= (1 + 2\rho_k) \|f'(x_{k+1})\|^2 + \frac{\|f'(x_{k+1})\|^4}{(\langle f'(x_k), H_k \rangle)^2} \|H_k\|^2 \\ &\leq 3 \|f'(x_{k+1})\|^2 + \frac{\|f'(x_{k+1})\|^4}{\|f'(x_k)\|^4} \|H_k\|^2. \end{aligned} \quad (3.44)$$

It follows via dividing both sides of (3.44) by  $\|f'(x_{k+1})\|^4$  that

$$\frac{\|H_{k+1}\|^2}{\|f'(x_{k+1})\|^4} \leq \frac{\|H_k\|^2}{\|f'(x_k)\|^4} + \frac{3}{\|f'(x_{k+1})\|^2} \quad (3.45)$$

□



**Theorem 3.3.2.** *Consider the constrained optimization problem in (3.2). If the cost function  $f : Q \rightarrow R$  in (3.2) satisfies Assumption 1 in Section 2, then Algorithm 4 converges to the unique minimum of  $f$ .*

**Proof:** Let  $K = \{y \mid f(y) \leq f(x_0)\}$  where  $x_0$  denotes the initial point. Let  $x^*$  be the solution of (3.2). Then for any  $y \in K$  in view of (2.13), we have

$$f(y) \geq f(x^*) + \omega(\|y - x^*\|_{x^*}). \quad (3.46)$$

It follows from (3.46) that

$$\omega(\|y - x^*\|_{x^*}) \leq f(y) - f(x^*) \leq f(x_0) - f(x^*); \quad (3.47)$$

Note that  $\omega(t)$  is strictly increasing in  $t$ . Therefore,  $\|y - x^*\|_{x^*} \leq \bar{t}$  where  $\bar{t}$  is the unique positive root of the following equation

$$\omega(t) = f(x_0) - f(x^*). \quad (3.48)$$

Thus,  $K$  is closed bounded and hence compact.

Let  $\nu_{\max}(f''(x))$  denote the maximum eigenvalue of the Hessian matrix  $f''(x)$ . Then  $\theta = \max_{x \in K} \nu_{\max}(f''(x))$  exists since  $K$  is compact and  $\nu_{\max}(f''(x))$  is continuous. Because  $f''(x)$  is positive definite on  $Q$  by Assumption 1,  $\theta > 0$ . Moreover, for any  $x \in K$  and any direction  $u \in \mathbb{R}^n$ , we have

$$u^T f''(x)u = \|u\|_x^2 \leq \theta \|u\|^2. \quad (3.49)$$

Hence by the definition of  $\lambda_k$ , we obtain

$$\begin{aligned} \lambda_k &= \frac{|\langle f'(x_k), H_k \rangle|}{\|H_k\|_{x_k}} \\ &\geq \frac{|\langle f'(x_k), H_k \rangle|}{\sqrt{\theta} \|H_k\|}. \end{aligned} \quad (3.50)$$

By (3.38) and (3.39), (3.50) becomes

$$\lambda_k \geq \frac{\|f'(x_k)\|^2}{\sqrt{\theta} \|H_k\|}. \quad (3.51)$$

From Lemma 4, we have

$$f(x_{k+1}) \leq f(x_k) - \omega(\lambda_k). \quad (3.52)$$

Summing up the inequalities (3.52) for  $k = 0 \dots N$ , we obtain

$$\sum_{k=0}^N \omega(\lambda_k) \leq f(x_0) - f(x_{N+1}) \leq f(x_0) - f(x^*), \quad (3.53)$$

where  $x^*$  is the solution of (3.2). As a consequence of (3.53), we have

$$\sum_{k=0}^{\infty} \omega(\lambda_k) < +\infty. \quad (3.54)$$

Assume  $\liminf_{k \rightarrow \infty} \|f'(x_k)\| \neq 0$ . Then there exists  $\alpha > 0$  such that  $\|f'(x_k)\| \geq \alpha$  for infinitely often indexes  $k$ . Therefore, it follows from Lemma 5

$$\frac{\|H_i\|^2}{\|f'(x_i)\|^4} \leq \frac{\|H_{i-1}\|^2}{\|f'(x_{i-1})\|^4} + \frac{3}{\alpha^2}. \quad (3.55)$$

Summing up the above inequalities for  $i = 0, \dots, k$ , we get

$$\frac{\|H_k\|^2}{\|f'(x_k)\|^4} \leq \frac{\|H_0\|^2}{\|f'(x_0)\|^4} + \frac{3k}{\alpha^2}. \quad (3.56)$$

Let  $a = \frac{3}{\alpha^2}$  and  $b = \frac{\|H_0\|^2}{\|f'(x_0)\|^4} = \frac{1}{\|f'(x_0)\|^2}$ . Then it follows from (3.56)

$$\frac{\|f'(x_k)\|^4}{\|H_k\|^2} \geq \frac{1}{ka + b}. \quad (3.57)$$

Combining (3.51) and (3.57), we obtain

$$\lambda_k \geq \frac{c}{\sqrt{ka + b}} \quad (3.58)$$

where  $c = \frac{1}{\sqrt{\theta}}$ .

Let  $\{\beta_k\}$  be a sequence such that  $\beta_k = \frac{c}{\sqrt{ka+b}}$ . Then it is easy to show

$$\sum_{k=1}^{\infty} \beta_k^2 = +\infty. \quad (3.59)$$

Consider the sequence  $\{\omega(\beta_k)\}$ . Since  $a, b, c$  are constant, we have

$$\lim_{k \rightarrow \infty} \frac{\omega(\beta_k)}{\beta_k^2} = \lim_{t \rightarrow 0} \frac{t - \ln(1+t)}{t^2} = \frac{1}{2}. \quad (3.60)$$

It follows from (3.59) and (3.60)

$$\sum_{k=1}^{\infty} \omega(\beta_k) = +\infty. \quad (3.61)$$

Since  $\omega(t)$  is increasing with respect to  $t$ , by (3.58) and (3.61) we obtain

$$\sum_{k=1}^{\infty} \omega(\lambda_k) = +\infty \quad (3.62)$$

which is contradictory to (3.54). Therefore, we have

$$\liminf_{k \rightarrow \infty} \|f'(x_k)\| = 0. \quad (3.63)$$

Hence, the theorem follows.  $\square$

## 3.4 Interior Point Method

Given  $c \in \mathbb{R}^n$  and  $\alpha_i \in \mathbb{R}$ ,  $i = 1, \dots, m$ , we consider the following convex programming problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^T x, \\ \text{s.t.} \quad & f_i(x) \leq \alpha_i, \quad i = 1, \dots, m, \end{aligned} \quad (3.64)$$

where all functions  $f_i$ ,  $i = 1, \dots, m$  are convex. We also assume that this problem satisfies the Slater condition: There exists  $\bar{x} \in \mathbb{R}^n$  such that  $f_i(\bar{x}) < \alpha_i$  for all  $i = 1, \dots, m$ .

To apply the interior point method to this problem, we are required to construct the self-

concordant barrier for the domain. Let us assume that there exist standard self-concordant barriers  $F_i(x)$  for the inequality constraints  $f_i(x) \leq \alpha_i$ . Then the resulting barrier function for this problem is as follows

$$F(x) = \sum_{i=1}^m F_i(x). \quad (3.65)$$

Recall the framework of the interior point method in Chapter 1. Given a sequence  $\{\mu_t\}$  such that  $\mu_t > 0$  and  $\lim_{t \rightarrow \infty} \mu_t = 0$ , we minimize the new cost function  $f(x; \mu_t)$

$$f(x; \mu_t) = \frac{1}{\mu_t} c^T x + F(x) \quad (3.66)$$

in sequence and use the solution of current minimization problem as the initial guess for the next optimization. As  $\mu_t$  goes to zero in the limit, we obtain the minimum of the original problem.

In fact, as stated in [10], it is not necessary to obtain the exact minimum of the cost function  $f(x; \mu_t)$  for every given  $\mu_t$ . A common way used in practice is to perform one step or several steps of Newton method and then go to the next  $\mu$ . For more details, see [10].

Recall that the Newton method requires expensive computational cost involved in computing the inverse of the Hessian matrix of the cost function. Therefore, the gradient-based methods are preferred for large scale problems due to avoid computing the inverse of the Hessian matrix. Since  $c^T x$  is linear and  $F(x)$  is standard self-concordant,  $f(x; \mu_t)$  is still standard self-concordant for all  $\mu_t > 0$  by Corollary 4.1.1 in [57]. Consequently, instead of the Newton method, the damped conjugate gradient method can be used to minimize  $f(x; \mu_t)$  as follows.

**Algorithm 5. (Damped Conjugate Gradient Algorithm)**

step 0: Select an initial point  $x_0$  satisfying the constraints in (3.64), compute  $H_0 = -f'(x_0; \mu_t)$ , and set  $k = 0$ .

step k: If  $f'(x_k; \mu_t) = 0$ , then terminate. Otherwise, set

$$\lambda_k = \frac{|\langle f'(x_k; \mu_t), H_k \rangle|}{\|H_k\|_{x_k}}, \quad (3.67)$$

$$h_k = \frac{\lambda_k}{(1 + \lambda_k)\|H_k\|_{x_k}}, \quad (3.68)$$

$$x_{k+1} = x_k + h_k H_k, \quad (3.69)$$

$$\gamma_{k+1} = \frac{\|f'(x_{k+1}; \mu_t)\|^2}{-\langle f'(x_k; \mu_t), H_k \rangle}, \quad (3.70)$$

$$H_{k+1} = -f'(x_{k+1}; \mu_t) + \gamma_{k+1} H_k, \quad (3.71)$$

where  $\|H_k\|_{x_k} = \sqrt{H_k^T f''(x_k; \mu_t) H_k}$ .

Increment  $k$  and repeat until convergence.

In practice, for the interior point method, we may need just a few damped conjugate gradient iterations to reach the required accuracy. By the proof of Theorem 3.3.2, the sequence  $\{\lambda_k\}$  generated by Algorithm 5 satisfies  $\lambda_k \rightarrow 0$  if  $x \rightarrow x^*$  where  $x^*$  is the minimum of  $f(x; \mu_t)$ . Therefore, we can set the stopping criterium based on the value of  $\lambda_k$ . That is if  $\lambda_k$  is small enough, we go then to next  $\mu$ .

As a result, we give the proposed barrier interior point algorithm based on the truncated damped conjugate gradient method as follows.

**Algorithm 6. (Barrier Interior Point Algorithm)**

Cycle 0: Set  $\mu_0 = 1$  and the cycle counter  $t = 0$ . Select an initial point  $x_0$  satisfying the inequality constraints in (3.64).

Cycle  $t$ :

step 0: Compute the gradient  $f'(x_0; \mu_t)$  and  $\lambda_0 = \frac{\|f'(x_0; \mu_t)\|^2}{\|f'(x_0; \mu_t)\|_{x_0}^2}$ , set  $H_0 = -f'(x_0; \mu_t)$  and  $k = 0$ .

step  $k$ : If  $f'(x_k; \mu_t) = 0$ , then terminate. Otherwise, set

$$\begin{aligned} \lambda_k &= \frac{|\langle f'(x_k; \mu_t), H_k \rangle|}{\|H_k\|_{x_k}}, \\ h_k &= \frac{\lambda_k}{(1 + \lambda_k) \|H_k\|_{x_k}}, \\ x_{k+1} &= x_k + h_k H_k, \end{aligned}$$

where  $\|H_k\|_{x_k} = \langle f''(x_k; \mu_t) H_k, H_k \rangle$  and  $f''(x_k; \mu_t)$  is the Hessian of  $f(x_k; \mu_t)$ .

Set

$$\begin{aligned} \gamma_{k+1} &= \frac{\|f'(x_{k+1}; \mu_t)\|^2}{-\langle f'(x_k; \mu_t), H_k \rangle}, \\ H_{k+1} &= -f'(x_{k+1}; \mu_t) + \gamma_{k+1} H_k. \end{aligned}$$

Increment  $k$  and repeat until  $\lambda_k < \epsilon_1$  where  $\epsilon_1 > 0$  is a predefined tolerance.

Set  $x_0 = x_k$  and  $\mu_{t+1} = \theta\mu_t$  where  $\theta$  is a constant satisfying  $0 < \theta < 1$ . Then increment  $t$  and repeat until  $\mu < \epsilon_2$  where  $\epsilon_2 > 0$  is also a predefined tolerance.

**Remark 1.** The damped conjugate gradient of the inner iteration in Algorithm 6 is stopped when  $\epsilon_1$  is small enough. In practice, the choice of  $\epsilon_1 = 5 \times 10^{-2}$  is sufficient; that is proved by the numerical simulations in the next chapter.

## 3.5 Numerical Examples

In this section, we apply our algorithms to self-concordant functions to show convergence of the damped gradient and conjugate gradient methods.

### 3.5.1 Example One

Consider the following quadratically constrained quadratic optimization problem (QCQOP)

$$\min_{x \in \mathbb{R}^n} \quad q_0(x) = \alpha_0 + a_0^T x + \frac{1}{2} x^T A_0 x, \quad (3.72)$$

$$\text{subject to:} \quad q_i(x) = \alpha_i + a_i^T x + \frac{1}{2} x^T A_i x \leq \beta_i, i = 1, \dots, m, \quad (3.73)$$

where  $A_i, i = 0, \dots, m$  are positive semidefinite  $(n \times n)$ -matrices,  $x, a_i \in \mathbb{R}^n$  and  $\alpha_i \in \mathbb{R}$  and the superscript  $T$  denotes the transpose.

By Equation (4.3.4) in [57], the QCQOP problem could be rewritten in the same form as (3.64):

$$\min_{x \in \mathbb{R}^n, \tau \in \mathbb{R}} \quad \tau, \quad (3.74)$$

$$\text{subject to:} \quad q_0(x) \leq \tau, \quad (3.75)$$

$$q_i(x) \leq \beta_i, i = 1, \dots, m. \quad (3.76)$$

We use the logarithmic barrier technique to transform the constrained problem into an unconstrained one. The additional logarithmic barrier term for this problem is

$$F(x, \tau) = -\ln(\tau - q_0(x)) - \sum_{i=1}^m \ln(\beta_i - q_i(x)). \quad (3.77)$$

Combining  $\tau$  and the barrier term, we get the new optimization problem

$$\min_{x \in \mathbb{R}^n, \tau \in \mathbb{R}} f(x, \tau; \mu) = \frac{1}{\mu} \tau + F(x, \tau). \quad (3.78)$$

It is shown in [57] that  $f(x, \tau; \mu)$  in (3.78) satisfies Assumption 1 in Chapter 1 for all  $\mu > 0$ .

For a given  $\mu > 0$ , it is easy to compute the gradient  $f'(x, \tau; \mu)$  and the Hessian matrix  $f''(x, \tau; \mu)$  as follows

$$f'(x, \tau; \mu) = \begin{bmatrix} \frac{q'_0(x)}{\tau - q_0(x)} + \sum_{i=1}^m \frac{q'_i(x)}{\beta_i - q_i(x)} \\ \frac{1}{\mu} - \frac{1}{\tau - q_0(x)} \end{bmatrix} \quad (3.79)$$

$$f''(x, \tau; \mu) = \begin{bmatrix} \frac{q''_0(x)}{\tau - q_0(x)} + \frac{q'_0(x)q_0{}^T(x)}{(\tau - q_0(x))^2} + \sum_{i=1}^m \left( \frac{q''_i(x)}{\beta_i - q_i(x)} + \frac{q'_i(x)q_i{}^T(x)}{(\beta_i - q_i(x))^2} \right) & -\frac{q'_0(x)}{(\tau - q_0(x))^2} \\ -\frac{q_0{}^T(x)}{(\tau - q_0(x))^2} & \frac{1}{(\tau - q_0(x))^2} \end{bmatrix} \quad (3.80)$$

where  $q'_i(x) = A_i x + a_i$  and  $q''_i(x) = A_i$  for  $i = 0, \dots, m$ .

Given a sequence  $\{\mu_t\}$  such that  $\mu_t > 0$  for all  $t$  and  $\lim_{t \rightarrow \infty} \mu_t = 0$ , we minimize  $f(x; \mu_t)$  in sequence. As  $\mu_t$  goes to zero in the limit, we obtain the minimum of the original problem. In this chapter, we follow the strategy in [72] to choose the sequence  $\{\mu_t\}$ .

The proposed damped gradient algorithm for QCQOP is as follows.

#### Algorithm 7. (Damped Gradient Algorithm for QCQOP)

Cycle 0: Set  $\mu_0 = 1$  and the cycle counter  $t = 0$ . Select an initial point  $x_0$  satisfying the inequality constraints in (3.73).

Cycle t:

step 0: Compute the gradient  $f'(x_0; \mu_t)$  by (3.79), and set  $k = 0$ .

step k: If  $f'(x_k, \tau; \mu_t) = 0$ , then terminate. Otherwise, set

$$\begin{aligned}\lambda_k &= \frac{\|f'(x_k, \tau; \mu_t)\|^2}{\|f'(x_k, \tau; \mu_t)\|_{x_k}}, \\ h_k &= \frac{\lambda_k}{(1 + \lambda_k)\|f'(x_k, \tau; \mu_t)\|_{x_k}}, \\ x_{k+1} &= x_k - h_k f'(x_k, \tau; \mu_t),\end{aligned}$$

where  $\|f'(x_k, \tau; \mu_t)\|_{x_k} = \langle f''(x_k, \tau; \mu_t)f'(x_k, \tau; \mu_t), f'(x_k, \tau; \mu_t) \rangle$  and  $f''(x_k, \tau; \mu_t)$  is computed by (3.80).

Increment  $k$  and repeat until  $\lambda_k < 0.05$ .

Set  $x_0 = x_k$  and  $\mu_{t+1} = \theta\mu_t$  where  $\theta$  is a constant satisfying  $0 < \theta < 1$ . Then increment  $t$  and repeat until  $\mu < \epsilon_2$  where  $\epsilon_2 > 0$  is a predefined tolerance.

The proposed damped conjugate gradient method for QCQOP is as follows.

**Algorithm 8. (Damped Conjugate Gradient Algorithm for QCQOP)**

Cycle 0: Set  $\mu_0 = 1$  and the cycle counter  $t = 0$ . Select an initial point  $x_0$  satisfying the inequality constraints in (3.73).

Cycle t:

step 0: Compute the gradient  $f'(x_0, \tau; \mu_t)$  by (3.79), set  $H_0 = -f'(x_0, \tau; \mu_t)$ , and set  $k = 0$ .

step k: If  $f'(x_k, \tau; \mu_t) = 0$ , then terminate. Otherwise, set

$$\begin{aligned}\lambda_k &= \frac{-\langle f'(x_k, \tau; \mu_t), H_k \rangle}{\|H_k\|_{x_k}}, \\ h_k &= \frac{\lambda_k}{(1 + \lambda_k)\|H_k\|_{x_k}}, \\ x_{k+1} &= x_k + h_k H_k,\end{aligned}$$

where  $\|H_k\|_{x_k} = \langle f''(x_k, \tau; \mu_t)H_k, H_k \rangle$  and  $f''(x_k, \tau; \mu_t)$  is computed by (3.80).

Set

$$\begin{aligned}\gamma_{k+1} &= \frac{\|f'(x_{k+1}, \tau; \mu_t)\|^2}{-\langle f'(x_k, \tau; \mu_t), H_k \rangle}, \\ H_{k+1} &= -f'(x_{k+1}, \tau; \mu_t) + \gamma_{k+1}H_k.\end{aligned}$$

Increment  $k$  and repeat until  $\lambda_k < 0.05$ .



Set  $x_0 = x_k$  and  $\mu_{t+1} = \theta\mu_t$  where  $\theta$  is a constant satisfying  $0 < \theta < 1$ . Then increment  $t$  and repeat until  $\mu_t < \epsilon_2$  where  $\epsilon_2 > 0$  is a predefined tolerance.

We performed these two algorithms for two cases: a good initial guess and a poor initial guess. In addition, the performance of these two algorithms is compared against the traditional line search method and damped Newton method [57]. In particular, we take  $m = 3, n = 400$ .

Figure 3.2 illustrates the results of implementing the inner loop using the traditional line search and damped Newton methods with  $\mu = 1$  and a good initial guess.

Figure 3.3 illustrates the results of implementing the inner loop using the damped gradient, conjugate gradient and Newton methods with  $\mu = 1$  and a good initial guess.

Figure 3.4 illustrates the results of implementing the inner loop using the damped gradient, conjugate gradient and Newton methods with  $\mu = 1$  and a bad initial guess.

Figure 3.5 illustrates the results of implementing the outer loop with  $\mu$  from 1 to 0.

Table 3.1 shows the simulation time and accuracy using the traditional line search and damped Newton methods with  $\mu = 1$  and a good initial guess.

Table 3.2 shows the simulation time and accuracy using the damped gradient, conjugate gradient and Newton methods with  $\mu = 1$  and a good initial guess.

Table 3.3 shows the simulation time and accuracy using the damped gradient, conjugate gradient and Newton methods with  $\mu = 1$  and a bad initial guess.

These times were obtained using a 3.06 GHZ Pentium 4 machine, with 1Gb of memory, running Windows XP Professional.

algorithm	time(second)	accuracy
line search gradient method	695.8440	0.01
line search conjugate gradient method	86.9370	0.001
backtracking gradient method	11.1560	0.001
damped Newton method	1.4370	0.001

Table 3.1: Simulation time and accuracy with good initial guess

Simulation results show the convergence property of damped gradient and conjugate gradient methods. It is easy to see from the above simulation result that the damped gradient method has linear convergence rate and the damped conjugate gradient method super-linear convergence rate. Furthermore, because these two methods provide explicit step-size choice rules, they cost less time than traditional line search and backtracking methods. In addition, due to avoiding

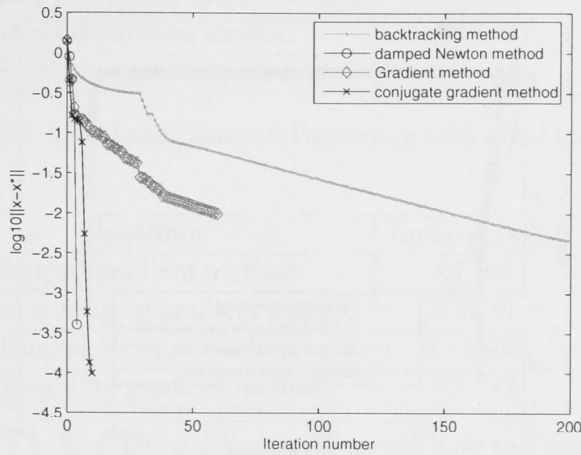


Figure 3.2: Error vs. iteration number with good initial guess for QCQOP from the backtracking gradient method, damped Newton method, line search gradient method and line search conjugate gradient method

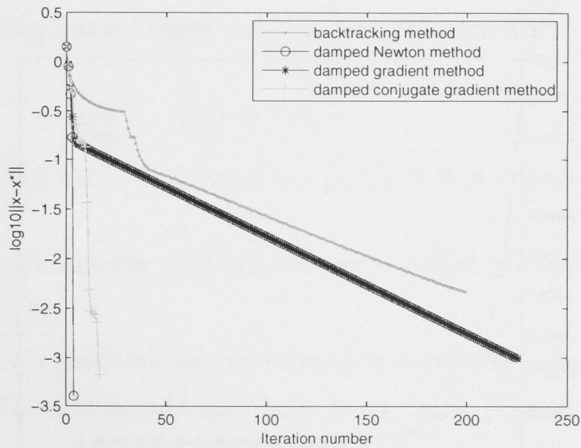


Figure 3.3: Error vs. iteration number with good initial guess for QCQOP from the backtracking gradient method, damped Newton method, damped gradient method and damped conjugate gradient method

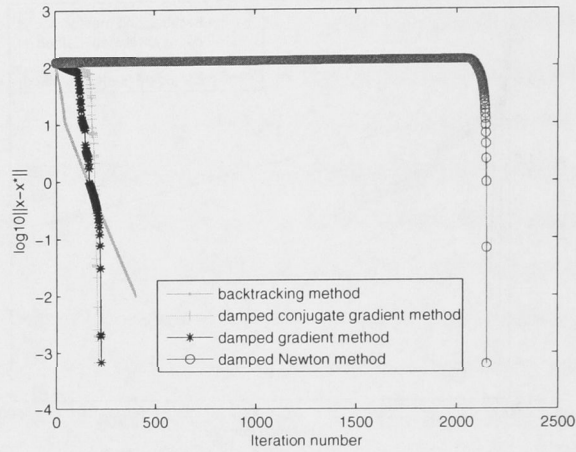


Figure 3.4: Error vs. iteration number with bad initial guess for QCQOP from the backtracking gradient method, damped Newton method, damped gradient method and damped conjugate gradient method

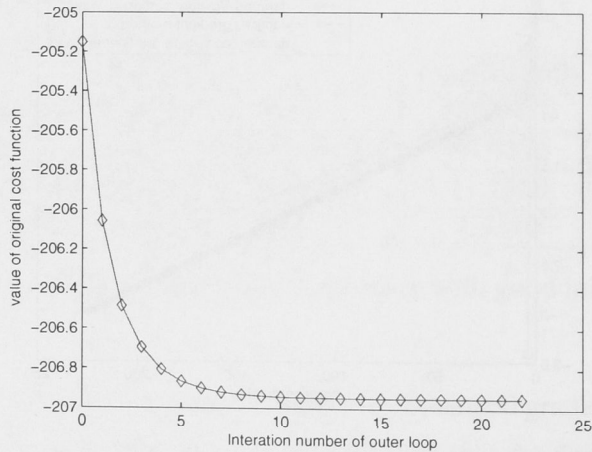


Figure 3.5: The value of the original cost function with  $\mu$  approaching 0 for OCQOP

algorithm	time(second)	accuracy
damped gradient method	10.2820	0.001
damped conjugate gradient method	0.8590	0.001
damped Newton method	1.4370	0.001
backtracking gradient method	11.1560	0.01

Table 3.2: Simulation time and accuracy with good initial guess

algorithm	time(second)	accuracy
damped gradient method	3.1560	0.001
damped conjugate gradient method	3.5470	0.001
damped Newton method	60.1560	0.001
backtracking gradient method	5.8750	0.01

Table 3.3: Simulation time and accuracy with bad initial guess

the computation the inverse of the Hessian matrix, the damped conjugate gradient method costs less time than the damped Newton method. If we have a bad initial guess, the damped gradient method performs better than other methods.

