Optimization and Its Applications

OPTIMIZATION THEORY AND METHODS

Nonlinear Programming

WENYU SUN
YA-XIANG YUAN

Springer

OPTIMIZATION THEORY AND METHODS
Nonlinear Programming

Springer Optimization and Its Applications

VOLUME 1

Aims and Scope

Optimization has been expanding in all directions at an astonishing rate during the last few decades. New algorithmic and theoretical techniques have been developed, the diffusion into other disciplines has proceeded at a rapid pace, and our knowledge of all aspects of the field has grown even more profound. At the same time, one of the most striking trends in optimization is the constantly increasing emphasis on the interdisciplinary nature of the field. Optimization has been a basic tool in all areas of applied mathematics, engineering, medicine, economics and other sciences.

The series *Springer Optimization and Its Applications* publishes undergraduate and graduate textbooks, monographs and state-of-the-art expository works that focus on algorithms for solving optimization problems and also study applications involving such problems. Some of the topics covered include nonlinear optimization (convex and nonconvex), network flow problems, stochastic optimization, optimal control, discrete optimization, multi-objective programming, description of software packages, approximation techniques and heuristic approaches.

OPTIMIZATION THEORY AND METHODS
Nonlinear Programming

By

WENYU SUN
Nanjing Normal University, Nanjing, China

YA-XIANG YUAN
Chinese Academy of Science, Beijing, China

Library of Congress Control Number: 2005042696

ISBN-10: 0-387-24975-3 e-ISBN: 0-387-24976-1

ISBN-13: 978-0-387-24975-9

Printed on acid-free paper.

Printed in the United States of America.

9 8 7 6 5 4 3 2 1

springer.com

Contents

Preface

Optimization is a subject that is widely and increasingly used in science, engineering, economics, management, industry, and other areas. It deals with selecting the best of many possible decisions in real-life environment, constructing computational methods to find optimal solutions, exploring the theoretical properties, and studying the computational performance of numerical algorithms implemented based on computational methods.

Along with the rapid development of high-performance computers and progress of computational methods, more and more large-scale optimization problems have been studied and solved. As pointed out by Professor Yuqi He of Harvard University, a member of the US National Academy of Engineering, optimization is a cornerstone for the development of civilization.

This book systematically introduces optimization theory and methods, discusses in detail optimality conditions, and develops computational methods for unconstrained, constrained, and nonsmooth optimization. Due to limited space, we do not cover all important topics in optimization. We omit some important topics, such as linear programming, conic convex programming, mathematical programming with equilibrium constraints, semi-infinite programming, and global optimization. Interested readers can refer to Dantzig [78], Walsch [347], Shu-Cheng Fang and S. Puthenpura [121], Luo, Pang, and Ralph [202], Wright [358], Wolkowitz, Saigal, and Vandenberghe [355].

The book contains a lot of recent research results on nonlinear programming including those of the authors, for example, results on trust region methods, inexact Newton method, self-scaling variable metric method, conic model method, non-quasi-Newton method, sequential quadratic programming, and nonsmooth optimization, etc.. We have tried to make the book

self-contained, systematic in theory and algorithms, and easy to read. For most methods, we motivate the idea, study the derivation, establish the global and local convergence, and indicate the efficiency and reliability of the numerical performance. The book also contains an extensive, not complete, bibliography which is an important part of the book, and the authors hope that it will be useful to readers for their further studies.

This book is a result of our teaching experience in various universities and institutes in China and Brazil in the past ten years. It can be used as a textbook for an optimization course for graduates and senior undergraduates in mathematics, computational and applied mathematics, computer science, operations research, science and engineering. It can also be used as a reference book for researchers and engineers.

We are indebted to the following colleagues for their encouragement, help, and suggestions during the preparation of the manuscript: Professors Kang Feng, Xuchu He, Yuda Hu, Liqun Qi, M.J.D. Powell, Raimundo J.B. Sampaio, Zhongci Shi, E. Spedicato, J. Stoer, T. Terlaky, and Chengxian Xu. Special thanks should be given to many of our former students who read early versions of the book and helped us in improving it. We are grateful to Edwin F. Beschler and several anonymous referees for many valuable comments and suggestions. We would like to express our gratitude to the National Natural Science Foundation of China for the continuous support to our research. Finally, we are very grateful to Editors John Martindale, Angela Quilici Burke, and Robert Saley of Springer for their careful and patient work.

Wenyu Sun, Nanjing Normal University
Yaxiang Yuan, Chinese Academy of Science
April 2005

Chapter 1

Introduction

1.1 Introduction

Optimization Theory and Methods is a young subject in applied mathematics, computational mathematics and operations research which has wide applications in science, engineering, business management, military and space technology. The subject is involved in optimal solution of problems which are defined mathematically, i.e., given a practical problem, the "best" solution to the problem can be found from lots of schemes by means of scientific methods and tools. It involves the study of optimality conditions of the problems, the construction of model problems, the determination of algorithmic method of solution, the establishment of convergence theory of the algorithms, and numerical experiments with typical problems and real life problems. Though optimization might date back to the very old extreme-value problems, it did not become an independent subject until the late 1940s, when G.B. Dantzig presented the well-known simplex algorithm for linear programming. After the 1950s, when conjugate gradient methods and quasi-Newton methods were presented, the nonlinear programming developed greatly. Now various modern optimization methods can solve difficult and large scale optimization problems, and become an indispensable tool for solving problems in diverse fields.

The general form of optimization problems is

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in X, \end{aligned} \tag{1.1.1}$$

where $x \in R^n$ is a decision variable, $f(x)$ an objective function, $X \subset R^n$

a constraint set or feasible region. Particularly, if the constraint set $X = R^n$, the optimization problem (1.1.1) is called an unconstrained optimization problem:

$$\min_{x \in R^n} f(x). \tag{1.1.2}$$

The constrained optimization problem can be written as follows:

$$\begin{array}{ll} \min_{x \in R^n} & f(x) \\ \text{s.t.} & c_i(x) = 0,\ i \in E, \\ & c_i(x) \geq 0,\ i \in I, \end{array} \tag{1.1.3}$$

where E and I are, respectively, the index set of equality constraints and inequality constraints, $c_i(x), (i = 1, \cdots, m \in E \cup I)$ are constraint functions. When both objective function and constraint functions are linear functions, the problem is called linear programming. Otherwise, the problem is called nonlinear programming.

This book mainly studies solving unconstrained optimization problem (1.1.2) and constrained optimization problem (1.1.3) from the view points of both theory and numerical methods. Chapters 2 to 7 deal with unconstrained optimization. Chapters 8 to 13 discuss constrained optimization. Finally, in Chapter 14, we give a simple and comprehensive introduction to nonsmooth optimization.

1.2 Mathematics Foundations

In this section, we shall review a number of results from linear algebra and analysis which are useful in optimization theory and methods.

Throughout this book, R^n will denote the real n-dimensional linear space of column vector x with components $x_1, \cdots, x_n$, and C^n the corresponding space of complex column vectors. For $x \in R^n$, x^T denotes the transpose of x, while, for $x \in C^n$, x^H is the conjugate transpose. A real $m \times n$ matrix $A = (a_{ij})$ defines a linear mapping from R^n to R^m and will be written as $A \in R^{m \times n}$ or $A \in L(R^n, R^m)$ to denote either the matrix or the linear operator. Similarly, a complex $m \times n$ matrix A will be written as $A \in C^{m \times n}$ or $