### 3.5.2 Example Two

Consider the following second-order cone programming (SOCP)

$$\min_{x \in \mathbb{R}^n} q(x) = c^T x, \quad (3.81)$$

$$\text{subject to : } \|A_i x + b_i\| \leq c_i^T x + d_i, i = 1, \dots, N, \quad (3.82)$$

where  $x \in \mathbb{R}^n$  is the optimization variable and the problem parameters are  $c, c_i \in \mathbb{R}^n$ ,  $A_i \in \mathbb{R}^{m \times n}$ ,  $b_i \in \mathbb{R}^m$  and  $d_i \in \mathbb{R}$ .

As before, we use the logarithmic barrier technique to transform the constrained problem into an unconstrained one. The additional logarithmic barrier term [72] for this problem is

$$F(x) = - \sum_{i=1}^N \ln((c_i^T x + d_i)^2 - \|A_i x + b_i\|^2). \quad (3.83)$$

Combining  $q(x)$  and the barrier term, we get the new optimization problem

$$\min_{x \in \mathbb{R}^n} f(x; \mu) = \frac{1}{\mu} q(x) + F(x). \quad (3.84)$$

It is shown in [72] that  $f(x; \mu)$  in (3.84) satisfies Assumption 1 in Chapter 1 for all  $\mu > 0$ .

For a given  $\mu > 0$ , the gradient  $f'(x; \mu)$  and the Hessian matrix  $f''(x; \mu)$  are as follows

$$f'(x; \mu) = \frac{1}{\mu} c - \sum_{i=1}^N \frac{2((c_i^T x + d_i)c_i - A_i^T(A_i x + b_i))}{(c_i^T x + d_i)^2 - \|A_i x + b_i\|^2} \quad (3.85)$$

$$f''(x; \mu) = - \sum_{i=1}^N \left( \frac{-4((c_i^T x + d_i)c_i - A_i^T(A_i x + b_i))((c_i^T x + d_i)c_i - A_i^T(A_i x + b_i))^T}{((c_i^T x + d_i)^2 - \|A_i x + b_i\|^2)^2} + \frac{2(c_i c_i^T - A_i^T A_i)}{(c_i^T x + d_i)^2 - \|A_i x + b_i\|^2} \right) \quad (3.86)$$

As before, we minimize  $f(x; \mu_t)$  in sequence for a given sequence  $\{\mu_t\}$  which satisfies  $\mu_t > 0$  and  $\lim_{t \rightarrow \infty} \mu_t = 0$ . As  $\mu_t$  goes to zero in the limit, we obtain the minimum of the original problem.

The proposed damped gradient algorithm for SOCP is as follows.

**Algorithm 9. (Damped Gradient Algorithm for SOCP)**

Cycle 0: Set  $\mu_0 = 1$  and the cycle counter  $t = 0$ . Select an initial point  $x_0$  satisfying the inequality constraints in (3.82).

Cycle t:

step 0: Compute the gradient  $f'(x_0; \mu_t)$  by (3.85), and set  $k = 0$ .

step k: If  $f'(x_k; \mu_t) = 0$ , then terminate. Otherwise, set

$$\begin{aligned} \lambda_k &= \frac{\|f'(x_k; \mu_t)\|^2}{\|f'(x_k; \mu_t)\|_{x_k}}, \\ h_k &= \frac{\lambda_k}{(1 + \lambda_k)\|f'(x_k; \mu_t)\|_{x_k}}, \\ x_{k+1} &= x_k - h_k f'(x_k; \mu_t), \end{aligned}$$

where  $\|f'(x_k; \mu_t)\|_{x_k} = \langle f''(x_k; \mu_t) f'(x_k; \mu_t), f'(x_k; \mu_t) \rangle$  and  $f''(x_k; \mu_t)$  is computed by (3.86).

Increment  $k$  and repeat until  $\lambda_k < 0.05$ .

Set  $x_0 = x_k$  and  $\mu_{t+1} = \theta \mu_t$  where  $\theta$  is a constant satisfying  $0 < \theta < 1$ . Then increment  $t$  and repeat until  $\mu < \epsilon_2$  where  $\epsilon_2 > 0$  is a predefined tolerance.

The proposed damped conjugate gradient method for SOCP is as follows.

**Algorithm 10. (Damped Conjugate Gradient Algorithm for SOCP)**

Cycle 0: Set  $\mu_0 = 1$  and the cycle counter  $t = 0$ . Select an initial point  $x_0$  satisfying the inequality constraints in (3.82).

Cycle t:

step 0: Compute the gradient  $f'(x_0; \mu_t)$  by (3.85), set  $H_0 = -f'(x_0; \mu_t)$ , and set  $k = 0$ .

step k: If  $f'(x_k; \mu_t) = 0$ , then terminate. Otherwise, set

$$\begin{aligned}\lambda_k &= \frac{|\langle f'(x_k; \mu_t), H_k \rangle|}{\|H_k\|_{x_k}}, \\ h_k &= \frac{\lambda_k}{(1 + \lambda_k)\|H_k\|_{x_k}}, \\ x_{k+1} &= x_k + h_k H_k,\end{aligned}$$

where  $\|H_k\|_{x_k} = \langle f''(x_k; \mu_t)H_k, H_k \rangle$  and  $f''(x_k; \mu_t)$  is computed by (3.86)..

Set

$$\begin{aligned}\gamma_{k+1} &= \frac{\|f'(x_{k+1}; \mu_t)\|^2}{-\langle f'(x_k; \mu_t), H_k \rangle}, \\ H_{k+1} &= -f'(x_k; \mu_t) + \gamma_{k+1}H_k.\end{aligned}$$

Increment  $k$  and repeat until  $\lambda_k < 0.05$ .

Set  $x_0 = x_k$  and  $\mu_{t+1} = \theta\mu_t$  where  $\theta$  is a constant satisfying  $0 < \theta < 1$ . Then increment  $t$  and repeat until  $\mu < \epsilon_2$  where  $\epsilon_2 > 0$  is a predefined tolerance.

We carried out the damped gradient and conjugate gradient methods on a SOCP problem. In addition, the performance of these two methods is compared with the damped Newton method. In particular, we take  $m = 600$ ,  $n = 600$ ,  $N = 3$ .

Figure 3.6 illustrates the results of implementing the inner loop using the damped gradient, conjugate gradient and Newton methods with  $\mu = 1$ .

Figure 3.7 illustrates the result of implementing the outer loop with  $\mu$  from 1 to 0.

Table 3.4 shows the simulation time and accuracy using the damped gradient, conjugate gradient and Newton methods with  $\mu = 1$ .

Simulation results show convergence of our algorithms. In comparison with the damped Newton method, the damped conjugate method cost less time and has super-linear convergence rate.

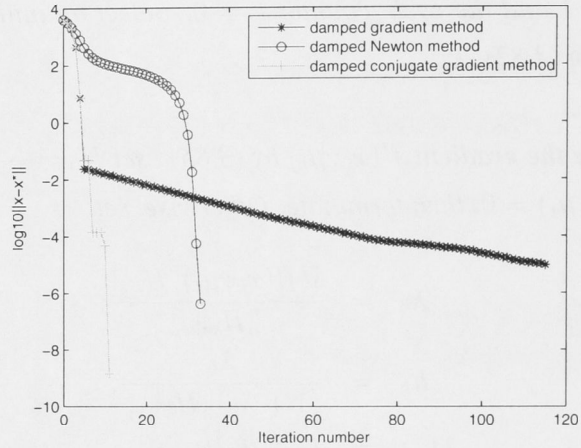


Figure 3.6: Error vs. iteration number for SOCP from the damped Newton method, damped gradient method and damped conjugate gradient method

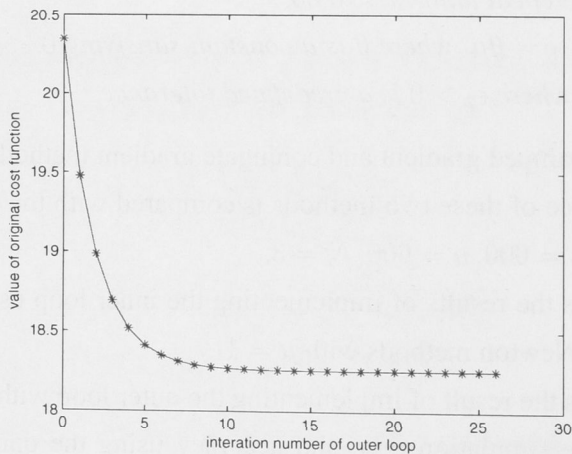


Figure 3.7: The value of the original cost function with  $\mu$  approaching 0 for SOCP

algorithm	time(second)	accuracy
damped gradient method	5.8750	$10^{-6}$
damped conjugate gradient method	0.5470	$10^{-6}$
damped Newton method	1.1520	$10^{-6}$

Table 3.4: Simulation time and accuracy

## 3.6 Conclusion

In this chapter, we present a damped gradient method and a damped conjugate gradient method for the numerical optimization of self-concordant functions. These two methods have been shown to make the cost function strictly decrease in each step and also have the sequence inside the domain of the cost function. Simulation results indicate that these two methods perform very well and converge to the minimums of self-concordant functions.



**Part II**

**Self-Concordant Functions On  
Riemannian Manifolds**

# Chapter 4

## Self-concordant Functions on Riemannian Manifolds

### 4.1 Introduction

**Background** Recall from Chapter 1 that self-concordant functions play an important role in developing interior point algorithms for solving certain convex constrained optimization problems including linear programming. It is therefore natural to attempt to extend the definition of self-concordance to functions on Riemannian manifolds, and then exploit this definition to derive novel optimization algorithms on Riemannian manifolds. In fact, the self-concordant concept has been extended to Riemannian manifolds in [71]. In that work, the author considered the convex programming problem

$$\begin{aligned} \min \quad & f_0(p) \\ \text{s.t.} \quad & f_i(p) \leq 0, \quad i = 1, \dots, m; \quad p \in M \end{aligned} \tag{4.1}$$

where  $M$  is a complete  $n$ -dimensional Riemannian manifold and developed a logarithmic barrier interior point method for solving it. Recall that in the Euclidean space, one approach for solving (4.1) is the barrier interior point method which uses the barrier function to enforce the constraint; this barrier function is chosen to be self-concordant. In order to extend this idea to Riemannian manifolds, it is necessary to extend the concept of self-concordant functions to Riemannian manifolds. To this end, the concept of a self-concordant function was defined on Riemannian manifolds and some of its properties were studied in [71]. Moreover, a Newton method with a

step-size choice rule was proposed to keep the iterates inside the constraint and guarantee the convergence.

We note that self-concordance appears to have not been defined precisely in [71]. For instance, the domain of a self-concordant function must be convex, but there are different types of convexity on a manifold, which was not considered in [71]. Furthermore, it was stated on p.352 that a self-concordant function goes to  $\infty$  on the boundary of its domain. However this does not appear to be true in general; perhaps the requirement that a self-concordant function is closed was accidentally omitted from the definition? There also appears to be a mistake on p.351; the reference lemma B. 2 in [37] is not true on a general manifold and hence its argument cannot be used. In addition, the properties of self-concordant functions were not extensively studied.

**Our work** In this chapter, we give a precise definition of a self-concordant function on a Riemannian manifold and derive properties of self-concordant functions which will be used to develop optimization algorithms; first a damped Newton method in this chapter, then a damped conjugate gradient method in Chapter 5. Convergence proofs of the damped Newton method are also given.

**Chapter outline** This chapter is organized as follows: Concepts of Riemannian manifolds are listed in Section 4.2. Section 4.3 defines self-concordant functions on manifolds. This definition is chosen to preserve as many nice properties of its original version in Euclidean space as possible. To facilitate the derivation and analysis of the proposed damped Newton method in Section 4.5, the Newton decrement is defined and analyzed in Section 4.4. It is shown that the damped Newton method has similar convergence properties to the algorithm for self-concordant functions in Euclidean space proposed in [57].

## 4.2 Concepts of Riemannian Manifolds

In this section, some fundamental concepts from differential geometry are introduced. However, we do not intend to present self-contained and complete exposure, and most of the proofs are omitted. See [30] for more details.

Let an  $n$ -dimensional smooth manifold be denoted as  $M$  which is an embedded manifold in  $\mathbb{R}^N$ . The differential structure of  $M$  is a set of local charts covering  $M$ . Each local chart is a pair of a neighborhood and a smooth mapping from this neighborhood to an open set in Euclidean space. The tangent space of  $M$  at a point  $p$  can be denoted as  $T_p M$ . It is the set of linear mappings from all smooth functions passing through the point  $p$  to real numbers, satisfying

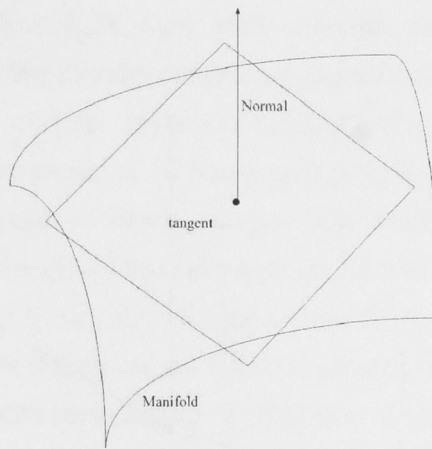


Figure 4.1: The tangent and normal spaces

the derivative condition. For  $n$ -dimensional manifolds, the tangent space at every point is an  $n$ -dimensional vector space with origin at this point of tangency. The normal space is the orthogonal complement of the tangent space in the ambient space. Figure 4.1 shows the tangent space and normal space at a point on a manifold.

A smooth manifold  $M$  is called Riemannian manifold if it is endowed with a metric structure. In Euclidean space, a vector can be moved parallel to itself by just moving the base of the arrow. For the manifold if a tangent vector is moved to another point on the manifold parallel to itself in its ambient space, it is generally not a tangent vector to the new point. For example, see Figure 4.2. However, we can transport tangent vectors along paths on the manifold by infinitesimally removing the component of the transported vector in the normal space. Figure 4.3 describes the following idea. Assume that we want to move a tangent vector  $\Delta$  along the curve  $\gamma(t)$  on the manifold. Then in every infinitesimal step, we first move  $\Delta$  parallel to itself in the ambient Euclidean space and then remove the normal component.

Let  $M$  denote a smooth  $n$ -dimensional geodesically complete Riemannian manifold. Recall that  $C^k$  smooth means derivatives of the order  $k$  exist and are continuous. For convenience, by smooth, we mean  $C^\infty$ , that is, derivatives of all orders exist. Let  $T_p M$  denote the tangent space at the point  $p \in M$ . Since  $M$  is a Riemannian manifold, it comes with an inner product  $\langle \cdot, \cdot \rangle_p$  on  $T_p M$  for each  $p \in M$ . This induces the norm  $\| \cdot \|_p$  given by  $\|X\|_p = \langle X, X \rangle_p^{\frac{1}{2}}$  for  $X \in T_p M$ .

There is a natural way (precisely, the Levi-Civita connection) of defining acceleration on a Riemannian manifold which is consistent with the metric structure. A curve with zero acceleration at every point is called a geodesic. Since  $M$  is geodesically complete, given a point

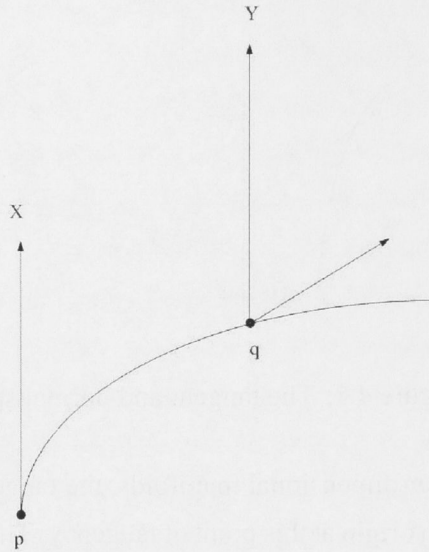


Figure 4.2: Move a tangent vector parallel to itself to another point on the manifold

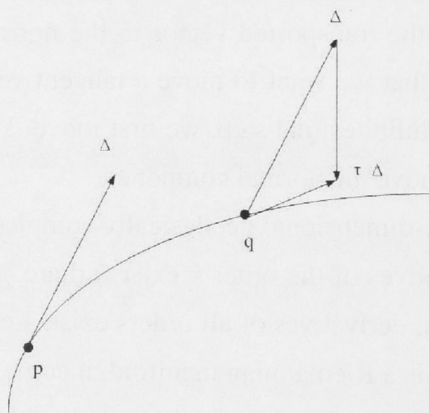


Figure 4.3: Parallel transport (infinitesimal space)

$p \in M$  and a tangent vector  $X \in T_pM$ , there exists a unique geodesic  $\gamma_X : \mathbb{R} \rightarrow M$  such that  $\gamma_X(0) = p$  and  $\dot{\gamma}_X(0) = X$ . We therefore define an exponential map  $\text{Exp}_p : T_pM \rightarrow M$  by  $\text{Exp}_p(X) = \gamma_X(1)$  for all  $X \in T_pM$ . Note that  $\text{Exp}_p tX$  is the geodesic emanating from  $p$  in the direction  $X$ . Another consequence of  $M$  being geodesically complete is that any two points on  $M$  can be joined by a geodesic of shortest length. The distance  $d(p, q)$  between two points  $p, q \in M$  is defined to be the length of this minimizing geodesic. Since the length of the curve  $\gamma : [0, 1] \rightarrow M$ ,  $\gamma(t) = \text{Exp}_p tX$ , is  $\|X\|_p$ , it follows that if  $q = \text{Exp}_p X$  then  $d(p, q) \leq \|X\|_p$ , where the inequality is possible if there exists a shorter geodesic connecting  $p$  and  $q$ .

If  $\gamma : [0, 1] \rightarrow M$  is a smooth curve from  $p = \gamma(0)$  to  $q = \gamma(1)$ , there is an associated linear isomorphism  $\tau_{pq} : T_pM \rightarrow T_qM$  called parallel transport. One of its properties is that lengths of vectors and angles between vectors are preserved, i.e.  $\forall X, Y \in T_pM$ ,  $\langle \tau_{pq}X, \tau_{pq}Y \rangle_q = \langle X, Y \rangle_p$ . For a point  $p \in M$  and a tangent vector  $X \in T_pM$ , we use  $\tau_{p\text{Exp}_p(tX)}$  to denote the parallel transport from the point  $p$  to the point  $\text{Exp}_p tX$  along the geodesic emanating from  $p$  in the direction  $X$ .

Let  $N$  be an open subset of  $M$ . Consider the function  $f : N \rightarrow \mathbb{R}$ . Given  $p \in N$  and  $X \in T_pN$ , the first, second and third covariant derivatives of  $f$  are defined as follows:

$$\nabla_X f(p) = \left. \frac{d}{dt} \right|_{t=0} \{f(\text{Exp}_p tX)\}, \tag{4.2}$$

$$\nabla_X^2 f(p) = \left. \frac{d^2}{dt^2} \right|_{t=0} \{f(\text{Exp}_p tX)\}, \tag{4.3}$$

$$\nabla_X^3 f(p) = \left. \frac{d^3}{dt^3} \right|_{t=0} \{f(\text{Exp}_p tX)\}. \tag{4.4}$$

The gradient of  $f$  at  $p \in N$ , denoted by  $\text{grad}_p f$ , is defined as the unique tangent vector in  $T_pN$  such that  $\nabla_X f(p) = \langle \text{grad}_p f, X \rangle$  for all  $X \in T_pN$ .

The Hessian of  $f$  at  $p \in N$  is the unique symmetric bilinear form  $\text{Hess}_p f$  defined by the property

$$\text{Hess}_p f(X, X) = \nabla_X^2 f(p), \quad X \in T_pN. \tag{4.5}$$

Note that (4.5) fully defines  $\text{Hess}_p f$  since

$$\text{Hess}_p f(X, Y) = \frac{\text{Hess}_p f(X + Y, X + Y) - \text{Hess}_p f(X, X) - \text{Hess}_p f(Y, Y)}{2} \tag{4.6}$$

for  $X, Y \in T_p N$ .

### 4.3 Self-Concordant Functions

Referring to Definition 1 in chapter 1, extending the definition of self-concordance to Riemannian manifolds requires carefully defining the convex set. Intuitively, the convex set on Riemannian manifolds can be determined by the geodesics connecting two points. However, there could be more than one geodesic connecting two points on Riemannian manifolds. For instance, for any two different points on the sphere, there exist two geodesics joining them. Therefore, there is no single best definition of convexity of selected subset [70]. The definition of convexity used extensively in [70] is concerned with all geodesics of the whole Riemannian manifolds connecting two points. On the other hand, this definition limits the definition of convex functions since in most cases, the cost functions defined on Riemannian manifolds are locally convex. To be more general, our definition goes as follows. We say a subset  $N$  of  $M$  is convex if for any  $p, q \in N$ , out of all the geodesics connecting  $p$  and  $q$ , there is precisely one which is contained in  $N$ . Note that this is a weaker condition than that used extensively in [70]. Then, a function  $f : N \subset M \rightarrow \mathbb{R}$  is said to be convex if  $N$  is a convex set and for any geodesic  $\gamma : [0, 1] \rightarrow N$ , the function  $f \circ \gamma : [0, 1] \rightarrow \mathbb{R}$  satisfies the usual definition of convexity, namely

$$f(\gamma(t)) \leq (1-t)f(\gamma(0)) + tf(\gamma(1)), \quad t \in [0, 1]. \quad (4.7)$$

If  $f : N \rightarrow \mathbb{R}$  is  $C^\infty$ -smooth and  $N$  is convex, then  $f$  is convex if and only if  $\nabla_X^2 f(p) \geq 0$  for all  $p \in N$  and  $X \in T_p N$ .

The epigraph  $\text{epi}(f)$  of  $f$  is defined by

$$\text{epi}(f) = \{(p, t) \in N \times \mathbb{R} \mid f(p) \leq t\}. \quad (4.8)$$

A function  $f$  is said to be closed convex if its epigraph  $\text{epi}(f)$  is both convex and a closed subset of  $M \times \mathbb{R}$ . The convexity of  $\text{epi}(f)$  is explained as follows. Let  $(p, t), (q, s) \in \text{epi}(f)$  and  $\text{Exp}_p \theta X_p$  be the geodesic connecting  $p$  and  $q$  where  $q = \text{Exp}_p X_p$ ,  $X_p \in T_p N$  and  $\theta \in [0, 1]$ . Then  $\text{epi}(f)$  is said to be convex if for all  $\theta \in [0, 1]$ ,

1.  $\text{Exp}_p \theta X_p$  is the unique geodesic connecting  $p$  and  $q$  such that  $\text{Exp}_p \theta X_p \in N$ ;
2.  $(\text{Exp}_p \theta X_p, t + \theta(s - t)) \in \text{epi}(f)$ .

It is also easy to see from the above definition that  $\text{epi}(f)$  being convex implies  $f$  being convex over  $N$ .

Consequently, our definition of self-concordance given below, differs from the Euclidean definition in 1 in chapter 1.

**Definition 2.** *Let  $M$  be a smooth  $n$ -dimensional geodesically complete Riemannian manifold. Let  $f : N \subset M \rightarrow \mathbb{R}$  be a  $C^3$ -smooth closed function. Then  $f$  is self-concordant if*

1.  $N$  is an open convex subset of  $M$ ;
2.  $f$  is convex on  $N$ ;
3. there exists a constant  $M_f > 0$  such that the inequality

$$|\nabla_X^3(f(p))| \leq M_f(\nabla_X^2 f(p))^{3/2} \tag{4.9}$$

holds for all  $p \in N$  and  $X \in T_p N$ .

The reason why  $f$  is required to be closed in Definition 2 is to ensure that  $f$  behaves nicely on the boundary of  $N$ ; this is shown in the following proposition.

**Proposition 4.3.1.** *Let  $f : N \rightarrow \mathbb{R}$  be self-concordant. Let  $\partial(N)$  denote the boundary of  $N$ . Then for any  $\bar{p} \in \partial(N)$  and any sequence of points  $p_k \in N$  converging to  $\bar{p}$  we have  $f(p_k) \rightarrow \infty$ .*

**Proof:** The proof is a straightforward generalization of the proof of Theorem 4.1.4 in [57]. For  $k = 2, 3, \dots$ , define  $X_k \in T_{p_1} N$  to be such that  $p_k := \text{Exp}_{p_1} X_k$  and  $p_k \in N$ . Since  $f$  is convex, in view of (4.7), we have

$$f(\text{Exp}_{p_1} tX_k) \leq (1 - t)f(p_1) + tf(p_k) \tag{4.10}$$

where  $0 \leq t \leq 1$ .

It follows from (4.10) that if  $0 < t \leq 1$  then

$$f(p_1) + \frac{f(\text{Exp}_{p_1} tX_k) - f(p_1)}{t} \leq f(p_k). \tag{4.11}$$



As  $t \rightarrow 0$ , from (4.11) we have

$$\begin{aligned} f(p_1) + \lim_{t \rightarrow 0} \frac{f(\text{Exp}_{p_1} t X_k) - f(p_1)}{t} &= f(p_1) + \nabla_{X_k} f(p_1) \\ &= f(p_1) + \langle \text{grad}_{p_1} f, X_k \rangle \\ &\leq f(p_k). \end{aligned} \tag{4.12}$$

Therefore the sequence  $\{f(p_k)\}$  is bounded below by

$$f(p_k) \geq f(p_1) + \langle \text{grad}_{p_1} f, X_k \rangle \tag{4.13}$$

where we recall that  $X_k \in T_{p_1} N$  is such that  $p_k := \text{Exp}_{p_1} X_k$ .

Assume to the contrary that the sequence  $\{f(p_k), k \geq 1\}$  is bounded from above. Then it has a limit point  $\bar{f}$ . By considering a subsequence if necessary, we can regard it as a unique limit point of the sequence. Let  $z_k = (p_k, f(p_k))$ . Then we have

$$z_k = (p_k, f(p_k)) \rightarrow \bar{z} = (\bar{p}, \bar{f}). \tag{4.14}$$

By definition,  $z_k \in \text{epi}(f)$ . However, we have  $\bar{z} \notin \text{epi}(f)$  since  $\bar{p} \notin N$ . That is a contradiction since  $f$  is closed. □

**Proposition 4.3.2.** *Let  $f_i : N \subset M \rightarrow R$  be self-concordant with constants  $M_{f_i}$ ,  $i = 1, 2$  and let  $\alpha, \beta > 0$ . Then the function  $f(x) = \alpha f_1(x) + \beta f_2(x)$  is self-concordant with the constant*

$$M_f = \max \left\{ \frac{1}{\sqrt{\alpha}} M_{f_1}, \frac{1}{\sqrt{\beta}} M_{f_2} \right\}. \tag{4.15}$$

**Proof:** This proof is similar to the proof of Theorem 4.1.1 in [57]. Since  $f_i$ ,  $i = 1, 2$  are closed convex on  $N$ ,  $f$  is closed convex on  $N$ , which can be easily proved in view of the proof of Theorem 3.1.5 in [57]. Moreover, for any fixed  $p \in N$  and  $X \in T_p N$ , we have

$$|\nabla_X^3 f_i(p)| \leq M_{f_i} (\nabla_X^2 f_i(p))^{\frac{3}{2}}, \quad i = 1, 2. \tag{4.16}$$

Now, consider two cases.

Case One:  $\alpha \nabla_X^2 f_1(p) + \beta \nabla_X^2 f_2(p) = 0$ .

Since  $f_1$  and  $f_2$  are both self-concordant, we have

$$\nabla_X^2 f_1(p) \geq 0, \quad (4.17)$$

$$\nabla_X^2 f_2(p) \geq 0. \quad (4.18)$$

Therefore from the assumption, we obtain

$$\nabla_X^2 f_1(p) = 0, \quad (4.19)$$

$$\nabla_X^2 f_2(p) = 0. \quad (4.20)$$

By the definition of self-concordance, it follows from (4.19) and (4.20) that

$$\nabla_X^3 f_1(p) = 0, \quad (4.21)$$

$$\nabla_X^3 f_2(p) = 0. \quad (4.22)$$

Hence, it follows that

$$|\nabla_X^3 f(p)| \leq M_f (\nabla_X^2 f(p))^{\frac{3}{2}} \quad (4.23)$$

where  $M_f = \max \left\{ \frac{1}{\sqrt{\alpha}} M_{f_1}, \frac{1}{\sqrt{\beta}} M_{f_2} \right\}$ .

Case Two:  $\alpha \nabla_X^2 f_1(p) + \beta \nabla_X^2 f_2(p) \neq 0$ .

Denote  $\omega_i = \nabla_X^2 f_i(p)$ . Since  $\omega_i > 0$ ,  $i = 1, 2$  by the assumption, we have

$$\begin{aligned} \frac{|\nabla_X^3 f(p)|}{(\nabla_X^2 f(p))^{\frac{3}{2}}} &\leq \frac{|\alpha \nabla_X^3 f_1(p)| + |\beta \nabla_X^3 f_2(p)|}{[\alpha \nabla_X^2 f_1(p) + \beta \nabla_X^2 f_2(p)]^{\frac{3}{2}}} \\ &\leq \frac{\alpha M_{f_1} \omega_1^{\frac{3}{2}} + \beta M_{f_2} \omega_2^{\frac{3}{2}}}{[\alpha \omega_1 + \beta \omega_2]^{\frac{3}{2}}}. \end{aligned} \quad (4.24)$$

Note that the last inequality is not changing when we replace  $(\omega_1, \omega_2)$  by  $(t\omega_1, t\omega_2)$  with  $t > 0$ . Consequently, we can assume that  $\alpha\omega_1 + \beta\omega_2 = 1$ . Let  $\xi = \alpha\omega_1$ . Then the right hand side of (4.24) becomes

$$\frac{M_{f_1}}{\sqrt{\alpha}} \xi^{\frac{3}{2}} + \frac{M_{f_2}}{\sqrt{\beta}} (1 - \xi)^{\frac{3}{2}}. \quad (4.25)$$

Now, consider (4.25) as a function in  $\xi \in [0, 1]$ .

This function is convex in  $\xi$  [57]. As a result, its maximum is either  $\xi = 0$  or  $\xi = 1$ .

This completes the proof.  $\square$

If a function  $f$  is self-concordant with the constant  $M_f$ , then the function  $M_f^{-2}f$  is self-concordant with the constant 1 as can be directly checked by a simple computation. As such, we assume  $M_f = 2$  for the rest of this chapter. Such functions are called **standard** self-concordant.

Consider a function defined on  $N$ :

$$f : N \subset M \rightarrow \mathbb{R}. \quad (4.26)$$

For the simplicity of the analysis in this chapter, we assume that  $f$  in (4.26) additionally satisfies

**Assumption 2.**

$$\nabla_X^2 f(p) > 0, \quad \forall p \in N, X \in T_p N.$$

Then, the second order covariant differentials can be used to define a Dikin-type ellipsoid  $W(p; r) \subset T_p N$  – for any  $p \in N$ , and  $r > 0$ ,

$$W(p; r) := \{X_p \in T_p N \mid [\nabla_{X_p}^2 f(p)]^{1/2} < r\}. \quad (4.27)$$

Mapping all the elements in  $W(p; r)$  by the exponential map  $\text{Exp}_p$  yields a subset  $Q(p; r)$  of  $M$  where

$$Q(p; r) = \{q \in M \mid q = \text{Exp}_p X_p, X_p \in W(p; r)\}. \quad (4.28)$$

A self-concordant function also has the following interesting property:

**Proposition 4.3.3.**  $\forall p \in N, Q(p; 1) \subseteq N$ .

This property gives a safe bound for the line search along geodesics for optimization problems so that the search will always be in the admissible domain. We need the following lemma to prove it.

**Lemma 6.** *Let  $f : N \rightarrow \mathbb{R}$  in (4.26) be a standard self-concordant function satisfying Assumption 2. For a point  $p \in N$  and a non-zero tangent vector  $X \in T_p N$ , recall the definitions of*

$\text{Exp}_p tX$  and  $\tau_{p\text{Exp}_p(tX)}$  in Section 4.2. Let  $U = \{t \in \mathbb{R} | \text{Exp}_p tX \in N\}$ . Define a function  $\phi : U \rightarrow \mathbb{R}$  as follows

$$\phi(t) := [\nabla_{\tau_{p\text{Exp}_p(tX)}^2}^2 f(\text{Exp}_p tX)]^{-1/2}. \quad (4.29)$$

Then, the following results hold:

1.  $|\phi'(t)| \leq 1$ ;
2. If  $\phi(0) > 0$ , then,  $(-\phi(0), \phi(0)) \subseteq U$ .

**Proof:**

1. It can be calculated that

$$\begin{aligned} \phi'(t) &= -\frac{\frac{d}{dt}[\nabla_{\tau_{p\text{Exp}_p(tX)}^2}^2 f(\text{Exp}_p tX)]}{2[\nabla_{\tau_{p\text{Exp}_p(tX)}^2}^2 f(\text{Exp}_p tX)]^{3/2}} \\ &= -\frac{\nabla_{\tau_{p\text{Exp}_p(tX)}^3}^3 f(\text{Exp}_p tX)}{2[\nabla_{\tau_{p\text{Exp}_p(tX)}^2}^2 f(\text{Exp}_p tX)]^{3/2}}. \end{aligned}$$

The claim 1 follows directly from the definition of self-concordant function.

2. In view of Proposition 4.3.1, we have  $f(\text{Exp}_p tX)$  goes to  $\infty$  as  $\text{Exp}_p tX$  approaches the boundary of  $N$ . It implies that the function  $\nabla_{\tau_{p\text{Exp}_p(tX)}^2}^2 f(\text{Exp}_p tX)$  cannot be bounded. Therefore, we have

$$\phi(t) \rightarrow \infty \quad \text{as} \quad \text{Exp}_p tX \rightarrow \partial(N). \quad (4.30)$$

Since the function  $f$  satisfies Assumption 2, by (4.30), we obtain

$$U \equiv \{t | \phi(t) > 0\}. \quad (4.31)$$

By the claim 1, we have

$$\phi(t) \geq \phi(0) - |t|. \quad (4.32)$$

Combining (4.31) and (4.32), it follows that

$$(-\phi(0), \phi(0)) \subseteq U. \quad (4.33)$$

In the following, two groups of properties will be given to reveal the relationship between two different points on a geodesic. They are delicate characteristics of self-concordant functions. In fact, they are the foundation for the polynomial complexity of self-concordant functions.

**Proposition 4.3.4.** *For any  $p \in N$  and  $X_p \in T_p N$ , such that for  $t \in [0, 1]$  the geodesic  $\text{Exp}_p t X_p$  is contained in  $N$ . Let  $q = \text{Exp}_p X_p$ . If  $f : N \rightarrow \mathbb{R}$  in (4.26) is a self-concordant function, the following results hold:*

$$[\nabla_{\tau_{pq} X_p}^2 f(q)]^{1/2} \geq \frac{[\nabla_{X_p}^2 f(p)]^{1/2}}{1 + [\nabla_{X_p}^2 f(p)]^{1/2}}, \quad (4.34)$$

$$\nabla_{\tau_{pq} X_p} f(q) - \nabla_{X_p} f(p) \geq \frac{\nabla_{X_p}^2 f(p)}{1 + [\nabla_{X_p}^2 f(p)]^{1/2}}, \quad (4.35)$$

$$f(q) \geq f(p) + \nabla_{X_p} f(p) + [\nabla_{X_p}^2 f(p)]^{1/2} - \ln(1 + [\nabla_{X_p}^2 f(p)]^{1/2}), \quad (4.36)$$

where  $\tau_{pq}$  is the parallel transport from  $p$  to  $q$  along the geodesic  $\text{Exp}_p t X_p$ .

*Proof.* Let  $\phi(t)$  be the same function defined in Lemma 6, where one can see that  $\phi(1) \leq \phi(0) + 1$ . This is equivalent to (4.34) taking into account that  $\phi(0) = [\nabla_{X_p}^2 f(p)]^{-1/2}$ , and  $\phi(1) = [\nabla_{\tau_{pq} X_p}^2 f(q)]^{-1/2}$ . Furthermore,

$$\begin{aligned} & \nabla_{\tau_{pq} X_p} f(q) - \nabla_{X_p} f(p) \\ &= \int_0^1 \nabla_{\tau_{\text{Exp}_p t X_p} X_p}^2 f(\text{Exp}_p t X_p) dt, \\ &= \int_0^1 \frac{1}{t^2} \nabla_{t \tau_{\text{Exp}_p t X_p} X_p}^2 f(\text{Exp}_p t X_p) dt \end{aligned} \quad (4.37)$$

which leads to (4.35) using the inequality (4.34).

For the inequality (4.36), notice that:

$$\begin{aligned}
 & f(q) - f(p) - \nabla_{X_p} f(p) \\
 &= \int_0^1 (\nabla_{\tau_p \text{Exp}_p t X_p} f(\text{Exp}_p t X_p) - \nabla_{X_p} f(p)) dt \\
 &= \int_0^1 \frac{1}{t} [(\nabla_{t \tau_p \text{Exp}_p t X_p} f(\text{Exp}_p t X_p) - \nabla_{t X_p} f(p))] dt \\
 &\geq \int_0^1 \frac{\nabla_{t X_p}^2 f(p)}{t(1 + [\nabla_{t X_p}^2 f(p)]^{1/2})} dt.
 \end{aligned}$$

Let  $r = [\nabla_{X_p}^2 f(p)]^{1/2}$ . The last integral becomes

$$\int_0^1 \frac{tr^2}{1 + tr} dt = r - \ln(1 + r),$$

which leads to the inequality (4.36) by replacing  $r$  with its original form.  $\square$

**Proposition 4.3.5.** *For any  $p \in N$  and  $X_p \in W(p; 1)$ , let  $q = \text{Exp}_p X_p$ . If  $f : N \rightarrow \mathbb{R}$  in (4.26) is a self-concordant function, then there holds:*

$$(1 - [\nabla_{X_p}^2 f(p)]^{1/2})^2 \nabla_{X_p}^2 f(p) \leq \nabla_{\tau_{pq} X_p}^2 f(q) \leq \frac{\nabla_{X_p}^2 f(p)}{(1 - [\nabla_{X_p}^2 f(p)]^{1/2})^2}, \quad (4.38)$$

$$\nabla_{\tau_{pq} X_p} f(q) - \nabla_{X_p} f(p) \leq \frac{\nabla_{X_p}^2 f(p)}{1 - [\nabla_{X_p}^2 f(p)]^{1/2}}, \quad (4.39)$$

$$f(q) \leq f(p) + \nabla_{X_p} f(p) - [\nabla_{X_p}^2 f(p)]^{1/2} - \ln(1 - [\nabla_{X_p}^2 f(p)]^{1/2}), \quad (4.40)$$

where  $\tau_{pq}$  is the parallel transport from  $p$  to  $q$  along the geodesic  $\text{Exp}_p t X_p$ .

*Proof.* Let  $\psi(t)$  be a function defined in the following form:

$$\psi(t) = \frac{d^2}{dt^2} f(\text{Exp}_p t X_p). \quad (4.41)$$

where  $t \in [0, 1]$ .

Since  $X_p \in W(p; 1)$ , we have  $\text{Exp}_p t X_p \in N$  for all  $t \in [0, 1]$ .

Taking the first order derivative of  $\psi$ , we obtain

$$\begin{aligned}
 |\psi'(t)| &= \left| \frac{d^3}{dt^3} f(\text{Exp}_p tX_p) \right| \\
 &= |\nabla_{\tau_p \text{Exp}_p(tX_p) X_p}^3 f(\text{Exp}_p(tX_p))| \\
 &\leq 2(\nabla_{\tau_p \text{Exp}_p(tX_p)}^2 f(\text{Exp}_p(tX_p)))^{\frac{1}{2}} (\nabla_{\tau_p \text{Exp}_p(tX_p)}^2 f(\text{Exp}_p(tX_p))) \\
 &= 2(\nabla_{\tau_p \text{Exp}_p(tX_p)}^2 f(\text{Exp}_p(tX_p)))^{\frac{1}{2}} \psi(t) \\
 &= \frac{2}{t} (\nabla_{t\tau_p \text{Exp}_p(tX_p)}^2 f(\text{Exp}_p(tX_p)))^{\frac{1}{2}} \psi(t) \\
 &\leq \frac{2}{t} \frac{t[\nabla_{X_p}^2 f(p)]^{\frac{1}{2}}}{1 - t[\nabla_{X_p}^2 f(p)]^{\frac{1}{2}}} \psi(t).
 \end{aligned} \tag{4.42}$$

Here the last part is obtained by applying  $\phi(1) \geq \phi(0) - 1$  from Lemma 6.

Integrating both sides of the inequality (4.42) from 0 to 1, we have

$$(1 - [\nabla_{X_p} f(p)]^{\frac{1}{2}})^2 \leq \frac{\psi(1)}{\psi(0)} \leq \frac{1}{(1 - [\nabla_{X_p} f(p)]^{\frac{1}{2}})^2} \tag{4.43}$$

which is equivalent to the inequality (4.38).

Combining the inequality (4.38) and the formula (4.37), one obtains

$$\begin{aligned}
 \nabla_{\tau_p q X_p} f(q) - \nabla_{X_p} f(p) &\leq \int_0^1 \frac{1}{t^2} \frac{\nabla_{tX_p}^2 f(p)}{(1 - [\nabla_{tX_p}^2 f(p)]^{1/2})^2} dt \\
 &= \frac{\nabla_{X_p}^2 f(p)}{1 - [\nabla_{X_p}^2 f(p)]^{1/2}},
 \end{aligned}$$

which proves the inequality (4.39).

Combining this result and using the same technique as that used in the proof of the last

property, there holds:

$$\begin{aligned}
 & f(q) - f(p) - \nabla_{X_p} f(p) \\
 = & \int_0^1 \nabla_{\tau_p \text{Exp}_p t X_p} f(\text{Exp}_p t X_p) dt - \nabla_{X_p} f(p) \\
 = & \int_0^1 \left\{ \frac{1}{t} [\nabla_{t \tau_p \text{Exp}_p t X_p} f(\text{Exp}_p t X_p)] - \nabla_{X_p} f(p) \right\} dt \\
 \leq & \int_0^1 \frac{\nabla_{t X_p}^2 f(p)}{t(1 - [\nabla_{t X_p}^2 f(p)]^{1/2})} dt \\
 = & -[\nabla_{X_p}^2 f(p)]^{1/2} - \ln(1 - [\nabla_{X_p}^2 f(p)]^{1/2}).
 \end{aligned}$$

As such, the inequality (4.40) is obtained by a simple transformation of this inequality.  $\square$

**Proposition 4.3.6.** *Let  $f : N \rightarrow \mathbb{R}$  in (4.26) be a self-concordant function. For any  $p \in N$ , and  $X_p \in T_p N$ , if  $r = \sqrt{\nabla_{X_p}^2 f(p)} < 1$ , there holds:*

$$\begin{aligned}
 & (1 - r + \frac{r^2}{3}) \nabla_{X_p}^2 f(p) \\
 \leq & \int_0^1 \nabla_{\tau_p \text{Exp}_p t X_p}^2 f(\text{Exp}_p t X_p) dt \\
 \leq & \frac{\nabla_{X_p}^2 f(p)}{1 - r}.
 \end{aligned} \tag{4.44}$$

**Proof:** In view of the right inequality of (4.38), we have

$$\begin{aligned}
 & \int_0^1 \nabla_{\tau_p \text{Exp}_p t X_p}^2 f(\text{Exp}_p t X_p) dt \\
 = & \int_0^1 \frac{1}{t^2} \nabla_{t \tau_p \text{Exp}_p t X_p}^2 f(\text{Exp}_p t X_p) dt \\
 \leq & \int_0^1 \frac{1}{t^2} \frac{\nabla_{t X_p}^2 f(p)}{(1 - \sqrt{\nabla_{t X_p}^2 f(p)})^2} dt \\
 = & \int_0^1 \frac{\nabla_{X_p}^2 f(p)}{(1 - t \sqrt{\nabla_{X_p}^2 f(p)})^2} dt \\
 = & \frac{\nabla_{X_p}^2 f(p)}{1 - r}.
 \end{aligned} \tag{4.45}$$



Similarly, in view of the left inequality of (4.38), we have

$$\begin{aligned} & \left(1 - r + \frac{r^2}{3}\right) \nabla_{X_p}^2 f(p) \\ & \leq \int_0^1 \nabla_{\tau \text{Exp}_p t X_p}^2 f(\text{Exp}_p t X_p) dt. \end{aligned} \tag{4.46}$$

□

## 4.4 Newton Decrement

Consider the following auxiliary quadratic cost defined on  $T_p M$

$$N_{f,p}(X) := f(p) + \nabla_X f(p) + \frac{1}{2} \nabla_X^2 f(p). \tag{4.47}$$

**Definition 3.** Let  $f : N \rightarrow \mathbb{R}$  in (4.26) be a self-concordant function. The Newton decrement  $X_N(f, p)$  is defined as the minimal solution to the auxiliary cost function given by (4.47). More specifically,

$$X_N(f, p) := \arg \min_{X \in T_p M} N_{f,p}(X). \tag{4.48}$$

Similar to the case in Euclidean space, the Newton decrement can be characterized in many ways. The following theorem summarizes its properties.

**Theorem 4.4.1.** Let  $f : N \rightarrow \mathbb{R}$  in (4.26) be a self-concordant function,  $p$ , a given point in  $N \subseteq M$ , and  $X_N$ , its Newton decrement defined at  $p$ . The following results hold:

$$\text{Hess}_p(X_N, X) = -\nabla_X f(p), \quad \forall X \in T_p M, \tag{4.49}$$

$$\sqrt{\nabla_{X_N}^2 f(p)} = \max\{\nabla_X f(p) | X \in T_p M, \nabla_X^2 f(p) \leq 1\}. \tag{4.50}$$

*Proof.* Since  $p$  is a given point on the manifold  $M$ , the claimed results can be converted into their local representation in Euclidean space. More specifically, consider the following quadratic function:

$$\begin{aligned} q(x) & := \frac{1}{2} x^\top A x + b^\top x + c, \\ \text{where } & A \in \mathbb{R}^n \times \mathbb{R}^n, A^\top = A, b \in \mathbb{R}^n, c \in \mathbb{R}. \end{aligned} \tag{4.51}$$

Let  $x^*$  denote the optimal point. Then, the gradient of  $q$  at  $x^*$  must be a zero vector. i.e.,

$$y^\top (Ax^* + b) = 0. \quad \forall y \in R^n.$$

This is the local representation of (4.49).

On the other hand,

$$\begin{aligned} |y^\top b| &= |y^\top Ax^*| = |y^\top A^{1/2} A^{1/2} x^*| \\ &\leq (y^\top Ay)^{1/2} ((x^*)^\top Ax^*)^{1/2}, \end{aligned}$$

where the equality holds if and only if  $y = x^*$ . As such,

$$\begin{aligned} \max\{y^\top b \mid y \in R^n, y^\top Ay \leq 1\} &= \max\left\{\frac{y^\top b}{\sqrt{y^\top Ay}} \mid y \in R^n\right\} \\ &= \sqrt{(x^*)^\top Ax^*}. \end{aligned}$$

This is the local representation of (4.50). Therefore, the proof is complete.  $\square$

## 4.5 A Damped Newton Algorithm for Self-Concordant Functions

Consider now the minimization problem of self-concordant functions on a smooth manifold. First, let us establish the existence of the minimal point:

**Theorem 4.5.1.** *Let  $f : N \rightarrow R$  in (4.26) be a self-concordant function. Let  $\lambda_f(p)$  be defined as follows:*

$$\lambda_f(p) := \max_{X \in T_p M} \frac{|\nabla_X f(p)|}{\sqrt{\nabla_X^2 f(p)}}, \quad \text{for } p \in N. \quad (4.52)$$

Then we have

1.  $\lambda_f(p) = \sqrt{\nabla_{X_N}^2 f(p)}$ ,
2. if  $\lambda_f(p) < 1$  for some  $p \in N$ , then there exists a unique point  $p_f^* \in N$  such that

$$f(p_f^*) = \min\{f(p) \mid p \in N\}.$$

*Proof.* 1. In view of (4.50), if we fix  $p$ , since  $\nabla_X f(p)$  is linear and  $\nabla_X^2 f(p)$  bilinear on  $X$  for any  $X \in T_p N$ , we have

$$\begin{aligned} \sqrt{\nabla_{X_N}^2 f(p)} &= \max\{\nabla_X f(p) | X \in T_p M, \nabla_X^2 f(p) \leq 1\} \\ &= \max_{X \in T_p M} \frac{|\nabla_X f(p)|}{\sqrt{\nabla_X^2 f(p)}}, \quad \text{for } p \in N \\ &= \lambda_f(p). \end{aligned} \tag{4.53}$$

2. Let  $p_0$  be a point such that  $\lambda_f(p_0) < 1$ . For any  $q \in N$  such that  $f(q) < f(p_0)$ , from (4.36) we have

$$\begin{aligned} f(q) &\geq f(p_0) - \lambda_f(p_0) [\nabla_{X_{p_0q}}^2 f(p_0)]^{1/2} \\ &\quad + [\nabla_{X_{p_0q}}^2 f(p_0)]^{1/2} - \ln(1 + [\nabla_{X_{p_0q}}^2 f(p_0)]^{1/2}). \end{aligned}$$

Hence,

$$\frac{\ln(1 + [\nabla_{X_{p_0q}}^2 f(p_0)]^{1/2})}{[\nabla_{X_{p_0q}}^2 f(p_0)]^{1/2}} \geq 1 - \lambda.$$

Since

$$\lim_{t \rightarrow +\infty} \frac{\ln(1+t)}{t} = 0,$$

there exists a constant  $c > 0$ , such that

$$[\nabla_{X_{p_0q}}^2 f(p_0)]^{1/2} \leq c. \tag{4.54}$$

Hence these  $X_{p_0q}$  contained in the compact set defined by inequality (4.54). Consider the map from the any tangent vector  $X$  to its geodesic  $\text{Exp}(X)$ . This map is continuous by the definition of geodesic. This indicates the image of compact set defined by inequality (4.54) is also compact. Therefore, a minimal point exists.

On the other hand, let  $p^*$  denote a minimal point. Then,

$$\begin{aligned} f(q) &\geq f(p^*) + [\nabla_{X_{p^*q}}^2 f(p^*)]^{1/2} - \ln(1 + [\nabla_{X_{p^*q}}^2 f(p^*)]^{1/2}) \\ &> f(p^*), \forall q \in N, q \neq p^*. \end{aligned} \tag{4.55}$$

The uniqueness is proved.

□

Consider optimization problems of the form

$$\min_{x \in N} f : N \subset M \rightarrow \mathbb{R}. \quad (4.56)$$

In general, it is hard to solve (4.56) since the domain of  $f$  is an open subset of a Riemannian manifold. However, if  $f$  has nice properties, there exist powerful techniques to solve (4.56). In this section, we concentrate on the special case of the optimization problem (4.56) when  $f$  in (4.56) satisfies the following assumption.

**Assumption 3.** *The function  $f$  in (4.56) is self-concordant, has a minimum and  $\nabla_X^2 f(p) > 0$ ,  $\forall p \in N, X \in T_p N$ . By scaling  $f$  if necessary, it is assumed without loss of generality that  $f$  satisfies (4.9) with  $M_f = 2$ .*

Assumption 3 guarantees that  $f$  has a unique minimum on  $N$ . Let  $K = \{p \in N \mid f(p) \leq f(p_0), p_0 \in N\}$  and be compact. Then Assumption 3 implies that there exist  $\alpha, \theta > 0$  such that

$$\theta \|X\|_p^2 \leq \nabla_X^2 f(p) \leq \alpha \|X\|_p^2 \quad \forall p \in K, X \in T_p N. \quad (4.57)$$

Consider the following damped Newton method for solving (4.56) when  $f$  in (4.56) satisfies Assumption 3.

**Algorithm 11. (Damped Newton Algorithm)**

step 0: Randomly generate an initial point  $p_0 \in N$  and compute  $\text{grad}_{p_0} f$ . Set  $k = 0$ .

step k: If  $\text{grad}_{p_k} f = 0$ , then terminate. Otherwise, compute the Newton decrement  $X_{Nk}$  by (4.49). Then compute

$$\lambda_k = \nabla_{X_{Nk}}^2 f(p_k), \quad (4.58)$$

$$t_k = \frac{1}{1 + \lambda_k}, \quad (4.59)$$

$$p_{k+1} = \text{Exp}_{p_k} t_k X_{Nk}, \quad (4.60)$$

where  $\text{Exp}_{p_{k-1}} t X_N$  is the exponential map of the Newton decrement at  $p_{k-1}$ .

The following theorem establishes the convergence properties of the proposed damped Newton algorithm.

**Theorem 4.5.2.** *Let the minimal point of  $f(p)$  be denoted as  $p^*$ , and  $p$  is any admissible point in  $Q(p^*; 1)$ .*

1. *The following inequality holds:*

$$[\nabla_{X_{pp^*}}^2 f(p)]^{1/2} \leq \frac{\lambda_f(p)}{1 - \lambda_f(p)} \quad (4.61)$$

where  $X_{pp^*} \in T_p N$  satisfying  $p^* = \text{Exp}_p X_{pp^*}$ .

2. *If  $\lambda_f(p) < 1$ , then*

$$0 \leq f(p) - f(p^*) \leq -\lambda_f(p) - \ln(1 - \lambda_f(p)). \quad (4.62)$$

3. *For the proposed Damped Newton Method algorithm, there holds:*

$$f(p^*) \leq f(p_k) \leq f(p_{k-1}) - \omega(\lambda_{k-1}), \quad (4.63)$$

where  $\omega(t) = t - \ln(1 + t)$ .

*Proof.* 1. Let  $[\nabla_{X_{pp^*}}^2 f(p)]^{1/2}$  be denoted as  $r(p)$ . In view of (4.35) and (4.39) we have:

$$\frac{r(p)^2}{1 - r(p)} \geq -\nabla_{X_{pp^*}} f(p) \geq \frac{r^2(p)}{1 + r(p)} \geq 0. \quad (4.64)$$

On the other hand, there holds

$$|\nabla_{X_{pp^*}} f(p)| \leq \lambda_f(p)r(p),$$

by the definition of  $\lambda_f(p)$ . Therefore,

$$\lambda_f(p) \geq \frac{r(p)}{1 + r(p)},$$

where  $r$  can be solved as follows:

$$r(p) \leq \frac{\lambda_f(p)}{1 - \lambda_f(p)},$$

which is (4.61).

2. Based on (4.36) and the inequality (4.61) obtained above, one has:

$$\begin{aligned} f(p^*) - f(p) &\geq \nabla_{X_{pp^*}} f(p) + r(p) - \ln(1 + r(p)) \\ &\geq r(p) - \ln(1 + r(p)) - \lambda_f(p)r(p). \end{aligned} \quad (4.65)$$

Let an auxiliary function  $g(x, y)$  be defined as:

$$\begin{aligned} g(x, y) &= x - \ln(1 + x) - xy + y - \ln(1 - y), \\ &\quad \forall x \geq 0, 1 > y \geq 0. \end{aligned}$$

It can be easily checked that

$$g(x, 0) = x - \ln(1 + x) \geq 0,$$

and

$$g(0, y) = y - \ln(1 - y) \geq 0.$$

If there is a point  $(x_0, y_0)$  such that  $g(x_0, y_0) < 0$ , this function must have a minimal interior point. The gradient will be zero at such a point. However, it can be calculated that

$$\begin{aligned} \frac{\partial g}{\partial x} \Big|_{(x_0, y_0)} &= 1 - \frac{1}{1 + x_0} - y_0 = 0, \\ \frac{\partial g}{\partial y} \Big|_{(x_0, y_0)} &= -x_0 + 1 + \frac{1}{1 - y_0} = 0. \end{aligned}$$

The solution to this system of equations satisfies

$$(1 - y_0)(1 + x_0) = 1.$$

As such, at the minimal point there holds:

$$g(x_0, y_0) = x_0 - x_0 y_0 + y_0 = x_0(1 - y_0) + y_0 > 0,$$

which is a contradiction. Therefore, the minimum, if it exists, is achieved at the boundary.

Hence,

$$g(x, y) \geq 0, \quad \forall x \geq 0, 1 > y \geq 0.$$

Applying this inequality to (4.65), we obtain the right side of the inequality (4.62).

3. It is clear that  $p_{k+1} \in Q(p_k, 1)$  since

$$\nabla^2_{\frac{1}{1+\lambda_f(p_k)} X_{Nk}} f(p_k) = \left[ \frac{1}{1+\lambda_f(p_k)} \right]^2 \lambda_f(p_k)^2 < 1.$$

Applying (4.40), there holds

$$\begin{aligned} f(p_{k+1}) &\leq f(p_k) + \frac{1}{1+\lambda_f(p_k)} \nabla_{X_{Nk}} f(p_k) - \frac{1}{1+\lambda_f(p_k)} [\nabla_{X_{Nk}}^2 f(p_k)]^{1/2} \\ &\quad - \ln\left(1 - \frac{1}{1+\lambda_f(p_k)} [\nabla_{X_{Nk}}^2 f(p_k)]^{1/2}\right) \\ &= f(p_k) - \lambda_f(p_k) + \ln(1 + \lambda_f(p_k)), \end{aligned}$$

by the definition of  $\lambda_f(p_k)$  and the results in Theorem 4.4.1. Therefore, the inequality (4.63) is proved. □

Notice that the two functions

$$\lambda - \ln(1 + \lambda), \quad \forall \lambda \in (0, +\infty),$$

and

$$-\lambda - \ln(1 - \lambda), \quad \forall \lambda \in (0, 1),$$

are positive and monotonically increasing. The results proved in Theorem 4.5.2 have already given a set of error bounds for the function  $f(p)$  and estimation of the variable point  $p$  based on  $\lambda_f(p)$ . More specifically, the inequality (4.63) implies the following results:

**Corollary 4.5.1.** *For the Damped Newton algorithm, the  $\lambda_f(p_k)$  is bounded as follows:*

$$\lambda_f(p_k) - \ln(1 + \lambda_f(p_k)) \leq f(p_k) - f(p^*). \quad (4.66)$$

Furthermore, for a given precision  $\epsilon > 0$ , the number of iterations, denoted as  $N$ , required such that  $\lambda_f(p_N) < \epsilon$  is less than or equal to  $\frac{f(p_0) - f(p^*)}{\epsilon - \ln(1 + \epsilon)}$ .

**Theorem 4.5.3.** *Consider the optimization problem in (4.56). If the cost function  $f : N \rightarrow \mathbb{R}$  in (4.56) satisfies Assumption 3, then Algorithm 11 converges to the unique minimum of  $f$ .*

**Proof:** If Algorithm 11 terminates after a finite number of iterations, then there exists a finite number  $k$  such that

$$\text{grad}_{p_k} f = 0. \quad (4.67)$$

Hence, Algorithm 11 converges to the unique minimum of  $f$ .

Otherwise, let  $K = \{p \in N \mid f(p) < f(p_0)\}$  where  $p_0$  denotes the initial point. Let  $p^*$  be the solution of (4.56). Then for any  $p \in K$  in view of (4.36), we have

$$f(p) \geq f(p^*) + \omega([\nabla_{X_{pp^*}}^2 f(p)]^{\frac{1}{2}}) \quad (4.68)$$

where  $X_{pp^*} \in T_p N$  such that  $p^* = \text{Exp}(X_{pp^*})$  with the distance  $d(p, p^*)$  between  $p$  and  $p^*$  defined as

$$d(p, p^*) \leq \|X_{pp^*}\|_p. \quad (4.69)$$

It follows from (4.68) that

$$\omega([\nabla_{X_{pp^*}}^2 f(p)]^{\frac{1}{2}}) \leq f(p) - f(p^*) \leq f(p_0) - f(p^*). \quad (4.70)$$

Note that  $\omega(t)$  is strictly increasing in  $t$ . Therefore,

$$[\nabla_{X_{pp^*}}^2 f(p)]^{\frac{1}{2}} \leq \bar{t} \quad (4.71)$$

where  $\bar{t}$  is the unique positive root of the following equation

$$\omega(\bar{t}) = f(p_0) - f(p^*). \quad (4.72)$$

In view of (4.57), we have

$$[\nabla_{X_{pp^*}} f(p)]^{\frac{1}{2}} \geq \sqrt{\theta} \|X_{pp^*}\|_p. \quad (4.73)$$

Joining (4.69), (4.71) and (4.73), we obtain

$$d(p, p^*) \leq \frac{\bar{t}}{\sqrt{\theta}}. \quad (4.74)$$



Thus,  $K$  is closed bounded and hence compact.

In view of (4.63), we have

$$f(p_{k+1}) \leq f(p_k) - \omega(\lambda_k). \quad (4.75)$$

Summing up the inequalities (4.75) for  $k = 0 \dots N$ , we obtain

$$\sum_{k=0}^N \omega(\lambda_k) \leq f(p_0) - f(p_{N+1}) \leq f(p_0) - f(p^*), \quad (4.76)$$

where  $p^*$  is the solution of (4.56). As a consequence of (4.76), we have

$$\omega(\lambda_k) \rightarrow 0 \text{ as } k \rightarrow \infty, \quad (4.77)$$

and therefore  $\lambda_k \rightarrow 0$  as  $k \rightarrow \infty$ .

By (4.57) and (4.58), it follows from (4.77) that

$$X_{Nk} \rightarrow 0 \text{ as } k \rightarrow \infty. \quad (4.78)$$

Therefore, the theorem follows. □

## 4.6 Conclusion

This chapter reports our effort to generalize the definition and known results for self-concordant functions in Euclidean space to manifolds. It lays a comparative solid foundation to facilitate the construction of barrier functions for interior-point algorithms on manifolds.

For the proposed self-concordant function defined on a general class of smooth manifolds, a number of desirable properties are obtained. These include the feasibility of a Dikin-type ellipsoid and several inequalities that characterizes the similarity between self-concordant functions and quadratic functions along the geodesics of the manifold in various inequalities. Under the convexity condition on manifold defined by second order covariant differential, it is also shown that the optimal solution is global.

A Newton decrement is defined for this specific class of functions. This concept is analyzed in regards to the relationship between first order covariant derivatives along Newton direction and

along general direction, and to the maximal ratio of the norm of first order covariant derivative and that of second order derivative. The later facilitate the definition of the index  $\lambda_f(p)$ . With those theoretical preparation, the existence of global optimal solution is shown when  $\lambda_f(p) < 1$  holds for a point  $p$ .

A damped Newton algorithm is proposed to guarantee the convergence to the minimum of a self-concordant function.

## Chapter 5

# Damped Conjugate Gradient Methods on Riemannian Manifolds

### 5.1 Introduction

**Background** For the optimization of self-concordant functions in Euclidean space, Nesterov and Nemirovskii [58] developed a damped Newton method. Then, to reduce the computational cost, we present a damped gradient method and a conjugate gradient method. On the other hand, the notion of a self-concordant function has deep roots in geometry. In Chapter 4, self-concordance has been defined on Riemannian manifolds and the corresponding damped Newton method is proposed. As a result, they can provide guidance for the construction of efficient interior-point methods on smooth manifolds. However, the Newton-based method, on Riemannian manifolds as well as in Euclidean space, has a main drawback as a numerical optimization method. It is that in order to obtain the Newton descent direction, a linear system has to be solved at each iteration, which increases the computational cost. Alternatively, the conjugate gradient method can converge to the solution super-linearly without solving a linear system. In [67], Smith generalized the conjugate gradient method on Riemannian manifolds, which uses the exact geodesic (like a line in Euclidean space) search method to find the step-size. However, the geodesic search is only accepted in theory since it is often hard to compute in practice. Therefore, due to nice properties of self-concordant functions, we are motivated to develop a damped conjugate gradient method with a novel step-size rule for the optimization of such functions on Riemannian manifolds.

**Our work** Our method provides an explicit step-size rule based on the conjugate gradient

method. It is proved to converge to the optimal solution of a self-concordant function. The main advantage of our method is that it only uses the first and second covariant derivatives of the cost function without the need of computing a linear system. In each step, the complexity of our method is  $O(n^2)$  instead of  $O(n^3)$  for the damped Newton method, where  $n$  is the dimension of the Riemannian manifold.

**Chapter outline** The rest of this chapter is organized as follows. We will review the conjugate gradient method developed in [25] for optimization of cost functions on Riemannian manifolds in Section 4.3. Then in Section 5.3, the damped conjugate gradient method is proposed for optimization of the self-concordant function on Riemannian manifolds and it is proved that this method converges to the minimum of the self-concordant function. At last, we finish this chapter with a conclusion in 5.4.

## 5.2 Conjugate Gradient Method On Riemannian manifolds

Let  $M$  denote a smooth  $n$ -dimensional geodesic complete Riemannian manifold with Riemannian inner product  $\langle \cdot, \cdot \rangle_p$  on the tangent space to  $p \in M$ . We consider optimization problems of the form

$$\min_{x \in M} f : M \rightarrow \mathbb{R}. \quad (5.1)$$

In this section, we review the traditional conjugate gradient method on Riemannian manifolds proposed by Smith [67] to solve (5.1). One of the conjugate gradient algorithms to solve (5.1) goes as follows.

### Algorithm 12. (Conjugate Gradient Algorithm)

- step 0: Select an initial point  $p_0 \in N$ , compute  $H_0 = G_0 = -\text{grad}_{p_0} f$ , and set  $k = 0$ .
- step k: If  $\text{grad}_{p_k} f = 0$ , then terminate. Otherwise, compute

$$t_k = \min_{t \in \mathbb{R}} f(\text{Exp}_{p_k} tH_k), \quad (5.2)$$

such that  $\text{Exp}_{p_k} tH_k \in N$ .

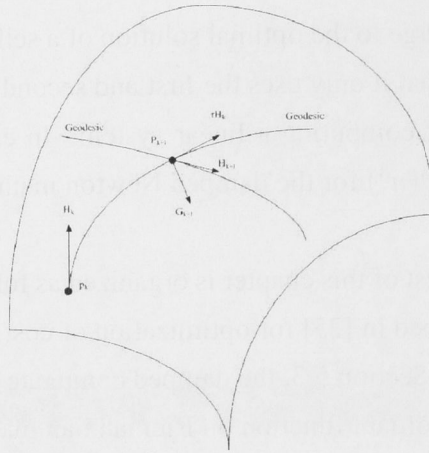


Figure 5.1: Conjugate gradient direction on Riemannian Manifolds

Then set

$$p_{k+1} = \text{Exp}_{p_k} t_k H_k, \tag{5.3}$$

$$G_{k+1} = -\text{grad}_{p_{k+1}} f, \tag{5.4}$$

$$\gamma_{k+1} = \frac{\langle G_{k+1}, G_{k+1} \rangle_{p_{k+1}}}{\langle G_k, H_k \rangle_{p_k}}, \tag{5.5}$$

$$H_{k+1} = G_{k+1} + \gamma_{k+1} \tau_{p_k p_{k+1}} H_k, \tag{5.6}$$

where  $\tau_{p_k p_{k+1}}$  is the parallel transport with respect to the geodesic from  $p_k$  to  $p_{k+1}$ . If  $k + 1 \pmod{n - 1} = 0$ , set  $H_{k+1} = G_{k+1}$ . Increment  $k$  and repeat until convergence.

Figure 5.1 sketches the conjugate gradient algorithm on a curved space. We select the conjugate descent method to determine  $\gamma_k$ ; other choice is possible:

$$\gamma_{k+1} = \frac{\langle G_{k+1}, G_{k+1} \rangle_{p_{k+1}}}{\langle G_k, H_k \rangle_{p_k}}. \tag{5.7}$$

The above algorithm uses the geodesic search to find the optimal step-size. However, in general, the geodesic search is only accepted in theory because it is often too hard to compute in practice. Since self-concordant functions on Riemannian manifolds has nice properties, we are motivated to introduce a novel step-size rule based on the conjugate gradient method for optimization of such functions.

## 5.3 Damped Conjugate Gradient Method

Let  $M$  denote a smooth  $n$ -dimensional geodesic complete Riemannian manifold with Riemannian structure  $g$ . Throughout this chapter,  $f$  denotes a real-valued function defined on an open convex subset  $N$  of  $M$ . Recall that, a subset  $N$  of  $M$  is convex if for any  $p, q \in N$ , the geodesic connecting  $p$  and  $q$  is a subset of  $N$ . Consider optimization problems of the form

$$\min_{x \in N} f : N \subset M \rightarrow R. \quad (5.8)$$

In general, it is hard to solve (5.8) since the domain of  $f$  is an open subset of a Riemannian manifold. However, if  $f$  has nice properties, there exist powerful techniques to solve (5.8). In Chapter 4, the case when  $f$  is self-concordant on Riemannian manifolds is considered. In this section, we focus on the special case of the optimization problem 5.8 when  $f$  in 5.8 satisfies the Assumption 3.

In this section, a damped conjugate gradient method for optimization of (5.8) is presented when  $f$  satisfies Assumption 3.

Suppose we are at a point  $p_k$  at time  $k$ . Given an appropriate step-size  $t_k$  and conjugate gradient direction  $H_k$  defined in (5.6), the conjugate gradient method sets  $p_{k+1} = \text{Exp}_{p_k} t_k H_k$ .

From (4.40), provided  $p_{k+1} \in W^0(p_k; 1)$ , we have

$$f(p_k) - f(p_{k+1}) \geq -\nabla_{t_k H_k} f(p_k) + [\nabla_{t_k H_k}^2 f(p_k)]^{1/2} + \ln(1 - [\nabla_{t_k H_k}^2 f(p_k)]^{1/2}). \quad (5.9)$$

We propose choosing  $t_k$  to maximize the right side hand in (5.9). Later in Theorem 5.3.1, it is proved that such a strategy guarantees convergence to the minimum of the cost function. Initially, we assume that  $\nabla_{H_k} f(p_k) < 0$ . Later, in Lemma 7, it is proved that this assumption is correct. Hence,  $t_k$  is required to be positive.

The right side of (5.9) is of the form  $\psi(t_k)$  where  $\psi(t) = \alpha t + \ln(1 - \beta t)$  with  $\alpha = -\nabla_{H_k} f(p_k) + \sqrt{\nabla_{H_k}^2 f(p_k)}$  and  $\beta = \sqrt{\nabla_{H_k}^2 f(p_k)}$ . Note that  $\beta$  will be strictly positive if we are not at the minimum of  $f$ . Therefore,  $\psi$  is defined on the interval  $[0, 1/\beta)$ . If  $t_k \in [0, 1/\beta)$ ,  $p_{k+1} \in W^0(p_k)$  as required for (5.9) to be a valid bound.

Differentiating  $\psi(t)$  yields

$$\psi'(t) = \alpha - \frac{\beta}{1 - \beta t}, \quad (5.10)$$

$$\psi''(t) = -\frac{\beta^2}{(1 - \beta t)^2} < 0, \quad (5.11)$$

showing that  $\psi(t)$  is concave on its domain  $[0, 1/\beta)$ . It achieves its maximum at

$$t = \frac{\alpha - \beta}{\alpha\beta}. \quad (5.12)$$

Let  $\lambda_k = \frac{-\nabla_{H_k} f(p_k)}{\sqrt{\nabla_{H_k}^2 f(p_k)}}$ . Substituting  $\alpha$  and  $\beta$  into  $t$ , we obtain

$$t_k = \frac{\lambda_k}{(1 + \lambda_k)\sqrt{\nabla_{H_k}^2 f(p_k)}}. \quad (5.13)$$

Therefore, the proposed damped conjugate gradient algorithm for (5.8) is as follows.

**Algorithm 13. (Damped Conjugate Gradient Algorithm)**

step 0: Select an initial point  $p_0 \in N$ , compute  $H_0 = G_0 = -\text{grad}_{p_0} f$ , and set  $k = 0$ .

step k: If  $\text{grad}_{p_k} f = 0$ , then terminate. Otherwise, compute

$$\lambda_k = \frac{-\nabla_{H_k} f(p_k)}{\sqrt{\nabla_{H_k}^2 f(p_k)}}, \quad (5.14)$$

$$t_k = \frac{\lambda_k}{(1 + \lambda_k)\sqrt{\nabla_{H_k}^2 f(p_k)}}, \quad (5.15)$$

$$p_{k+1} = \text{Exp}_{p_k} t_k H_k, \quad (5.16)$$

$$G_{k+1} = -\text{grad}_{p_{k+1}} f, \quad (5.17)$$

$$\gamma_{k+1} = \frac{\langle G_{k+1}, G_{k+1} \rangle_{p_{k+1}}}{\langle G_k, H_k \rangle_{p_k}}, \quad (5.18)$$

$$H_{k+1} = G_{k+1} + \gamma_{k+1} \tau_{p_k p_{k+1}} H_k, \quad (5.19)$$

where  $\tau_{p_k p_{k+1}}$  is the parallel translation with respect to the geodesic from  $p_k$  to  $p_{k+1}$ . If  $k + 1 \bmod n - 1 = 0$ , set  $H_{k+1} = G_{k+1}$ . Increment  $k$  and repeat until convergence.

The convergence of Algorithm 13 is demonstrated in Theorem 5.3.1 with the help of Lemma

7, 8 and 9.

**Lemma 7.** *Let the cost function  $f : N \rightarrow \mathbb{R}$  in (5.8) satisfy Assumption 3. Assume  $p_0$  is such that  $\text{grad}_{p_0} f \neq 0$ . Then either 1) Algorithm 13 terminates after a finite number iterations if  $\text{grad}_{p_k} f = 0$  at a certain  $k$ , or 2) Algorithm 13 generates an infinite sequence  $\{p_k\}$  of points (That is, there are no divisions by zeros) if zero gradient never encountered in the iteration and moreover,  $\forall k, \nabla_{H_k} f(p_k) = \langle \text{grad}_{p_k} f, H_k \rangle_{p_k} < 0$ .*

**Proof:**

1. If Algorithm 13 terminates, it means that there exists a finite number  $k$  such that

$$\text{grad}_{p_k} f = 0 \tag{5.20}$$

2. If Algorithm 13 generates an infinite sequence  $\{p_k\}$  of points, it implies that for all  $k$

$$\text{grad}_{p_k} f \neq 0. \tag{5.21}$$

The further proof is by induction. Since we reset the conjugate direction to the negative gradient every  $n$  steps, without loss of generality, we only consider the first  $n$  steps. When  $k = 0$ ,  $H_0 = -\text{grad}_{p_0} f$ . Then we get

$$\nabla_{H_0} f(p_0) = \langle \text{grad}_{p_0} f, -\text{grad}_{p_0} f \rangle_{p_0} = -\|\text{grad}_{p_0} f\|_{p_0}^2 < 0. \tag{5.22}$$

Assume that  $\nabla_{H_k} f(p_k) = \langle \text{grad}_{p_k} f, H_k \rangle_{p_k} < 0$  for some  $k \leq n - 1$ . It implies that  $p_{k+1}$  is well defined. Then we obtain

$$\begin{aligned} t_k &= \frac{\lambda_k}{(1 + \lambda_k) \sqrt{\nabla_{H_k}^2 f(p_k)}} \\ &= \frac{-\nabla_{H_k} f(p_k)}{(-\nabla_{H_k} f(p_k) + \sqrt{\nabla_{H_k}^2 f(p_k)}) \sqrt{\nabla_{H_k}^2 f(p_k)}}. \end{aligned} \tag{5.23}$$



Moreover, we have

$$\begin{aligned}
 & \nabla_{\tau_{p_k p_{k+1}} H_k} f(p_{k+1}) \\
 = & \nabla_{H_k} f(p_k) + \nabla_{\tau_{p_k p_{k+1}} H_k} f(p_{k+1}) - \nabla_{H_k} f(p_k) \\
 = & \nabla_{H_k} f(p_k) + \frac{\nabla_{\tau_{p_k p_{k+1}} H_k} f(p_{k+1}) - \nabla_{H_k} f(p_k)}{\nabla_{H_k} f(p_k)} \nabla_{H_k} f(p_k) \\
 = & \left( 1 + \frac{\nabla_{\tau_{p_k p_{k+1}} H_k} f(p_{k+1}) - \nabla_{H_k} f(p_k)}{\nabla_{H_k} f(p_k)} \right) \nabla_{H_k} f(p_k) \\
 = & \rho_k \nabla_{H_k} f(p_k)
 \end{aligned} \tag{5.24}$$

where  $\tau_{p_k p_{k+1}}$  is the parallel transport with respect to the geodesic from  $p_k$  to  $p_{k+1}$  and  $\rho_k = 1 + \frac{\nabla_{\tau_{p_k p_{k+1}} H_k} f(p_{k+1}) - \nabla_{H_k} f(p_k)}{\nabla_{H_k} f(p_k)}$ .

Furthermore,

$$\begin{aligned}
 & \nabla_{\tau_{p_k p_{k+1}} H_k} f(p_{k+1}) - \nabla_{H_k} f(p_k) \\
 = & \frac{1}{t_k} \int_0^1 \nabla_{t_k \tau_{p_k} \text{Exp}_{p_k} t t_k H_k}^2 f(\text{Exp}_{p_k} t t_k H_k) dt \\
 \leq & \frac{\nabla_{t_k H_k}^2 f(p_k)}{t_k (1 - t_k \sqrt{\nabla_{H_k}^2 f(p_k)})} \\
 = & \frac{t_k \nabla_{H_k}^2 f(p_k)}{1 - t_k \sqrt{\nabla_{H_k}^2 f(p_k)}} \\
 = & -\nabla_{H_k} f(p_k)
 \end{aligned} \tag{5.25}$$

where the inequality is obtained from (4.44) and the last equality by substituting  $t_k$ .

Since  $\nabla_{H_k} f(p_k) < 0$  by assumption and  $\nabla_{\tau_{p_k p_{k+1}} H_k} f(p_{k+1}) - \nabla_{H_k} f(p_k) > 0$  by (5.25), we have

$$0 > \frac{\nabla_{\tau_{p_k p_{k+1}} H_k} f(p_{k+1}) - \nabla_{H_k} f(p_k)}{\nabla_{H_k} f(p_k)} \geq \frac{-\nabla_{H_k} f(p_k)}{\nabla_{H_k} f(p_k)} = -1 \tag{5.27}$$

As such, we obtain

$$0 \leq \rho_k < 1. \tag{5.28}$$

For the conjugate gradient algorithm, since  $\gamma_{k+1} = \frac{\langle G_{k+1}, G_{k+1} \rangle_{p_{k+1}}}{\langle G_k, H_k \rangle_{p_k}} = \frac{\|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^2}{-\nabla_{H_k} f(p_k)}$ , we have

$$\begin{aligned}
 & \nabla_{H_{k+1}} f(p_{k+1}) \\
 &= \langle \text{grad}_{p_{k+1}} f, H_{k+1} \rangle_{p_{k+1}} \\
 &= \langle \text{grad}_{p_{k+1}} f, -\text{grad}_{p_{k+1}} f + \gamma_{k+1} \tau_{p_k p_{k+1}} H_k \rangle_{p_{k+1}} \\
 &= -\|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^2 + \frac{\|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^2}{-\nabla_{H_k} f(p_k)} \nabla_{\tau_{p_k p_{k+1}} H_k} f(p_{k+1}) \\
 &= -\|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^2 - \rho_k \|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^2 \\
 &= -(1 + \rho_k) \|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^2 < 0.
 \end{aligned} \tag{5.29}$$

Similarly, we can prove for all  $k$ ,

$$\nabla_{H_k} f(p_k) = \langle \text{grad}_{p_k} f, H_k \rangle_{p_k} < 0. \tag{5.30}$$

As a result, we complete the proof of Lemma 7. □

**Lemma 8.** *Let  $\{p_k\}$  be an infinite sequence of points generated by Algorithm 13 where the cost function  $f : N \rightarrow R$  satisfies Assumption 3. Then:*

1.  $\forall k, p_k \in N$ .
2. If  $\text{grad}_{p_{k+1}} f \neq 0$ , then  $\lambda_k > 0$ .
3. If  $\text{grad}_{p_{k+1}} f \neq 0$ , then  $f(p_{k+1}) \leq f(p_k) + \omega(\lambda_k) < f(p_k)$  where  $\omega(t) = t - \ln(1 + t)$ .

**Proof:**

1. From the earlier derivation, it was already proved that

$$p_{k+1} \in W(p_k) \tag{5.31}$$

Therefore, from Proposition 4.3.3 and the fact that  $p_0 \in N$ , it follows that  $p_k \in N$ .

2. Since  $\text{grad}_{p_{k+1}} f \neq 0$ , it follows from Lemma 7 that  $\nabla_{H_k} f(p_k) < 0$ . Then, it implies that  $\lambda_k > 0$  by the definition of  $\lambda_k$ .
3. Substituting  $t_k$  into (5.9), we obtain

$$f(p_{k+1}) \leq f(p_k) - \omega(\lambda_k) \quad (5.32)$$

where  $\omega(t) = t - \ln(1+t) > 0$  since  $\lambda_k > 0$  by 2. □

**Lemma 9.** *Let  $\{p_k\}$  and  $\{H_k\}$  be infinite sequences generated by Algorithm 13 where the cost function  $f : N \rightarrow R$  satisfies Assumption 3. If  $\text{grad}_{p_k} f \neq 0$ , then for all  $k$*

$$\frac{\|H_{k+1}\|_{p_{k+1}}^2}{\|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^4} \leq \frac{\|H_k\|_{p_k}^2}{\|\text{grad}_{p_k} f\|_{p_k}^4} + \frac{3}{\|\text{grad}_{p_k} f\|_{p_k}^2}. \quad (5.33)$$

**Proof:** First we consider the iterations when  $k+1 \pmod n \neq 0$ . Note that from (5.24) and the inequality (5.28), we have

$$\begin{aligned} (\nabla_{H_{k+1}} f(p_{k+1}))^2 &= (1 + \rho_k)^2 \|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^4 \\ &\geq \|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^4. \end{aligned} \quad (5.34)$$

Moreover, in view of (5.17), (5.19), (5.24) and (5.29), we have

$$\begin{aligned} &\|H_{k+1}\|_{p_{k+1}}^2 \\ &= \left\| -\text{grad}_{p_{k+1}} f + \gamma_{k+1} \tau_{p_k p_{k+1}} H_k \right\|_{p_{k+1}}^2 \\ &= \|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^2 + \frac{\|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^4}{(\nabla_{H_k} f(p_k))^2} \|\tau_{p_k p_{k+1}} H_k\|_{p_{k+1}}^2 \\ &+ 2 \frac{\|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^2}{\nabla_{H_k} f(p_k)} \nabla_{\tau_{p_k p_{k+1}} H_k} f(p_{k+1}) \\ &= \|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^2 + \frac{\|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^4}{(\nabla_{H_k} f(p_k))^2} \|H_k\|_{p_k}^2 \\ &+ 2\rho_k \|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^2 \\ &\leq 3\|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^2 + \frac{\|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^4}{\|\text{grad}_{p_k} f\|_{p_k}^4} \|H_k\|_{p_k}^2 \end{aligned} \quad (5.35)$$

Dividing both sides of (5.35) by  $\|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^4$ , we obtain

$$\frac{\|H_{k+1}\|_{p_{k+1}}^2}{\|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^4} \leq \frac{\|H_k\|_{p_k}^2}{\|\text{grad}_{p_k} f\|_{p_{k+1}}^4} + \frac{3}{\|\text{grad}_{p_k} f\|_{p_k}^2}. \quad (5.36)$$

Now, we consider the iterations when  $k + 1 \pmod n = 0$ . In these iterations, we reset the conjugate direction to the negative gradient. Therefore, we have

$$\begin{aligned} \frac{\|H_{k+1}\|_{p_{k+1}}^2}{\|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^4} &= \frac{1}{\|\text{grad}_{p_{k+1}} f\|_{p_{k+1}}^2} \\ &\leq \frac{\|H_k\|_{p_k}^2}{\|\text{grad}_{p_k} f\|_{p_k}^4} + \frac{3}{\|\text{grad}_{p_{k+1}} f\|_{p_k}^2}. \end{aligned} \quad (5.37)$$

As a consequence, this lemma follows. □

**Theorem 5.3.1.** *Consider the optimization problem in (5.8). If the cost function  $f : N \rightarrow \mathbb{R}$  in (5.8) satisfies Assumption 3, then Algorithm 13 converges to the unique minimum of  $f$ .*

**Proof:** If Algorithm 13 terminates after a finite number of iterations, then there exists a finite number  $k$  such that

$$\text{grad}_{p_k} f = 0. \quad (5.38)$$

Hence, Algorithm 13 converges to the unique minimum of  $f$ .

Otherwise, let  $K = \{p \in N \mid f(p) < f(p_0)\}$  where  $p_0$  denotes the initial point. Let  $p^*$  be the solution of (5.8). Then for any  $p \in K$  in view of (4.36), we have

$$f(p) \geq f(p^*) + \omega([\nabla_{X_{pp^*}}^2 f(p)]^{\frac{1}{2}}) \quad (5.39)$$

where  $X_{pp^*} \in T_p N$  such that  $p^* = \text{Exp}(X_{pp^*})$  with the distance  $d(p, p^*)$  between  $p$  and  $p^*$  defined as

$$d(p, p^*) = \|X_{pp^*}\|. \quad (5.40)$$

It follows from (5.39) that

$$\omega([\nabla_{X_{pp^*}}^2 f(p)]^{\frac{1}{2}}) \leq f(p) - f(p^*) \leq f(p_0) - f(p^*). \quad (5.41)$$

Note that  $\omega(t)$  is strictly increasing in  $t$ . Therefore,

$$[\nabla_{X_{pp^*}}^2 f(p)]^{\frac{1}{2}} \leq \bar{t} \tag{5.42}$$

where  $\bar{t}$  is the unique positive root of the following equation

$$\omega(t) = f(p_0) - f(p^*). \tag{5.43}$$

In view of (4.57), we have

$$[\nabla_{X_{pp^*}} f(p)]^{\frac{1}{2}} \geq \sqrt{\theta} \|X_{pp^*}\|_p. \tag{5.44}$$

Joining (5.40), (5.42) and (5.44), we obtain

$$d(p, p^*) \leq \frac{\bar{t}}{\sqrt{\theta}}. \tag{5.45}$$

Thus,  $K$  is closed bounded and hence compact.

By (4.57), (5.14) and (5.29), we obtain

$$\begin{aligned} \lambda_k &= \frac{-\nabla_{H_k} f(p_k)}{\sqrt{\nabla_{H_k}^2 f(p_k)}} \\ &\geq \frac{(1 + \rho_{k-1}) \|\text{grad}_{p_k} f\|_{p_k}^2}{\sqrt{\alpha} \|H_k\|_{p_k}} \\ &\geq \frac{\|\text{grad}_{p_k} f\|_{p_k}^2}{\sqrt{\alpha} \|H_k\|_{p_k}}. \end{aligned} \tag{5.46}$$

From Lemma 8, we have

$$f(p_{k+1}) \leq f(p_k) - \omega(\lambda_k). \tag{5.47}$$

Summing up the inequalities (5.47) for  $k = 0 \dots N$ , we obtain

$$\sum_{k=0}^N \omega(\lambda_k) \leq f(p_0) - f(p_{N+1}) \leq f(p_0) - f(p^*) \tag{5.48}$$

where  $p^*$  is the solution of (5.8). As a consequence of (5.48), we have

$$\sum_{k=0}^{\infty} \omega(\lambda_k) < +\infty. \quad (5.49)$$

Assume  $\liminf_{k \rightarrow \infty} \|\text{grad}_{p_k} f\|_{p_k} \neq 0$ . Then there exists  $d > 0$  such that  $\|\text{grad}_{p_k} f\|_{p_k} \geq d$  for all  $k$ . Therefore, it follows from Lemma 9

$$\frac{\|H_i\|_{p_i}^2}{\|\text{grad}_{p_i} f\|_{p_i}^4} \leq \frac{\|H_{i-1}\|_{p_{i-1}}^2}{\|\text{grad}_{p_{i-1}} f\|_{p_{i-1}}^4} + \frac{3}{\|\text{grad}_{p_{i-1}} f\|_{p_{i-1}}^2}. \quad (5.50)$$

Adding up the above inequalities for  $i = 0, \dots, k$ , we get

$$\frac{\|H_k\|_{p_k}^2}{\|\text{grad}_{p_k} f\|_{p_k}^4} \leq \frac{\|H_0\|_{p_0}^2}{\|\text{grad}_{p_0} f\|_{p_0}^4} + \frac{3k}{d^2}. \quad (5.51)$$

Let  $a = \frac{3}{d^2}$  and  $b = \frac{\|H_0\|_{p_0}^2}{\|\text{grad}_{p_0} f\|_{p_0}^4} = \frac{1}{\|\text{grad}_{p_0} f\|_{p_0}^2}$ . Then it follows from (5.51)

$$\frac{\|\text{grad}_{p_k} f\|_{p_k}^4}{\|H_k\|_{p_k}^2} \geq \frac{1}{ka + b}. \quad (5.52)$$

Combining (5.46) and (5.52), we obtain

$$\lambda_k \geq \frac{c}{\sqrt{ka + b}} \quad (5.53)$$

where  $c = \frac{1}{\sqrt{\alpha}}$ .

Let  $\{\beta_k\}$  be a sequence such that  $\beta_k = \frac{c}{\sqrt{ka + b}}$ . Then it is easy to show

$$\sum_{k=1}^{\infty} \beta_k^2 = +\infty. \quad (5.54)$$

Consider the sequence  $\{\omega(\beta_k)\}$ . Since  $a, b, c$  are constant, we have

$$\lim_{k \rightarrow \infty} \frac{\omega(\beta_k)}{\beta_k^2} = \lim_{t \rightarrow 0} \frac{t - \ln(1 + t)}{t^2} = \frac{1}{2}. \quad (5.55)$$

It follows from (5.54) and (5.55)

$$\sum_{k=1}^{\infty} \omega(\beta_k) = +\infty. \quad (5.56)$$

Since  $\omega(t)$  is increasing with respect to  $t$ , by (5.53) and (5.56) we obtain

$$\sum_{k=1}^{\infty} \omega(\lambda_k) = +\infty \quad (5.57)$$

which is contradictory to (5.49). Therefore, we have

$$\liminf_{k \rightarrow \infty} \|\text{grad}_{p_k} f\|_{p_k} = 0. \quad (5.58)$$

Hence, the theorem follows. □

## 5.4 Conclusion

In this chapter, we propose a damped conjugate gradient method for optimization of self-concordant functions on Riemannian manifolds. Such method is an ordinary conjugate gradient method but with a novel step-size selection rule which is proved to ensure that this algorithm converges to the global minimum. The advantage of the damped conjugate gradient method over the damped Newton method is that the former has a lower computational complexity. Both methods are applied to examples in the next section and shown to converge to the minimum of a self-concordant function.

## Chapter 6

# Application of Damped Methods on Riemannian Manifolds

In this chapter, we apply the damped Newton method in Chapter 4 and the damped conjugate gradient method in Chapter 5 to three examples.

In the first example, the cost function is defined on the hyperbola model and proved to be self-concordant.

In the second example, the cost function is defined on the part of the sphere and proved to be self-concordant. Hence, it can be thought of as a barrier function on the given domain.

In the third example, given some points  $p_1, \dots, p_k$  on the Hyperboloid model  $I_n$ , the problem is to find the point on  $I_n$  which minimizes the mean squared intrinsic distance to every point of  $p_1, \dots, p_k$ . This minimum is also known as the “Karcher mean”, first introduced in [42] as the centre of mass on a Riemannian manifold. The methods to find the “Karcher mean” on Riemannian manifolds have been well studied. For instance, see [55, 2]. However, the problems in [55, 2] are defined on Riemannian manifolds with positive curvatures. Even though these methods can still be used to find the “Karcher mean” on Riemannian manifolds with negative curvatures, until now, we are not aware of the particular method based on the properties of negative curvatures. In [42], it is shown that the “Karcher mean” function defined on the Riemannian manifolds with negative curvatures is convex. By this result, since the Hyperboloid model has constant negative curvature, we proved that the “Karcher mean” function defined on this model is self-concordant. Simulation results show our method converges to the “Karcher mean” of given points on the Hyperboloid model super-linearly.



## 6.1 Example One

Consider the following optimization problem:

$$\begin{aligned} & \text{minimize} && f(x) := x_1 + x_2 \\ & \text{subject to:} && x_1, x_2 > 0, x = (x_1, x_2)^T \in H \end{aligned}$$

where  $H$  is a hyperbola satisfying  $x_1x_2 = 1$ . Let  $T_xH$  be the tangent space of  $H$  at  $x$ , i. e.,  $T_xH = \{\alpha h | h = (-x_1, x_2)^T, \alpha \in \mathbb{R}\}$ . For any  $x = (x_1, x_2)^T \in H$ ,  $h_1 = \alpha(-x_1, x_2)^T \in T_xH$  and  $h_2 = \beta(-x_1, x_2)^T \in T_xH$  where  $\alpha, \beta \in \mathbb{R}$ , the Riemannian metric is defined as

$$\langle h_1, h_2 \rangle_x = \alpha\beta. \tag{6.1}$$

Then, the geodesic on the hyperbola can be calculated as:

$$\text{Exp}_x tX = Ax \tag{6.2}$$

Where  $A = \begin{pmatrix} e^{-at} & 0 \\ 0 & e^{at} \end{pmatrix}$  and  $X = \alpha h \in T_xH$ . Hence, covariant differentials can be calculated as follows:

$$\begin{aligned} \nabla_X f(x) &= (-x_1 + x_2)\alpha \\ \nabla_X^2 f(x) &= (x_1 + x_2)\alpha^2 \\ \nabla_X^3 f(x) &= (-x_1 + x_2)\alpha^3 \end{aligned}$$

It can be seen that  $\nabla_X^2 f(x)$  is positive definite. Then

$$\frac{(\nabla_X^3 f(x))^2}{(\nabla_X^2 f(x))^3} = \frac{(x_2 - x_1)^2}{(x_1 + x_2)^3} \leq 1 \tag{6.3}$$

As such,  $f(x)$  is a self-concordant function.

Now we can apply the proposed damped Newton algorithm.

**Algorithm 14.** (*Damped Newton Algorithm*)

*step 0:* randomly generate a feasible initial point  $x^0$ .

Step k	x	f(x)	$\lambda(x)$
0	(6.0000,0.1667)	6.1667	2
1	(0.2525,3.9601)	4.2126	2.3490
2	(2.9852,0.3350)	3.3202	1.8064
3	(0.4208,2.3762)	2.7971	1.4545
4	(1.9173,0.5216)	2.4389	1.1692
5	(0.6487,1.5417)	2.1903	0.8938
6	(1.2471,0.8018)	2.0490	0.6034
7	(0.9379,1.0662)	2.0041	0.3111
8	(1.0057,0.9943)	2.0000	0.0906
9	(1.0000,1.0000)	2.0000	0.0081
10	(1.0000,1.0000)	2.0000	0.0001
11	(1.0000,1.0000)	2.0000	0.0000

Table 6.1: The simulation result for Example 1

step  $k$ : calculate the  $k$ -th step according to:

$$x^k = \text{EXP}_{x^{k-1}}\left(\frac{1}{1 + \lambda(x^{k-1})} X_N\right) = A^{k-1} x^{k-1}$$

where

$$A^{k-1} = \begin{pmatrix} e^{-\frac{1}{1+\lambda(x^{k-1})}\alpha^{k-1}} & 0 \\ 0 & e^{\frac{1}{1+\lambda(x^{k-1})}\alpha^{k-1}} \end{pmatrix},$$

$$\lambda(x^{k-1}) = \sqrt{(x_1^{k-1} + x_2^{k-1})\alpha^2}$$

$$X_N = \alpha^{k-1}(-x_1^{k-1}, x_2^{k-1})^T, \quad \alpha^{k-1} = -\frac{x_2^{k-1} - x_1^{k-1}}{x_1^{k-1} + x_2^{k-1}}.$$

The simulation result is shown in Table 2.

## 6.2 Example Two

Consider the following simple optimization problem:

$$\begin{aligned} \min \quad & f(x) := -\ln(x_1 x_2) \\ \text{subject to:} \quad & x_1, x_2 > 0, x = (x_1, x_2) \in S^1, \end{aligned}$$

where  $S^1$  is unit circle. We define a Riemannian metric as the induced metric from the ambient Euclidean space. Let  $x \in S^1$  and  $h \in T_x S^1$  have unit length. Then the geodesic on the unit circle is  $\text{Exp}_x th = x \cos(t) + h \sin(t)$ . Hence, the following covariant differentials can be calculated:

$$\begin{aligned}\nabla_h f(x) &= -\frac{h_1}{x_1} - \frac{h_2}{x_2} \\ \nabla_h^2 f(x) &= \frac{h_1^2}{x_1^2} + \frac{h_2^2}{x_2^2} + 2 \\ \nabla_h^3 f(x) &= -\left(\frac{h_1^3}{x_1^3} + \frac{h_1}{x_1} + \frac{h_2^3}{x_2^3} + \frac{h_2}{x_2}\right)\end{aligned}$$

It is obvious that  $\nabla_h^2 f(x)$  is positive definite.

Let  $x \in S^1$  and  $h \in T_x S^1$  and  $\|h\| = 1$ . Notice that  $h = (-x_2, x_1)$  or  $h = (x_2, -x_1)$ . Therefore,

$$\frac{(\nabla_h^3 f(x))^2}{(\nabla_h^2 f(x))^3} = \frac{4\left(\frac{h_1^3}{x_1^3} + \frac{h_1}{x_1} + \frac{h_2^3}{x_2^3} + \frac{h_2}{x_2}\right)^2}{\left(\frac{h_1^2}{x_1^2} + \frac{h_2^2}{x_2^2} + 2\right)^3} \leq 4. \quad (6.4)$$

As such,  $f(x)$  is self-concordant function. Now the damped Newton algorithm proposed in this paper becomes:

**Algorithm 15.** (*Damped Newton Algorithm*)

- step 0: randomly generate a feasible initial point  $x_0$ .
- step k: calculate the  $k$ -th step according to:

$$x^k = x^{k-1} \cos\left(\frac{\|X_N\|}{1 + \lambda(x^{k-1})}\right) + \frac{X_N}{\|X_N\|} \sin\left(\frac{\|X_N\|}{1 + \lambda(x^{k-1})}\right),$$

where

$$X_N = \left( \frac{x_1 x_2^2 (x_1^2 - x_2^2)}{x_2^4 + x_1^2 x_2^4 + x_1^4 (1 + x_2^2)}, \frac{-x_1^4 x_2 + x_1^2 x_2^3}{x_2^4 + x_1^2 x_2^4 + x_1^4 (1 + x_2^2)} \right)^T$$

and

$$\lambda(x) = \sqrt{2 + \frac{X_{N1}^2}{\|X_N\| x_1^2} + \frac{X_{N2}^2}{\|X_N\| x_2^2}}.$$

The following figure is a simulation result with the initial point  $(0.4359, 0.9000)^T$ . It demonstrates the quadratic convergence of the proposed algorithm. Now consider the following opti-

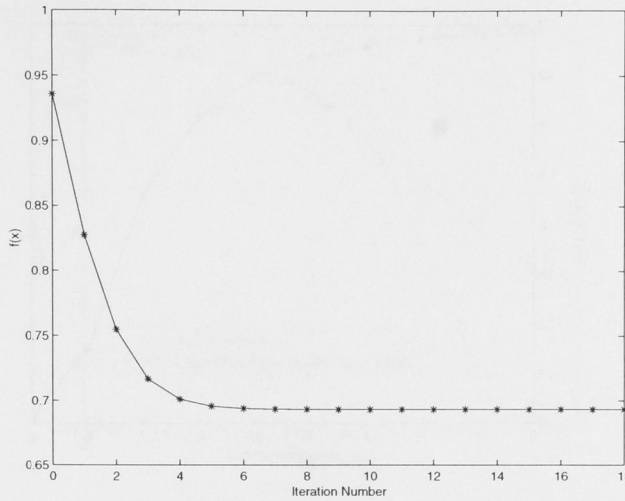


Figure 6.1: The result of damped Newton method for the self-concordant function defined on the circle

mization problem:

$$\begin{aligned} \min f(x) &:= -\ln x_1 - \dots - \ln x_n, \\ \text{subject to : } &\begin{cases} x = (x_1, x_2, \dots, x_n) \in S^{n-1}, \\ 0 < x_1, \dots, x_n < 1, \end{cases} \end{aligned} \tag{6.5}$$

where  $S^{n-1}$  is the unit sphere with  $x^T x = 1$ . Here, we define a Riemannian metric as the induced metric from the ambient Euclidean space, i.e.  $\langle y, z \rangle = y^T z$  where  $y, z \in T_x S^{n-1}$ . Let  $x \in S^{n-1}$  and  $h = (h_1, h_2, \dots, h_n) \in T_x S^{n-1}$  have unit length (if not, we can normalize it). Then, the geodesic on the sphere is  $\text{Exp}_x th = x \cos(t) + h \sin(t)$ , and the parallel transport along the geodesic  $\tau h = h \cos(t) - x \sin(t)$ . Therefore, the following covariant differentials can be calculated:

$$\begin{aligned} \nabla_h f(x) &= -\frac{h_1}{x_1} - \frac{h_2}{x_2}, \dots, -\frac{h_n}{x_n} \\ \nabla_h^2 f(x) &= \frac{h_1^2}{x_1^2} + \frac{h_2^2}{x_2^2} + \dots + \frac{h_n^2}{x_n^2} + n \\ \nabla_h^3 f(x) &= -2\left(\frac{h_1^3}{x_1^3} + \frac{h_1}{x_1} + \frac{h_2^3}{x_2^3} + \frac{h_2}{x_2} + \dots + \frac{h_n^3}{x_n^3} + \frac{h_n}{x_n}\right) \end{aligned}$$

The following procedure is to prove that the function  $f$  is self-concordant defined on  $S^{n-1}$ .

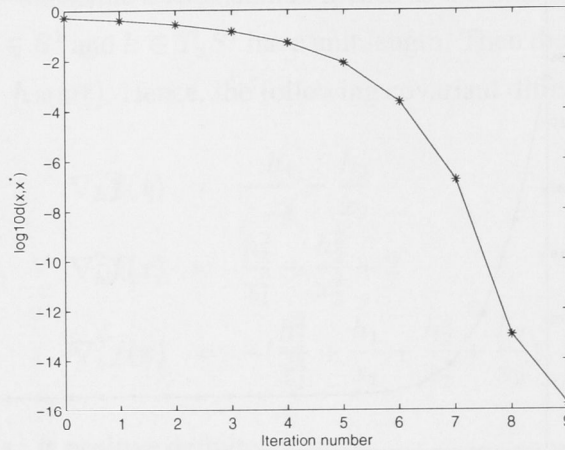


Figure 6.2: The result of damped Newton method for the self-concordant function defined on high-dimension sphere

It is obvious that for any  $h \in T_x S^{n-1}$ , the second covariant differentials  $\nabla_h^2 f(x)$  are always positive.

Let  $y_1 = \frac{h_1}{x_1}, \dots, y_n = \frac{h_n}{x_n}, y = (y_1, y_2, \dots, y_n)^T, b = ((y_1^2 + 1), (y_2^2 + 1), \dots, (y_n^2 + 1))^T$ . Then, we have

$$\begin{aligned}
 & \frac{(\nabla_h^3 f(x))^2}{(\nabla_h^2 f(x))^3} \\
 &= \frac{4(y_1^3 + \dots + y_n^3 + y_1 + \dots + y_n)}{(y_1^2 + \dots + y_n^2 + n)^3} \\
 &= 4 \frac{(y_1(y_1^2 + 1) + \dots + y_n(y_n^2 + 1))^2}{((y_1^2 + 1) + \dots + (y_n^2 + 1))^3} \\
 &\leq \frac{4(y_1^2 + \dots + y_n^2)}{(y_1^2 + 1) + \dots + (y_n^2 + 1)} \\
 &\leq 4
 \end{aligned} \tag{6.6}$$

Therefore, the function  $f$  is a self-concordant function defined on  $S^{n-1}$  with  $M_f = 2$ .

We apply damped Newton and damped conjugate gradient algorithms to this problem. In particular,  $n = 10$ . Figure 6.3 illustrates the result of the damped Newton method on function  $f$ . This result also demonstrates the quadratic convergence of the damped Newton algorithm.

Figure 6.3 illustrates the result of the damped conjugate gradient method on function  $f$ . This result also shows the superlinear convergence of the damped conjugate gradient method.

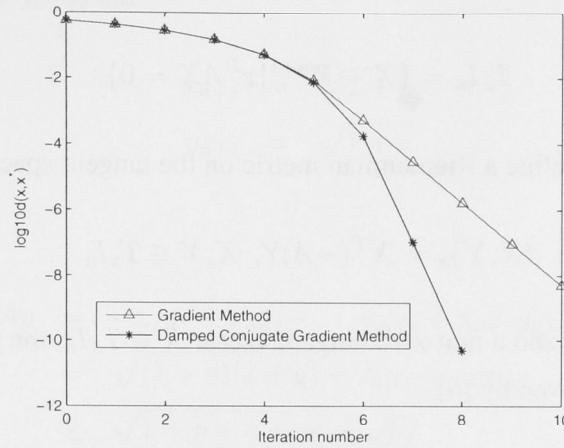


Figure 6.3: The result of damped conjugate gradient method for the self-concordant function defined on high-dimension sphere

### 6.3 Example Three

In this subsection, we consider the problem of computing the center of mass of a set of given points defined on the hyperboloid model. Before defining this problem, we first introduce the geometric properties of the hyperboloid model. In  $\mathbb{R}^{n+1}$ , consider the following quadratic form  $Q$ ,

$$Q(x) = - \sum_{i=1}^n x_i^2 + x_{n+1}^2. \tag{6.7}$$

Let  $A = \text{diag}(\underbrace{-1, -1, \dots, -1}_n, 1)$ . Then  $Q$  can be represented in terms of  $A$  by

$$Q(x) = x^T Ax. \tag{6.8}$$

Given this quadratic form, the upper fold  $I_n$  of the hyperboloid is determined by the formula

$$I_n = \{x \in \mathbb{R}^{n+1} | Q(x) = 1, x_{n+1} > 0\}. \tag{6.9}$$

The set  $I_n$  can be regarded as a differentiable hypersurface in  $\mathbb{R}^{n+1}$  since it is an open subset of the pre-image of a regular value of a differentiable function. In particular, it inherits from

$\mathbb{R}^{n+1}$  a differentiable structure of dimension  $n$ . For any  $x \in I_n$ , the tangent space  $T_x I_n$  is

$$T_x I_n = \{X \in \mathbb{R}^{n+1} | x^T A X = 0\}. \quad (6.10)$$

For any  $x \in I_n$ , we define a Riemannian metric on the tangent space of  $x$  by

$$\langle X, Y \rangle_x = X^T (-A) Y, \quad X, Y \in T_x I_n. \quad (6.11)$$

Given a point  $x \in I_n$  and a non-zero tangent vector  $X \in T_x I_n$ , the geodesic emanating from  $x$  in the direction  $X$  is given by [6]

$$\text{Exp}_x tX = x \cosh(\theta t) + \frac{1}{\theta} X \sinh(\theta t) \quad (6.12)$$

where  $\theta = \sqrt{X^T (-A) X}$ .

The intrinsic distance between  $x \in I_n$  and  $y \in I_n$  on this hyperboloid model is given by [73]

$$d(x, y) = \text{arccosh}(x^T A y). \quad (6.13)$$

Recall that  $\text{arccosh}(t) = \ln(t + \sqrt{t^2 - 1})$  for  $t > 1$ . Since  $d(x, y) \geq 0$ , we can expect that  $x^T A y \geq 1$  holds for all  $x, y \in I_n$ . Since we require this result elsewhere, we prove it from first principles.

**Proposition 6.3.1.** *For any two points  $x, y \in I_n$ , there holds*

$$x^T A y \geq 1. \quad (6.14)$$

**Proof:** Since  $x, y \in I_n$ , we have

$$-x_1^2 - x_2^2 - \cdots - x_n^2 + x_{n+1}^2 = 1, \quad (6.15)$$

$$-y_1^2 - y_2^2 - \cdots - y_n^2 + y_{n+1}^2 = 1. \quad (6.16)$$

Let  $p = x_1^2 + x_2^2 + \cdots + x_n^2$  and  $q = y_1^2 + y_2^2 + \cdots + y_n^2$ . Since  $x_{n+1} > 0$  and  $y_{n+1} > 0$ , it

follows from (6.15) and (6.16) that

$$x_{n+1} = \sqrt{1+p}, \quad (6.17)$$

$$y_{n+1} = \sqrt{1+q}. \quad (6.18)$$

Then, we obtain

$$\begin{aligned} x^T Ay &= -x_1y_1 - x_2y_2 - \dots - x_ny_n + x_{n+1}y_{n+1} \\ &= \sqrt{(1+p)(1+q)} - x_1y_1 - x_2y_2 - \dots - x_ny_n \\ &\geq \sqrt{1+p+q+pq} - \sqrt{pq} \\ &\geq \sqrt{1+2\sqrt{pq}+pq} - \sqrt{pq} \\ &= 1 + \sqrt{pq} - \sqrt{pq} \\ &= 1. \end{aligned} \quad (6.19)$$

□

Consider the following optimization problem

$$\arg \min_{p \in I_n} f(p) = \frac{1}{2} \sum_{i=1}^n d^2(p, p_i) = \frac{1}{2} \sum_{i=1}^n \operatorname{arccosh}^2(p^T Ap_i) \quad (6.20)$$

where  $p_i \in I_n$ ,  $i = 1, \dots, m$ . The solution of (6.20) is called the Karcher mean [42] of the given points  $p_1, \dots, p_m$ .

By computation, we obtain the first, second and third covariant derivatives of the cost function at  $p \in I_n$  in the direction  $H \in T_p I_n$  by

$$\nabla_H f(p) = \theta \sum_{i=1}^m \operatorname{arccosh}(p^T Ap_i) \frac{X^T Ap_i}{\sqrt{(p^T (-A)p_i)^2 - 1}}, \quad (6.21)$$

$$\begin{aligned} \nabla_H^2 f(p) &= \theta^2 \sum_{i=1}^m \left[ \frac{(X^T Ap_i)^2}{(p^T Ap_i)^2 - 1} \right. \\ &\quad \left. + \operatorname{arccosh}(p^T Ap_i) \frac{p^T Ap_i [(p^T Ap_i)^2 - (X^T Ap_i)^2 - 1]}{(p^T Ap_i)^2 - 1)^{\frac{3}{2}}} \right], \end{aligned} \quad (6.22)$$

$$\begin{aligned} \nabla_H^3 f(p) &= \theta^3 \sum_{i=1}^m \left[ \frac{X^T Ap_i}{((p^T Ap_i)^2 - 1)^{\frac{3}{2}}} - \frac{(X^T Ap_i)^3}{((p^T Ap_i)^2 - 1)^{\frac{5}{2}}} \right] \\ &\quad (3p^T Ap_i (p^T Ap_i^2 - 1)^{\frac{1}{2}} - \operatorname{arccosh}(p^T Ap_i) (2p^T Ap_i^2 + 1)) \end{aligned} \quad (6.23)$$



where  $\theta = \sqrt{H^T(-A)H}$  and  $X = H/\theta$ .

By Theorem 2.1 in Page 111 in [71], since  $I_n$  is simply connected, complete with negative sectional curvature, the function  $f(p)$  in (6.20) is strictly convex. Hence, this implies that for all  $p \in I_n$  and  $H \in T_p I_n$

$$\nabla_H^2 f(p) > 0. \tag{6.24}$$

For any  $p \in I_n$ , the gradient  $\text{grad}_p f$  of  $f$  is given by

$$\text{grad}_p f = (k^T A p)p - k \tag{6.25}$$

where  $k = \sum_{i=1}^m \frac{\arccos(p^T A p_i) p_i}{\sqrt{(p^T(-A)p_i)^2 - 1}}$ .

To prove that  $f$  in (6.20) is self-concordant, we need several auxiliary inequalities.

**Proposition 6.3.2.** *For any  $x \geq 1$ , the following inequality holds.*

1.

$$\text{arccosh}(x) \leq \sqrt{x^2 - 1} \leq x \text{arccosh}(x) \tag{6.26}$$

where the equalities hold only at  $x = 1$ .

2.

$$3x\sqrt{x^2 - 1} \leq \text{arccosh}(x)(2x^2 + 1) \tag{6.27}$$

where the equality holds only at  $x = 1$ .

3.

$$(3x\sqrt{x^2 - 1} - \text{arccosh}(x)(2x^2 + 1))^2 \leq 4(x^2 - 1)(x \text{arccosh}(x))^2 \tag{6.28}$$

where the equality holds only at  $x = 1$ .

**Proof:**

1. Let  $g(x) = \sqrt{x^2 - 1} - \operatorname{arccosh}(x)$  where  $x \geq 1$ . Then it can be calculated that

$$\begin{aligned} g'(x) &= \frac{x}{\sqrt{x^2 - 1}} - \frac{1}{\sqrt{x^2 - 1}} \\ &= \frac{x - 1}{\sqrt{x^2 - 1}}. \end{aligned} \quad (6.29)$$

Therefore, when  $x > 1$ , we obtain

$$g'(x) > 0. \quad (6.30)$$

Thus,  $g(x)$  is monotonically increasing with respect to  $x$  when  $x > 1$ . Since  $g$  is continuous on  $x$  and  $g(1) = 0$ , we have

$$g(x) \geq g(1) = 0. \quad (6.31)$$

Hence, it follows from (6.31) that

$$\operatorname{arccosh}(x) \leq \sqrt{x^2 - 1} \quad (6.32)$$

where the equality holds only at  $x = 1$ .

Similarly, we can prove that

$$\sqrt{x^2 - 1} \leq x \operatorname{arccosh}(x) \quad (6.33)$$

where the equality holds only at  $x = 1$ .

2. Let  $g(x) = 3x\sqrt{x^2 - 1} - \operatorname{arccosh}(x)(2x^2 + 1)$  where  $x \geq 1$ . Then it can be calculated

$$\begin{aligned} g'(x) &= 3\sqrt{x^2 - 1} + \frac{3x^2}{\sqrt{x^2 - 1}} - \frac{2x^2 + 1}{\sqrt{x^2 - 1}} - 4x \operatorname{arccosh}(x) \\ &= 4(\sqrt{x^2 - 1} - x \operatorname{arccosh}(x)). \end{aligned} \quad (6.34)$$

In view of (6.26), we obtain

$$g'(x) < 0 \quad (6.35)$$

where  $x > 1$ .

Thus,  $g(x)$  is monotonically decreasing with respect to  $x$  when  $x > 1$ . Since  $g$  is continuous on  $x$  and  $g(1) = 0$ , we have

$$g(x) \leq g(1) = 0. \quad (6.36)$$

Hence, it follows from (6.41) that

$$3x\sqrt{x^2 - 1} \leq \operatorname{arccosh}(x)(2x^2 + 1) \quad (6.37)$$

where the equality holds only at  $x = 1$ .

3. Let  $g(x) = (3x\sqrt{x^2 - 1} - \operatorname{arccosh}(x)(2x^2 + 1))^2 - 4(x^2 - 1)(x\operatorname{arccosh}(x))^2$  where  $x \geq 1$ . Then it can be calculated

$$\begin{aligned} g'(x) &= 8(3x\sqrt{x^2 - 1} - \operatorname{arccosh}(x)(2x^2 + 1))\sqrt{x^2 - 1} \\ &\quad + \operatorname{arccosh}(x)(2x\operatorname{arccosh}(x) - 4x^2\sqrt{x^2 - 1}). \end{aligned} \quad (6.38)$$

By (6.26), when  $x > 1$ , we obtain

$$2x\operatorname{arccosh}(x) < 4x^2\sqrt{x^2 - 1}. \quad (6.39)$$

In view of (6.27) and (6.39), since  $\operatorname{arccosh}(x) > 0$  when  $x > 1$ , we have

$$g'(x) < 0. \quad (6.40)$$

where  $x > 1$ .

Thus,  $g(x)$  is monotonically decreasing with respect to  $x$  when  $x > 1$ . Since  $g$  is continuous on  $x$  and  $g(1) = 0$ , we have

$$g(x) \leq g(1) = 0. \quad (6.41)$$

Hence, it follows from (6.41) that

$$(3x\sqrt{x^2 - 1} - \operatorname{arccosh}(x)(2x^2 + 1))^2 \leq 4(x^2 - 1)(x\operatorname{arccosh}(x))^2 \quad (6.42)$$

where the equality holds only at  $x = 1$ .

□

Without loss of generality, now we consider the following optimization problem

$$\begin{aligned} \min \quad & f_0(p) = \frac{1}{2}d^2(p, p_0) = \frac{1}{2}\operatorname{arccosh}(p^T Ap_0) \\ \text{s.t.} \quad & p \in I_n \end{aligned} \quad (6.43)$$

where  $p_0 \in I_n$  is given. Similarly, we obtain the first, second and third covariant derivatives of  $f_0$  as follows.

$$\nabla_H f_0(p) = \theta \operatorname{arccosh}(p^T Ap_0) \frac{X^T Ap_0}{\sqrt{(p^T(A)p_0)^2 - 1}}, \quad (6.44)$$

$$\begin{aligned} \nabla_H^2 f_0(p) = & \theta^2 \left[ \frac{(X^T Ap_0)^2}{(p^T Ap_0)^2 - 1} \right. \\ & \left. + \operatorname{arccosh}(p^T Ap_0) \frac{p^T Ap_0 [(p^T Ap_0)^2 - (X^T Ap_0)^2 - 1]}{((p^T Ap_0)^2 - 1)^{\frac{3}{2}}} \right], \end{aligned} \quad (6.45)$$

$$\begin{aligned} \nabla_H^3 f_0(p) = & \theta^3 \left[ \frac{X^T Ap_0}{((p^T Ap_0)^2 - 1)^{\frac{3}{2}}} - \frac{(X^T Ap_0)^2}{((p^T Ap_0)^2 - 1)^{\frac{5}{2}}} \right] \\ & (3p^T Ap_0((p^T Ap_0)^2 - 1)^{\frac{1}{2}} - \operatorname{arccosh}(p^T Ap_0)(2(p^T Ap_0)^2 + 1)), \end{aligned} \quad (6.46)$$

where  $\theta = \sqrt{H^T(-A)H}$  and  $X = H/\theta$ .

**Lemma 10.** *Given any  $p \in I_n$ , and  $H \in T_p I_n$ , the following result holds:*

$$(X^T Ap_0)^2 \leq (p^T Ap_0)^2 - 1 \quad (6.47)$$

where  $\theta = \sqrt{H^T(-A)H}$ ,  $X = H/\theta$  and  $X = 0$  if  $H = 0$ .

**Proof:** For simplicity, let  $B = p^T Ap_0$ . If  $H = 0$ , since  $p^T Ap_0 \geq 1$  by Proposition 6.3.1, we

have

$$B^2 - 1 \geq 0 = (X^T A p_0)^2. \tag{6.48}$$

Otherwise, consider two cases here.

Case one:  $p = p_0$ . Then we have  $B = 1$  and  $X^T A p_0 = 0$ . Consequently, we obtain

$$B^2 - 1 \geq 0 = (X^T A p_0)^2. \tag{6.49}$$

Case two:  $p \neq p_0$ . Then we have  $B > 1$  and  $X^T A p_0 \neq 0$ . Let  $\gamma(t)$  denote the geodesic emanating from  $p$  in the direction  $H$ . Then we have  $\gamma(t) = \text{Exp}_p tH = p \cosh(\theta t) + X \sinh(\theta t)$  where  $\theta = \sqrt{H^T(-A)H}$  and  $X = H/\theta$ . Consider the function  $f_0$  in (6.43). In view of Equation (1.2.2) of Theorem 1.2 in [42], we have

$$\nabla_H^2 f_0(p) \geq \|H\|_p^2. \tag{6.50}$$

Substituting (6.45) into (6.50), we obtain

$$\frac{\sqrt{B^2 - 1} - \text{Barccosh}(B)}{(B^2 - 1)^{\frac{3}{2}}} (X^T A p_0)^2 + \frac{\text{Barccosh}(B)}{\sqrt{(B^2 - 1)}} \geq 1. \tag{6.51}$$

It follows from (6.51) that

$$(X^T A p_0)^2 \leq B^2 - 1. \tag{6.52}$$

This completes the proof of the lemma. □

With the help of Proposition 6.3.2 and Lemma 10, we can show that  $f_0$  in (6.43) is self-concordant.

**Lemma 11.** *The function  $f_0$  in (6.43) is a self-concordant function defined on the  $n$ -dimensional hyperboloid model with the constant  $M_{f_0} = \sqrt{\frac{16}{27}}$ .*

**Proof:** Given a point  $p = p_0$  and a tangent vector  $H \in T_p I_n$ , we have

$$\nabla_H^2 f(p) = \|H\|_p^2, \tag{6.53}$$

$$\nabla_H^3 f(p) = 0. \tag{6.54}$$

Then, it holds

$$|\nabla_H^3 f(p)| \leq \sqrt{\frac{16}{27}} \left( \nabla_H^2 f(p) \right)^{\frac{3}{2}}. \quad (6.55)$$

Otherwise, given a point  $p \in I_n$ ,  $p \neq p_0$  and a nonzero tangent vector  $H \in T_p I_n$ , let  $B = p^T A p_0$  and  $w = X^T A p_0$ . Then we have

$$\begin{aligned} \frac{(\nabla_H^3 f_0(p))^2}{(\nabla_H^2 f_0(p))^3} &= \frac{\left[ \frac{w}{(B^2-1)^{\frac{3}{2}}} - \frac{w^3}{(B^2-1)^{\frac{5}{2}}} \right]^2 \left[ 3B\sqrt{B^2-1} - (2B^2+1)\operatorname{arccosh}(B) \right]^2}{\left[ \frac{\sqrt{B^2-1} - \operatorname{Barccosh}(B)}{(B^2-1)^{\frac{3}{2}}} w^2 + \frac{\operatorname{Barccosh}(B)}{\sqrt{B^2-1}} \right]^3} \\ &= \frac{[3B\sqrt{B^2-1} - (2B^2+1)\operatorname{arccosh}(B)]^2 [(B^2-1)w - w^3]^2}{\sqrt{B^2-1} [(\sqrt{B^2-1} - \operatorname{Barccosh}(B))w^2 + (B^2-1)\operatorname{Barccosh}(B)]^3} \\ &= \frac{[3B\sqrt{B^2-1} - (2B^2+1)\operatorname{arccosh}(B)]^2}{\sqrt{B^2-1} (\operatorname{Barccosh}(B) - \sqrt{B^2-1})^3} \\ &\quad \frac{w^2 (B^2-1-w^2)^2}{\left[ \frac{\operatorname{Barccosh}(B)}{\operatorname{Barccosh}(B) - \sqrt{B^2-1}} (B^2-1) - w^2 \right]^3} \end{aligned} \quad (6.56)$$

Let  $t = w^2$ . Then we have the range of  $t$  by Lemma 10

$$0 \leq t \leq B^2 - 1. \quad (6.57)$$

For simplicity, let  $a = B^2 - 1$  and  $b = \frac{\operatorname{Barccosh}(B)}{\operatorname{Barccosh}(B) - \sqrt{B^2-1}} (B^2 - 1)$ . By (6.26), we have

$$b > a > 0. \quad (6.58)$$

Now we consider the following auxiliary function

$$\pi(t) = \frac{t(a-t)^2}{(b-t)^3} \quad (6.59)$$

where  $t$  satisfies (6.57).

By computation, we obtain the first and second order derivatives of  $\pi$

$$\pi'(t) = \frac{(a-t)[(2a-3b)t+ab]}{(b-t)^4} \tag{6.60}$$

$$\pi''(t) = \frac{(a-t)(2a-3b)(b-t) - ((2a-3b)t+ab)(b-4a+3t)}{(b-t)^5}. \tag{6.61}$$

Setting  $\pi'(t) = 0$ , it is easy to compute critical points  $t^*$  of  $\pi$

$$\begin{aligned} t^* &= a, \\ t^* &= \frac{ab}{3b-2a}. \end{aligned}$$

It is easy to see that these two critical points satisfy (6.57). Then substituting  $t^*$  into (6.61), we obtain

$$\pi''(a) = \frac{3a}{(b-a)^3} \tag{6.62}$$

$$\pi''\left(\frac{ab}{3b-2a}\right) = \frac{-3a(b-a)}{\left(b-\frac{ab}{3b-2a}\right)^4} \tag{6.63}$$

By (6.58), we obtain

$$\pi''(a) > 0, \tag{6.64}$$

$$\pi''\left(\frac{ab}{3b-2a}\right) < 0. \tag{6.65}$$

Since  $\pi(t)$  is continuous on  $t$ , it follows from (6.65) that  $\pi$  achieves its maximum at  $t = \frac{ab}{3b-2a}$ . By computation, we can get such maximum

$$\pi\left(\frac{ab}{3b-2a}\right) = \frac{4}{27} \frac{a^3}{b^2(b-a)}. \tag{6.66}$$

Combining (6.56) and (6.66), we obtain

$$\frac{(\nabla_H^3 f_0(p))^2}{(\nabla_H^2 f_0(p))^3} \leq \frac{4}{27} \frac{[3B\sqrt{B^2-1} - (2B^2+1)\operatorname{arccosh}(B)]^2}{(B^2-1)(B\operatorname{arccosh}(B))^2} \tag{6.67}$$

In view of (6.28), it follows from (6.67) that

$$\frac{(\nabla_H^3 f_0(p))^2}{(\nabla_H^2 f_0(p))^3} \leq \frac{16}{27}. \quad (6.68)$$

As a consequence, we conclude that  $f_0$  is self-concordant with  $M_{f_0} = \sqrt{\frac{16}{27}}$ .  $\square$

**Lemma 12.** *The function  $f$  in (6.20) is a self-concordant function defined on the  $n$ -dimensional hyperboloid model with the constant  $M_f = \sqrt{\frac{16}{27}}$ .*

**Proof:** In view of Proposition 4.3.2 and Lemma 11, it is easy to conclude that  $f$  in (6.20) is self-concordant with the constant  $M_f = \sqrt{\frac{16}{27}}$ .  $\square$

Since the function  $f$  in (6.20) is self-concordant, we are able to apply our damped Newton and conjugate gradient algorithms to find the minimum of  $f$  on  $I_n$ . In particular, we take  $n = 19$ . First, we consider building up the damped Newton algorithm for (6.20).

In view of (4.49), given a point  $p \in I_n$ , the Newton direction  $X_N$  of  $f$  at  $p$  is the unique tangent vector determined by

$$\text{Hess}_p f(X_N, H) = \langle -\text{grad}_p f, H \rangle_p \quad \text{for all tangent vectors } H \in T_p I_n. \quad (6.69)$$

By (4.6) and (6.22), we obtain

$$\begin{aligned} \text{Hess}_p f(X_N, H) = & \sum_{i=1}^m \left[ \frac{1}{(p^T A p_i)^2 - 1} X_N^T A p_i H^T A p_i + \frac{\text{arccosh}(p^T A p_i) p^T A p_i}{\sqrt{(p^T A p_i)^2 - 1}} \right. \\ & \left. X_N^T (-A) H - \frac{\text{arccosh}(p^T A p_i) p^T A p_i}{((p^T A p_i)^2 - 1)^{\frac{3}{2}}} \right]. \end{aligned} \quad (6.70)$$

Then, combining (6.69) and (6.70), we get

$$\begin{aligned} \sum_{i=1}^m \left[ \left( \frac{1}{(p^T A p_i)^2 - 1} - \frac{\text{arccosh}(p^T A p_i) p^T A p_i}{((p^T A p_i)^2 - 1)^{\frac{3}{2}}} \right) \left( X_N^T A p_i p_i^T A p p - p_i p_i^T A X_N \right) \right. \\ \left. + \left( \frac{\text{arccosh}(p^T A p_i) p^T A p_i}{\sqrt{(p^T A p_i)^2 - 1}} \right) X_N = -\text{grad}_p f \right] \end{aligned} \quad (6.71)$$

Thus, the solution of the linear system (6.71) is the Newton direction of  $f$  at  $p$  which can be used for the damped Newton method. To solve this linear system, we adopt the linear conjugate gradient method modified directly from Golub and Van Loan [29] as follows.



**Algorithm 16. (Linear Conjugate Gradient Algorithm)**

step 0: Given a point  $p \in I_n$ , compute  $R_0 = -\text{grad}_p f$ , and set  $P_0 = R_0$  and  $k = 0$ .

step k: If  $k = n$ , then terminate. If  $k = 1$ , set

$$P_1 = R_0. \tag{6.72}$$

Otherwise, compute

$$Q_k = \sum_{i=1}^m \left[ \left( \frac{1}{(p^T A p_i)^2 - 1} - \frac{\text{arccosh}(p^T A p_i) p^T A p_i}{((p^T A p_i)^2 - 1)^{\frac{3}{2}}} \right) \left( P_k^T A p_i p_i^T A p p - p_i p_i^T A P_k \right) + \left( \frac{\text{arccosh}(p^T A p_i) p^T A p_i}{\sqrt{(p^T A p_i)^2 - 1}} \right) P_k, \tag{6.73}$$

$$\beta_{k-1} = \frac{\langle R_{k-1}, R_{k-1} \rangle_p}{\langle R_{k-2}, R_{k-2} \rangle_p}, \tag{6.74}$$

$$P_k = R_{k-1} + \beta_{k-1} P_{k-1}, \tag{6.75}$$

$$\alpha_k = \frac{\langle R_{k-1}, R_{k-1} \rangle_p}{\langle P_k, Q_k \rangle_p}, \tag{6.76}$$

$$X_k = X_{k-1} + \alpha_k P_k, \tag{6.77}$$

$$R_k = R_{k-1} - \alpha_k Q_k. \tag{6.78}$$

Increment  $k$  and repeat until  $k = n$ .

After  $n$  steps, Algorithm 16 generates a finite sequence  $\{X_1, \dots, X_n\}$  and the Newton direction of  $f$  at  $p$  is given by

$$X_N = X_n. \tag{6.79}$$

Now, we are able to construct the damped Newton method for solving (6.20).

**Algorithm 17.** step 0: Randomly generate an initial point  $p_0 \in I_n$  and compute  $\text{grad}_{p_0} f$ . Set  $k = 0$ .

step k: If  $\text{grad}_{p_k} f = 0$ , then terminate. Otherwise, compute  $X_{N_{p_k}}$  using Algorithm 16. Then,

compute

$$\lambda_k = \nabla_{X_{N_{p_k}}}^2 f(p_k), \quad (6.80)$$

$$t_k = \frac{1}{1 + \lambda_k}, \quad (6.81)$$

$$p_{k+1} = p_k \cosh \left( \sqrt{X_{N_{p_k}}(-A)X_{N_{p_k}}} t_k \right) + \frac{X_{N_{p_k}}}{\sqrt{X_{N_{p_k}}(-A)X_{N_{p_k}}}} \sinh \left( \sqrt{X_{N_{p_k}}(-A)X_{N_{p_k}}} t_k \right), \quad (6.82)$$

where  $\nabla_{X_{N_{p_k}}}^2 f(p_k)$  is from (6.22).

Increment  $k$  and repeat until convergence.

On the other hand, the damped conjugate gradient method for solving (6.20) is given as follows.

**Algorithm 18. (Conjugate Gradient Algorithm)**

step 0: Select an initial point  $p_0 \in I_n$ , compute  $H_0 = G_0 = -(\text{grad}_{p_0} f)$  by (6.86), and set  $k = 0$ .

step k: If  $\text{grad}_{p_k} f = 0$ , then terminate. Otherwise, compute

$$\lambda_k = \frac{-\nabla_{H_k} f(p_k)}{\sqrt{\nabla_{H_k}^2 f(p_k)}}, \quad (6.83)$$

$$t_k = \frac{\lambda_k}{(1 + \lambda_k) \sqrt{\nabla_{H_k}^2 f(p_k)}}, \quad (6.84)$$

$$p_{k+1} = p_k \cosh \left( \sqrt{H_k(-A)H_k} t_k \right) + \frac{H_k}{\sqrt{H_k(-A)H_k}} \sinh \left( \sqrt{H_k(-A)H_k} t_k \right), \quad (6.85)$$

$$G_{k+1} = -\text{grad}_{p_{k+1}} f, \quad (6.86)$$

$$\gamma_{k+1} = \frac{\langle G_{k+1}, G_{k+1} \rangle_{p_{k+1}}}{\langle G_k, H_k \rangle_{p_k}}, \quad (6.87)$$

$$H_{k+1} = G_{k+1} + \gamma_{k+1} \tau_{p_k p_{k+1}} H_k, \quad (6.88)$$

where  $\tau_{p_k p_{k+1}}$  is the parallel transport with respect to the geodesic from  $p_k$  to  $p_{k+1}$ . If  $k + 1 \bmod n - 1 = 0$ , set  $H_{k+1} = G_{k+1}$ . Increment  $k$  and repeat until convergence.

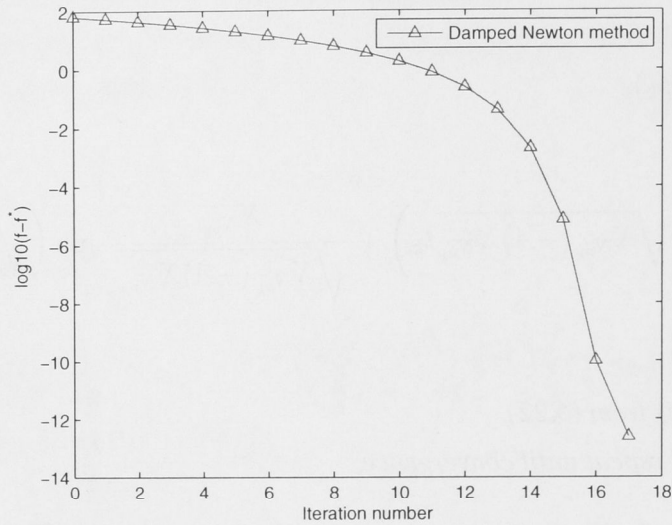


Figure 6.4: The result of the the damped Newton method for the self-concordant function defined on the Hyperboloid model

Figure 6.5 illustrates the result of the damped Newton method and conjugate gradient method on function  $f$  in (6.20). Table 6.2 shows the simulation time and accuracy using the damped conjugate gradient and Newton methods. From Figure 6.5, it can be see that the damped Newton method converges to the minimum quadratically, but the damped conjugate gradient method gets to the minimum super-linearly. Although the damped conjugate gradient method requires more steps, due to avoid computing the linear system, it costs less time than the damped Newton method, seen from Table 6.2.

algorithm	time(second)	accuracy
damped conjugate gradient method	0.062	$10^{-5}$
damped Newton method	0.313	$10^{-5}$

Table 6.2: Simulation time and accuracy

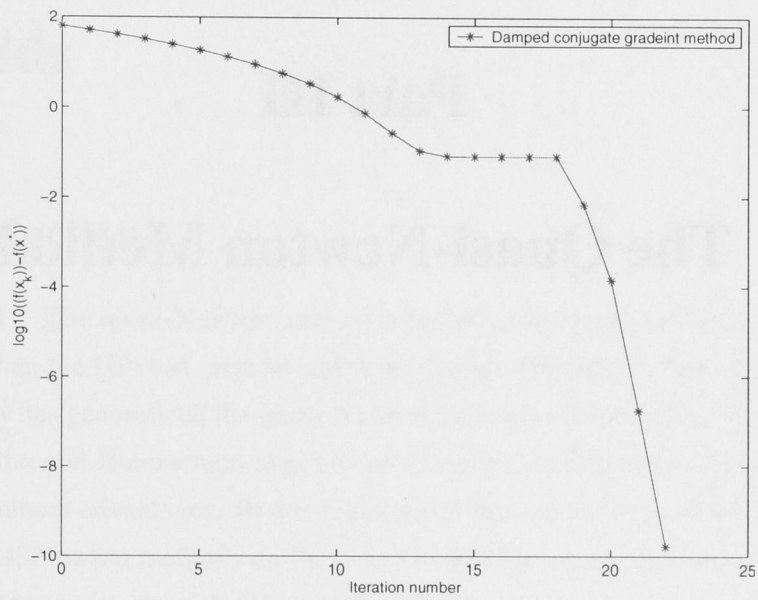


Figure 6.5: The result of the damped conjugate gradient method for the self-concordant function defined on the Hyperboloid model

## **Part III**

# **The Quasi-Newton Method**

# Chapter 7

## A Quasi-Newton Method On Smooth Manifolds

### 7.1 Introduction

**Background** The quasi-Newton method is preferred by engineers since it has lower computational cost than the Newton method and super-linear convergence rate. As we mentioned in Chapter 1, Gabay has generalized the quasi-Newton method to Riemannian manifolds. However, in recent years, the non-Riemannian methods on smooth manifolds have drawn more attentions due to their prominent advantages. To our best knowledge, we are not aware of any publications generalizing quasi-Newton methods on the non-Riemannian cases. Hence, we are motivated to develop a non-Riemannian quasi-Newton method for the optimization on smooth manifolds.

**Our work** In this chapter, we present a numerical non-Riemannian quasi-Newton method on smooth manifolds. This method is proved to converge to the local minimum of the cost function. The super-linear convergence rate was demonstrated by simulation results on the Grassmann manifold.

**Chapter outline** The rest of this chapter is organized as follows. We first give the preliminaries associated with smooth manifolds in Section 7.2. Then a non-Riemannian quasi-Newton method is presented for the optimization on smooth manifolds and it is proved to converge to the minimum of the cost function in Section 7.3. In Section 7.4 simulation results show our method has the super-linear convergence rate. This is followed by the conclusions in Section 7.5.

## 7.2 Preliminaries

In this section, we briefly introduce notations and concepts of local parameterization for smooth manifolds. For more details, see [46]. Let  $M$  be a smooth  $n$ -dimensional manifold. Then for every point  $p \in M$  there exists a smooth map

$$\psi_p : \Omega_p \subset \mathbb{R}^n \rightarrow U_p \subset M, \quad \psi_p(0) = p \quad (7.1)$$

where  $U_p$  is an open neighborhood of  $p$  and  $\Omega_p$  an open subset of  $\mathbb{R}^n$  around 0. Such a map is called a local parameterization around  $p$ . We use the triple  $(\psi_p, U_p, \Omega_p)$  to denote the local parameterization around  $p$ . Let  $f : M \rightarrow \mathbb{R}$  be a real-valued smooth function defined on a smooth manifold  $M$ . Given a point  $p \in M$  and a local parameterization  $(\psi_p, U_p, \Omega_p)$  around  $p$ , then the composition of  $f$  and  $\psi_p$  is called the local cost function and denoted by  $f \circ \psi_p$ . For simplification, let  $g_p : \Omega_p \rightarrow \mathbb{R}$  denote such local cost function satisfying

$$g_p(x) = f(\psi_p(x)). \quad (7.2)$$

## 7.3 Quasi-Newton Method On Smooth Manifolds

In this section, we develop a quasi-Newton method for the optimization of smooth functions defined on smooth manifolds.

Before we give our method, we review the BFGS method [64] for the optimization in Euclidean space. Consider the following optimization problem

$$\min_{x \in \mathbb{R}^n} f : \mathbb{R}^n \rightarrow \mathbb{R} : x \rightarrow f(x). \quad (7.3)$$

Let  $f'(x)$  denote the first derivative of  $f$  at  $x$ . Then the BFGS algorithm for solving (7.3) goes as follows.

### Algorithm 19. (BFGS Algorithm in Euclidean Space)

step 0: Select an initial point  $x_0$  and set  $B_0 = I_n$  where  $I_n$  denotes the  $n$ -dimensional identity matrix and  $k = 0$ . Compute  $H_0 = -B_0^{-1}f'(x_0)$ .

step k: If  $f'(x_k) = 0$ , then terminate. Otherwise, compute the step-size  $\lambda_{k+1}$  as follows.

$$\lambda_{k+1} = 2^{-l} \quad (7.4)$$

where  $l$  is the smallest positive integer such that

$$f(\lambda_{k+1}H_k) \leq f(x_k) + c_1\lambda_{k+1}H_k^T f'(x_k), \quad (7.5)$$

$$H_k^T f'(\lambda_{k+1}H_k) \geq c_2H_k^T f'(x_k), \quad (7.6)$$

where  $0 < c_1 < c_2 < 1$ .

Set

$$x_{k+1} = x_k + \lambda_{k+1}H_k, \quad (7.7)$$

$$s_k = x_{k+1} - x_k, \quad (7.8)$$

$$y_k = f'(x_{k+1}) - f'(x_k), \quad (7.9)$$

$$B_{k+1} = B_k + \frac{y_k y_k^T}{\langle y_k, s_k \rangle} + \frac{B_k s_k (B_k s_k)^T}{\langle B_k s_k, s_k \rangle}, \quad (7.10)$$

$$H_{k+1} = -B_{k+1}^{-1} f'(x_{k+1}). \quad (7.11)$$

Increment  $k$  and repeat until convergence.

In Algorithm 19, Equation (7.5) and (7.6) form the Wolfe conditions [61] used to determine the appropriate step-size in the line search. Equation (7.10) provides an approximation to the Hessian of  $f$  over successive iterations using only its first order information. Moreover, Algorithm 19 turns out to have super-linear convergence in [64].

Now, we consider to generalize the BFGS method for the optimization on smooth manifolds. Let  $M$  be a smooth  $n$ -dimensional manifold. For  $p \in M$ , let the triple  $(\psi_p, U_p, \Omega_p)$  be the local parameterization around  $p$ . Consider the following optimization problem

$$\min_{p \in M} f : M \rightarrow \mathbb{R} : p \rightarrow f(p). \quad (7.12)$$

For any  $p \in M$  and the local parameterization  $(\psi_p, U_p, \Omega_p)$  around  $p$ , let  $g_p(x)$  denote the local cost function of  $f$ . Moreover, let  $g'_p(x)$  and  $g''_p(x)$  denote the first and second derivatives of  $g_p(x)$  with respect to  $x$  respectively. Then a quasi-Newton method is developed for solving (7.12) as follows.

**Algorithm 20. (Quasi-Newton Algorithm)**

step 0: Select an initial point  $p_0 \in M$  and set  $B_0 = I_n$  where  $I_n$  denotes  $n$ -dimensional identity matrix and  $k = 1$ . Compute  $H_0 = -B_0^{-1}g'_{p_0}(0)$ .



step k: If  $g'_{p_k}(0) = 0$ , then terminate. Otherwise, compute the step-size  $\lambda_{k+1}$  as follows.

$$\lambda_{k+1} = 2^{-l} \tag{7.13}$$

where  $l$  is the smallest positive integer such that  $\lambda_{k+1}H_k \in \Omega_{p_k}$

$$g_{p_k}(\lambda_{k+1}H_k) \leq g_{p_k}(0) + c_1\lambda_{k+1}H_k^T g'_{p_k}(0), \tag{7.14}$$

$$H_k^T g'_{p_k}(\lambda_{k+1}H_k) \geq c_2 H_k^T g'_{p_k}(0), \tag{7.15}$$

where  $0 < c_1 < c_2 < 1$ .

Set

$$x_{k+1} = \lambda_{k+1}H_k, \tag{7.16}$$

$$s_k = x_{k+1}, \tag{7.17}$$

$$y_k = g'_{p_k}(x_{k+1}) - g'_{p_k}(0), \tag{7.18}$$

$$B_{k+1} = B_k + \frac{y_k y_k^T}{\langle y_k, s_k \rangle} + \frac{B_k s_k (B_k s_k)^T}{\langle B_k s_k, s_k \rangle}, \tag{7.19}$$

$$p_{k+1} = \psi_{p_k}(x_{k+1}), \tag{7.20}$$

$$H_{k+1} = -B_{k+1}^{-1} g'_{p_{k+1}}(0). \tag{7.21}$$

Increment  $k$  and repeat until convergence.

In Algorithm 21, Equation (7.14) and (7.15) form the Wolfe conditions [61] used to determine the appropriate step-size in the line search. They facilitate the proof of the convergence of Algorithm 21.

Now, we consider proving the convergence of Algorithm 21. Before we do that, we need the following two assumptions.

**Assumption 4.** Consider the smooth function  $f$  in (7.12). There exist

1. at least one minimum of  $f$ ;
2. a point  $p_0 \in M$  which defines a sub-level set  $K = \{p \in M \mid f(p) \leq f(p_0)\}$ ;
3. local parameterization  $\psi_p$  for all  $p \in K$ , defined in (7.1);

such that for all  $p \in K$ ,  $x \in \Omega_p$  and any  $u \in \mathbb{R}^n$ ,

$$\alpha \|u\|^2 \leq u^T g_{p_k}''(x)u \leq \beta \|u\|^2 \quad (7.22)$$

where  $\beta, \alpha > 0$  are constants.

**Assumption 5.** The local parameterization  $\psi_{p_k}$  used in Algorithm 21 does not involve the step number  $k$ .

Assumption 4 implies that for each  $p \in K$ , we can find a local parameterization such that the local cost function determined by it is convex on its local Euclidean space. For example, consider the convex function defined on Riemannian manifolds in [70]. Viewing the geodesics as one kind of local parameterization, then according to the definition of convex functions on Riemannian manifolds, we can find a neighborhood around the local minimum such that the local cost functions are convex.

Assumption 5 rules out the case that the vanishing of the gradient of the local cost function is an artifact due to the “shrinking“ of the local parameterization of  $\psi_{p_k}$ .

Equation (7.19) gives the formula which approximates the Hessian of the local cost function. It is worth noting that Equation (7.19) has the same expression as the BFGS update formula (7.10). Since the local cost function satisfies (7.22), according to Theorem 1.6.7 in [64], we conclude that if  $B_k$  is symmetric positive-definite, then  $B_{k+1}$ , given by (7.19), is also symmetric positive-definite. Furthermore, since the initial matrix  $B_0$  is identity, it is easy to see the matrix in  $\{B_k\}$  generated by Algorithm (21) is always symmetric positive-definite. As a result, at each step, Algorithm (21) yields a descent step. In addition, in view of Equation (28f) and (29) in [64], the trace and determinant of  $B_{k+1}$  are given by

$$\text{tr}(B_{k+1}) = \text{tr}(B_k) + \frac{\|y_k\|^2}{\langle y_k, s_k \rangle} - \frac{\|B_k s_k\|}{\langle B_k s_k, s_k \rangle}, \quad (7.23)$$

$$\det(B_{k+1}) = \det(B_k) \frac{\langle y_k, s_k \rangle}{\langle B_k s_k, s_k \rangle}. \quad (7.24)$$

To show the global convergence of Algorithm 21, we follow the frameworks in [8]. Initially, motivated by (7.23) and (7.24), we obtain the following lemmas, which lead to the convergence result.

**Lemma 13.** Let the cost function  $f : M \rightarrow \mathbb{R}$  in (7.12) satisfy Assumption 4. Then given

$p_0 \in K$ , applying Algorithm 21 to minimize  $f$ , we have

$$\frac{\|y_k\|^2}{\langle y_k, s_k \rangle} \leq \beta \quad (7.25)$$

where  $s_k$  and  $y_k$  are defined by (7.66) and (7.18) respectively.

**Proof:** Consider the local cost function  $g_{p_k}$  defined on  $\Omega_{p_k}$ . Since  $x_{k+1} \in \Omega_{p_k}$ , then we have

$$\begin{aligned} \int_0^1 g_{p_k}''(ts_k) dt s_k &= g_{p_k}'(x_{k+1}) - g_{p_k}'(0) \\ &= y_k. \end{aligned} \quad (7.26)$$

Let  $\bar{B}_{k+1} = \int_0^1 g_{p_k}''(ts_k) dt$ . Then it follows from (7.26) that

$$\bar{B}_{k+1} s_k = y_k. \quad (7.27)$$

In addition, for any  $v \neq 0 \in \mathbb{R}^n$ , we have

$$\begin{aligned} v^T \bar{B}_{k+1} v &= v^T \int_0^1 g_{p_k}''(ts_k) dt v \\ &\leq \int_0^1 \beta \|v\|^2 dt \\ &= \beta \|v\|^2 \end{aligned} \quad (7.28)$$

where the inequality comes from (7.22).

Then it follows from (7.28) that

$$\frac{v^T \bar{B}_{k+1} v}{\|v\|^2} \leq \beta. \quad (7.29)$$

Since  $\bar{B}_{k+1}$  is symmetric and positive definite, let  $v = \bar{B}_{k+1}^{-\frac{1}{2}} y_k$ . Combining (7.27) and (7.29), we obtain

$$\frac{\|y_k\|^2}{\langle y_k, s_k \rangle} \leq \beta. \quad (7.30)$$

□

**Lemma 14.** *Let the cost function  $f : M \rightarrow \mathbb{R}$  in (7.12) satisfy Assumption 4. Then given  $p_0 \in K$ , applying Algorithm 21 to minimize  $f$ , there exists a constant  $A > 0$  such that*

$$\operatorname{tr}(B_{k+1}) \leq Ak. \quad (7.31)$$

**Proof:** By (7.23), it is easy to see that

$$\operatorname{tr}(B_{i+1}) \leq \operatorname{tr}(B_i) + \frac{\|y_i\|^2}{\langle y_i, s_i \rangle}. \quad (7.32)$$

Summing up the inequalities (7.32) for  $i = 0, 1, \dots, k$ , we have

$$\begin{aligned} \operatorname{tr} B_{k+1} &\leq \operatorname{tr}(B_0) + \sum_{i=0}^k \frac{\|y_i\|^2}{\langle y_i, s_i \rangle} \\ &\leq \operatorname{tr}(B_0) + \beta k \\ &\leq Ak \end{aligned} \quad (7.33)$$

where  $A = \operatorname{tr}(B_0) + \beta$ .

This completes the proof of the lemma.  $\square$

**Lemma 15.** *Let the cost function  $f : M \rightarrow \mathbb{R}$  in (7.12) satisfy Assumption 4. Then given  $p_0 \in K$ , applying Algorithm 21 to minimize  $f$ , there exists a constant  $L > 0$  such that*

$$\det(B_{k+1}) \geq \beta \prod_{i=0}^k \frac{\|g'_{p_i}(0)\|^2 \langle y_i, s_i \rangle}{L \langle g'_{p_i}(0), s_i \rangle}. \quad (7.34)$$

**Proof:** Multiplying (7.24) from  $i = 1, \dots, k$ , we have

$$\det(B_{k+1}) = \det(B_0) \prod_{i=0}^k \frac{\langle y_i, s_i \rangle}{\langle B_i s_i, s_i \rangle}. \quad (7.35)$$

It follow from (7.23) and (7.30) that

$$\begin{aligned} \operatorname{tr}(B_{i+1}) - \operatorname{tr}(B_i) + \frac{\|B_i s_i\|}{\langle B_i s_i, s_i \rangle} &= \frac{\|y_i\|^2}{\langle y_i, s_i \rangle} \\ &\leq \beta. \end{aligned} \quad (7.36)$$

Then, by (7.36), we have

$$\beta + \operatorname{tr}(B_i) - \operatorname{tr}(B_{i+1}) \geq \frac{\|B_i s_i\|^2}{\langle B_i s_i, s_i \rangle}. \quad (7.37)$$

Summing up (7.37) from  $i = 0, 1, \dots, k$ , we obtain

$$\begin{aligned} \sum_{i=0}^k \frac{\|B_i s_i\|^2}{\langle B_i s_i, s_i \rangle} &\leq k\beta + \operatorname{tr} B_0 - \operatorname{tr}(B_{k+1}) \\ &\leq k\beta + \operatorname{tr}(B_0). \end{aligned} \quad (7.38)$$

Let  $L = \beta + \operatorname{tr}(B_0)$ . Then it follows from (7.38) that

$$\frac{1}{k+1} \sum_{i=0}^k \frac{\|B_i s_i\|^2}{\langle B_i s_i, s_i \rangle} \leq L. \quad (7.39)$$

Recall that given  $k$  positive numbers  $z_1, \dots, z_k$ , then the arithmetic mean is greater than the geometric mean. That is

$$\left( \frac{1}{k} \sum_{i=1}^k a_i \right)^k \geq \prod_{i=1}^k a_i. \quad (7.40)$$

Therefore, by (7.39), we obtain

$$L^{k+1} \geq \prod_{i=0}^k \frac{\|B_i s_i\|^2}{\langle B_i s_i, s_i \rangle}. \quad (7.41)$$

Multiplying (7.35) and (7.41), we have

$$L^{k+1} \det(B_{k+1}) \geq \prod_{i=0}^k \frac{\|B_i s_i\|^2 \langle y_i, s_i \rangle}{\langle B_i s_i, s_i \rangle^2}. \quad (7.42)$$

In view of (7.65), (7.66) and (7.21), we get

$$B_i s_i = -\lambda_{i+1} g'_{p_i}(0). \quad (7.43)$$

Substituting (7.43) into (7.42), we obtain

$$L^{k+1} \det(B_{k+1}) \geq \beta \prod_{i=0}^k \frac{\|g'_{p_i}(0)\|^2 \langle y_i, s_i \rangle}{\langle g'_{p_i}(0), s_i \rangle}. \quad (7.44)$$

Thus, it follow from (7.44) that

$$\det(B_{k+1}) \geq \beta \prod_{i=0}^k \frac{\|g'_{p_i}(0)\|^2 \langle y_i, s_i \rangle}{L \langle g'_{p_i}(0), s_i \rangle}. \quad (7.45)$$

As a result, this lemma follows.  $\square$

The following theorem show that Algorithm 21 converges to the minimum of  $f$  in (7.12) when it satisfies Assumption (4).

**Theorem 7.3.1.** *Let the cost function  $f : M \rightarrow \mathbb{R}$  in (7.12) satisfy Assumption 4. Then given  $p_0 \in K$ , applying Algorithm 21 to minimize  $f$ , it holds*

$$\liminf_{k \rightarrow \infty} \|g'_{p_k}(0)\| = 0. \quad (7.46)$$

**Proof:** By the definition,

$$\begin{aligned} \langle y_k, s_k \rangle &= \langle g'_{p_k}(x_{k+1}) - g'_{p_k}(0), s_k \rangle \\ &= \langle g'_{p_k}(x_{k+1}), s_k \rangle - \langle g'_{p_k}(0), s_k \rangle. \end{aligned} \quad (7.47)$$

In view of (7.15), it follows from (7.47) that

$$\langle y_k, s_k \rangle \geq (c_2 - 1) \langle g'_{p_k}(0), s_k \rangle. \quad (7.48)$$

By (7.14), we have

$$g_{p_k}(x_{k+1}) - g_{p_k}(0) \leq c_1 \langle g'_{p_k}(0), s_k \rangle. \quad (7.49)$$

Substituting (7.48) and (7.49) into (7.34), we obtain

$$\begin{aligned} \det(B_{k+1}) &\geq \beta \prod_{i=0}^k \frac{c_1 \|g'_{p_i}(0)\|^2 (1 - c_2)}{L(g_{p_k}(x_{k+1}) - g_{p_k}(0))} \\ &= \beta \theta^{k+1} \frac{\|g'_{p_i}(0)\|^2}{g_{p_k}(x_{k+1}) - g_{p_k}(0)} \end{aligned} \quad (7.50)$$

where  $\theta = \frac{c_1(1-c_2)}{L}$ .

Using the original cost function  $f$ , by (7.50) and (7.40), we have

$$\begin{aligned} \det(B_{k+1}) &\geq \beta \theta^{k+1} \frac{\prod_{i=0}^k \|g'_{p_i}(0)\|^2}{\prod_{i=0}^k (f(p_{i+1}) - f(p_i))} \\ &\geq \beta \theta^{k+1} \frac{\prod_{i=0}^k \|g'_{p_i}(0)\|^2}{\left(\frac{1}{k+1} \sum_{i=0}^k (f(p_{i+1}) - f(p_i))\right)^k} \\ &\geq \beta \frac{(L(k+1))^{k+1}}{(f(p_0) - f(p^*))^{k+1}} \prod_{i=0}^k \|g'_{p_i}(0)\|^2 \\ &\geq \beta (S(k+1))^{k+1} \prod_{i=0}^k \|g'_{p_i}(0)\|^2 \end{aligned} \quad (7.51)$$

where  $S = \frac{L}{f(p_0) - f(p^*)}$  and  $p^*$  is the local minimum.

Applying (7.40) again, by (7.51) we have

$$\begin{aligned} \beta (S(k+1))^{k+1} \prod_{i=0}^k \|g'_{p_i}(0)\|^2 &\leq \det(B_{k+1}) \\ &\leq \left(\frac{1}{n} \text{tr}(B_k)\right) \\ &\leq \left(\frac{1}{n} A(k+1)\right)^n \\ &\leq N(k+1)^n \end{aligned} \quad (7.52)$$

where the second inequality comes from Lemma (14) and  $N = \left(\frac{A}{n}\right)^n$ .

Assume that (7.46) does not hold. That is there exists an  $\epsilon > 0$  such that for all  $k$

$$\|g'_{p_k}(0)\|^2 \geq \epsilon > 0. \quad (7.53)$$

As a result of (7.53), in view of (7.52), we obtain

$$\beta(S\epsilon(k+1))^{k+1} \leq N(k+1)^n \tag{7.54}$$

which is impossible when  $k \rightarrow \infty$  due to an exponential growing faster than a polynomial.

Consequently, this theorem follows. □

## 7.4 Numerical Example

In this section, we apply our quasi-Newton algorithm to find the dominant eigenspace of a real symmetric matrix.

Let  $A \in \mathbb{R}^{n \times n}$  be a symmetric matrix, i.e.  $A = A^T$ . Let  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p \geq \dots \geq \lambda_n$  be the eigenvalues of  $A$ . Our aim is to find the subspace of  $\mathbb{R}^n$  spanned by the eigenvectors corresponding to the  $p$  largest eigenvalues  $\lambda_1, \dots, \lambda_k$ . This subspace is called the dominant eigenspace of  $A$ . To solve this problem, the natural way is to consider minimizing a cost function defined on the Grassmann manifold.

Recall that the Grassmann manifold  $Gr(n, p)$  is defined as the set of all  $p$ -dimensional subspaces of  $\mathbb{R}^n$ . Let  $St(n, p)$  denote the Stiefel manifold defined as all matrices  $X \in \mathbb{R}^{n \times p}$  satisfying  $X^T X = I_p$  where  $I_p$  is the  $p$ -dimensional identity matrix. Let  $X \in St(n, p)$  and  $[X]$  denote the subspace spanned by the columns of  $X$ . Then we have  $[X] \in Gr(n, p)$ . The Grassmann manifold  $Gr(n, p)$  can be thought of as a quotient manifold of the Stiefel manifold. This is explained as follows. Given two points  $X, Y \in St(n, p)$ , we say  $X$  and  $Y$  are equivalent, denoted by  $X \equiv Y$  if there exists a unitary matrix  $Q$  such that  $Y = XQ$ . It is easy to see that  $X \equiv Y$  if and only if  $[X] = [Y]$ . Thus, there exists a one-to-one correspondence between points on the Grassmann manifold  $Gr(n, p)$  and equivalence classes of  $St(n, p)$ . Given a point  $X \in St(n, p)$ , the tangent space  $T_{[X]}Gr(n, p)$  at  $[X] \in Gr(n, p)$  is given by

$$T_{[X]}Gr(n, p) = \{Z \in \mathbb{R}^{n \times p} : Z = X_{\perp} B, \quad B \in \mathbb{R}^{(n-p) \times p}\} \tag{7.55}$$

where  $X_{\perp} \in \mathbb{R}^{(n-p) \times p}$  means the orthogonal complement of  $X$  and satisfies  $[X \quad X_{\perp}]^T [X \quad X_{\perp}] = I_n$  where  $I_n$  is the  $n$ -dimensional identity matrix.

In [53], Manton proposed a general framework for optimizing cost functions defined on manifolds. As an example of how to apply the framework, the same paper considered local param-



parameterizations on the Grassmann manifold based on projections. Let  $X \in \mathbb{R}^{n \times p}$  be a matrix with the rank  $p$ . Then the projection operator  $\pi : \mathbb{R}^{n \times p} \rightarrow Gr(n, p)$  onto the Grassmann manifold  $Gr(n, p)$  is defined by

$$\pi(X) = \left[ \arg \min_{Q \in St(n, p)} \|X - Q\|_F \right]. \quad (7.56)$$

Then, given  $X \in St(n, p)$  and  $X_\perp$  - the orthogonal complement of  $X$ , in view of (7.55) and (7.56), the local parameterization  $\psi_{[X]} : \mathbb{R}^{(n-p) \times p} \rightarrow Gr(n, p)$  around  $[X] \in Gr(n, p)$  is given by

$$\psi_{[X]}(B) = \pi(X + X_\perp B). \quad (7.57)$$

Consider the following optimization problem:

$$\min f : Gr(n, p) \rightarrow \mathbb{R}, [X] \mapsto -\text{tr}(X^T A X) \quad (7.58)$$

where  $X \in St(n, p)$  and  $A = A^T$ . The solution of (7.58) is the dominant eigenspace of  $A$ .

In view of (7.57), the local cost function  $g_{[X]} : \mathbb{R}^{(n-p) \times p} \rightarrow \mathbb{R}$  around  $[X] \in Gr(n, p)$  is determined by

$$g_{[X]}(B) = f \circ \psi_{[X]}(B) = f(\pi(X + X_\perp B)). \quad (7.59)$$

By Proposition 22 in [53], an easy computation shows that

$$g'_{[X]}(0) = -X_\perp^T A X, \quad (7.60)$$

$$g'_{[X]}(B) = -X_\perp^T A (X + X_\perp B). \quad (7.61)$$

Let  $\text{vec}(X)$  denote the vector by stacking the columns of the matrix  $X$  into a vector. Then, the proposed quasi-Newton method for (7.58) goes as follows.

**Algorithm 21. (Quasi-Newton Algorithm)**

step 0: Select an initial point  $X_0 \in St(n, p)$  and set  $G_0 = I_{(n-p)p}$  where  $I_{(n-p)p}$  denotes  $(n-p)p$ -dimensional identity matrix and  $k = 1$ . Compute  $\hat{H}_0 = -G_0^{-1} \text{vec}(g'_{[X_0]}(0))$  and set  $H_0 = \text{uvec}(\hat{H}_0)$  where  $\text{uvec}$  forms an  $n \times p$  matrix in the reverse manner to the  $\text{vec}$  operator.

step k: If  $g'_{[X_k]}(0) = 0$ , then terminate. Otherwise, compute the step-size  $\lambda_{k+1}$  as follows.

$$\lambda_{k+1} = 2^{-l} \quad (7.62)$$

where  $l$  is the smallest positive integer such that

$$g_{[X_k]}(\lambda_{k+1}H_k) \leq g_{[X_k]} + c_1\lambda_{k+1}\text{tr}(H_k^T g'_{[X_k]}(0)), \quad (7.63)$$

$$\text{tr}(H_k^T g'_{[X_k]}(\lambda_{k+1}H_k)) \geq c_2\text{tr}(H_k^T g'_{[X_k]}(0)), \quad (7.64)$$

where  $0 < c_1 < c_2 < 1$ .

Set

$$B_{k+1} = \lambda_{k+1}H_k, \quad (7.65)$$

$$s_k = \text{vec}(B_{k+1}), \quad (7.66)$$

$$y_k = \text{vec}(g'_{[X_k]}(B_{k+1})) - \text{vec}(g'_{[X_k]}(0)), \quad (7.67)$$

$$G_{k+1} = G_k + \frac{y_k y_k^T}{\langle y_k, s_k \rangle} + \frac{G_k s_k (G_k s_k)^T}{\langle G_k s_k, s_k \rangle}, \quad (7.68)$$

$$X_{k+1} = \pi(B_{k+1}), \quad (7.69)$$

$$\hat{H}_{k+1} = -G_{k+1}^{-1} g'_{[X_{k+1}]}(0), \quad (7.70)$$

$$H_{k+1} = \text{uvec}(\hat{H}_{k+1}), \quad (7.71)$$

where  $\pi$  in (7.69) is defined in (7.56).

Increment  $k$  and repeat until convergence.

Figure 7.1 describes the result of applying our quasi-Newton method to solve (7.58). In particular, we take  $n = 10$ ,  $p = 6$ . This result demonstrates that the proposed quasi-Newton method has super-linear convergence rate.

## 7.5 Conclusions

In this chapter, we have presented a novel non-Riemannian quasi-Newton algorithm for the optimization on smooth manifolds. This algorithm is developed based on the local parameterization and proved to converge to the local minimum of the cost function. Furthermore, this algorithm is applied to minimize a cost function on the Grassmann manifold and the simulation

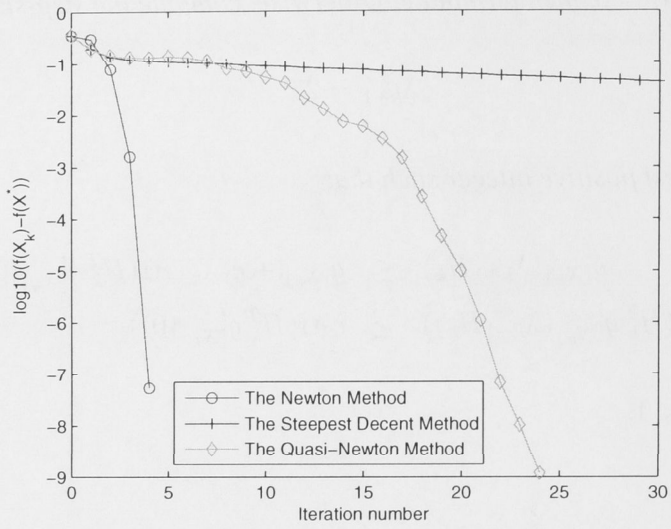


Figure 7.1: The result of Quasi-Newton Method compared against the Newton method and steepest decent method

results show the super-linear convergence of our method.

# Bibliography

- [1] S. Abe, J. Kawakami, and K. Hirasawa. Solving inequality constrained combinatorial optimization problems by the hopfield neural networks. *Neural Networks*, 5:663–670, 1992.
- [2] P. A. Absil, R. Mahony, and R. Sepulchre. Riemannian geometry of grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematicae*, 80(2):199220, Jan. 2004.
- [3] R. L. Adler, M. Dedieu J-P, Y. Joesph, M. Martens, and M. Shub. Newton’s method on riemannian manifolds and a geometric model for the human spine. *IMA J. Numerical Analysis*, 22:359–390, 2002.
- [4] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5:13–51, 1995.
- [5] F. Alizadeh, J. A. Haeberly, and M. Overton. Primal-dual interior point methods for semidefinite programming: convergence rates, stabilibty, and numerical analysis. *SIAM Journal on Optimization*, 8:746–768, 1998.
- [6] R. Benedetti and C. Petronio. *Lectures on Hyperbolic Geometry*. Springer-Verlag, Berlin, 1992.
- [7] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer-Verlag, New York, 1998.
- [8] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal. *Numerical Optimization: Theoretical and Practical Aspects*. Springer Verlag, New York, NY, 2003.
- [9] S. Boyd, L. E. Ghaoui, and V. Balakrishnan. *Linear Matrix Inequalities and Control Theory*. studies in applied mathematics 15. SIAM, Philadelphia, 1994.

- [10] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.
- [11] I. Brace. *Optimisation on Manifolds with Applications*. PhD thesis, University of Melbourne, Melbourne, March 2006.
- [12] I. Brace and J. H. Manton. An improved BFGS-on-manifold algorithm for computing weighted low rank approximations. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan, 2006.
- [13] R. W. Brockett. Differential geometry and the design of gradient algorithms. In ED. R. Green and S. T. Yau, editors, *Proceedings of the Symposium on Pure Mathematics*, pages 69–92, Providence, RI, 1993.
- [14] C. B. Brown and M. C. Bartholomew-Biggs. Some effective methods for unconstrained optimization based on the solution of systems of ordinary differential equations. *Journal on Optimization Theory and Applications*, 67:211–224, 1989.
- [15] C. G. Broyden. A new double-rank minimization algorithm. *Notices of the American Mathematical Society*, 16:670, 1969.
- [16] C. G. Broyden. The convergence of a class of double-rank minimization algorithms. *IMA Journal of Applied Mathematics*, 6(3):222–231, 1970.
- [17] W. Davidon. Technical report anl-5990. Technical report, Argonne National Laboratory, 1959.
- [18] P. Dedieu, J.-P. Priouret and G. Malajovich. Newton’s method on Riemannian manifolds: Covariant alpha-theory. *IMA Journal of Numerical Analysis*, 23:395–419, 2003.
- [19] A. Edelman, T. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*.
- [20] L. Faybusovich. Infinite dimensional semidefinite programming: regularized determinants and self-concordant barriers. *Fields Institute Communications*, 8:39–49, 1998.
- [21] R. Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13:317–322, 1970.

- [22] R. Fletcher. A nonlinear programming problem in statistics (educational testing). *SIAM Journal on Scientific and Statistical Computing*, 2:257–267, 1981.
- [23] R. Fletcher. *Practical Methods of Optimization, Vol I: Unconstrained Optimization*. Wiley, New York, 1987.
- [24] R. Fletcher and C. M. Reeves. Functions minimization by conjugate gradients. *Computer Journal*, 7:149–154, 1964.
- [25] D. Gabay. Minimizing a differentiable function over a differentiable manifold. *Journal of Optimization Theory and Applications*, 37(2):177–219, 1982.
- [26] P. E. Gill, W. Murray, A. Saunders, and M. H. Wright. On projected newton methods for linear programming and an equivalence to karmarkars projective method. *Math. Program.*, 36:183–209, 1986.
- [27] F. Glineur. Improving complexity of structured convex optimization problems using self-concordant barriers. *European Journal of Operational Research*, 143:291–310, 2002.
- [28] D. Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24(109):23–26, 1970.
- [29] G. H. Golub and C. F. V. Loan. *Matrix computations*. Johns Hopkins University Press, Baltimore, MD, 1996.
- [30] S. Helgason. *Differential Geometry, Lie Groups, and Symmetric Spaces*. Academic Press, New York, 1978.
- [31] C. Helmberg, F. Rendl, R. Vanderbei, and H. Wolkowicz. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, 6:342–361, 1996.
- [32] U. Helmke, K. Hüper, P. Lee, and J. B. Moore. Essential matrix estimation via newton-type methods. In *Proceedings of the MTNS*, Leuven, 2004.
- [33] U. Helmke, K. Hüper, and J. B. Moore. Quadratically convergent algorithms for optimal dextrous hand grasping. *IEEE Trans. Robot. Automat.*, 18(2):38–146, Apr. 2002.
- [34] U. Helmke and J. B. Moore. *Optimization and Dynamical Systems*. Springer-Verlag, London, 1996.

- [35] U. Helmke, S. Riardo, and J. B. Yoshizawa. Newton's algorithm in Euclidean Jordan algebras, with applications to robotics. *Communications in Information and Systems*, 2(3):283–297, 2002.
- [36] W. Yan, U. Helmke, and J. B. Moore. Global analysis of Oja's flow for neural networks. *IEEE Transactions on Neural Networks*, 5:674–683, 1993.
- [37] D. den Hertog. *Interior Point Approach to Linear, Quadratic and Convex Programming*. Mathematics and Its Applications 277. Kluwer Academic Publishers, London, 1994.
- [38] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.
- [39] K. Hüper and J. Trunpf. Newton-like methods for numerical optimization manifolds. In *Proceedings of the 38th Asilomar Conference on Signals, Systems and Computers*, pages 136–139, Asilomar Hotel and Conference Grounds, California, USA, 2004.
- [40] D. Jiang, J. B. Moore, and H. Ji. Self-concordant functions for optimization on smooth manifolds. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, pages 3631–3636, Bahamas, Dec. 2004.
- [41] D. Jiang and J. Wang. A recurrent neural network for real-time semidefinite programming. *IEEE Transactions on Neural Networks*, 10:81–93, 1999.
- [42] H. Karcher. Riemannian center of mass and mollifier smoothing. *Communication on Pure and Applied Mathematics*, 30:509–541, 1977.
- [43] N. Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.
- [44] M. P. Kennedy and L. O. Chua. Neural network for nonlinear programming. *IEEE Trans. Circuits Syst.*, 35:554–562, 1988.
- [45] L. G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979.
- [46] S. Lang. *Fundamentals of Differential Geometry*. Springer, New York, 1999.

- [47] P. Y. Lee. *Geometric Optimization for Computer Vision*. PhD thesis, Australian National University, Canberra, April 2005.
- [48] D. G. Luenberger. The gradient projection method along geodesics. *Management Science*, 18:620–631, 1972.
- [49] D. G. Luenberger. *Introduction to Linear and Nonlinear Programming*. Reading, 1973.
- [50] Y. Ma, J. Kosecka, and S. Sastry. Optimization criteria and geometric algorithms for motion and structure estimation. *International Journal of Computer Vision*, 44(3):219–249, 2001.
- [51] R. Mahony. *Optimization Algorithms on Homogeneous Spaces*. PhD thesis, Australian National University, Canberra, March 1994.
- [52] R. Mahony. The constrained newton method on a lie group and the symmetric eigenvalue problem. *Linear Algebra and Its Applications*, 248:67–89, 1996.
- [53] J. H. Manton. Optimisation algorithms exploiting unitary constraints. *IEEE Transactions on Signal Processing*, 50(3):635–650, Mar. 2002.
- [54] J. H. Manton. On the various generalisations of optimisation algorithms to manifolds. In *Sixteenth International Symposium on Mathematical Theory of Networks and Systems*, Katholieke Universiteit Leuven, Belgium, 2004.
- [55] J. H. Manton. A centroid (karcher mean) approach to the joint approximate diagonalisation problem: The real symmetric case. *Digital Signal Processing*, 16:468–478, 2006.
- [56] L. Nazareth. A relationship between BFGS and conjugate gradient algorithms and its implications for new algorithms. *SIAM Journal on Numerical Analysis*, 16(5):794–800, Oct. 1979.
- [57] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, Massachusetts, 2004.
- [58] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. studies in applied mathematics 13. SIAM, Philadelphia, 1994.
- [59] Y. Nesterov and M. Todd. On the Riemannian geometry defined by self-concordant barriers and interior-point methods. *Foundations of Computational Mathematics*, pages 333–361, 2002.



- [60] Y. Nishimori and S. Akaho. Learning algorithms utilizing quasi-geodesic flows on the stiefel manifold. *Neurocomputing*, 67:106–135, 2005.
- [61] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Verlag, New York, NY, 1999.
- [62] P. Pan. New ode methods for equality constrained optimization, part 1: equations. *Journal of Computational Mathematics*, 10:77–92, 1992.
- [63] J. Peng, C. Roos, and T. Terlaky. New complexity analysis of the primal-dual method for semidefinite optimization based on the netterov-todd direction. *Journal of Optimization Theory and Applications*, 109:327–343, 2001.
- [64] E. Polak. *Optimization: Algorithms and Cosistent Approximations*. Springer Verlag, New York, NY, 1997.
- [65] D. F. Shanno. Conditioning of quasi-newton methods for function minization. *Mathematics of Computation*, 24:647–656, 1970.
- [66] S. T. Smith. *Geometric Optimization Methods for Adaptive Filtering*. PhD thesis, Harvard University, Cambridge Massachusetts, 1993.
- [67] S. T. Smith. Optimization techniques on Riemannian manifolds. *Fields Institute Communications*, 3:113–146, 1994.
- [68] J. Sun and J. Zhang. Global convergence of conjugate gradient methods without line search. *Annals of Operations Research*, 103:161–173.
- [69] D. W. Tank and J. Hopfield. Simple neural optimization: an a/d converter, siganl decision network, and a linear programming circuit. *IEEE Trans. Circuits Syst.*, 33:533–541, 1986.
- [70] C. Udriste. *Convex Functions and Optimization Methods on Riemannian Manifolds*. Mathematics and Its Applications 297. Kluwer Academic Publishers, London, 1994.
- [71] C. Udriste. Optimization methods on Riemannian manifolds. *Algebras, Groups and Geometries*, 14:339–359, 1997.
- [72] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

- 
- [73] F. R. William. Hyperboloid geometry on a hyperboloid. *The American Mathematical Monthly*, 100:442–455, 1993.
- [74] M. Xiong, J. Wang, and P. Wang. Differential-algebraic approach to linear programming. *Journal of Optimization Theory and Applications*, 114:443–470, 2002.
- [75] S. Zhang. A new self-dual embedding for convex programming. Technical report seem2001-09, Department of Systems Engineering & Engineering Management, the Chinese University of Hong Kong, 2001.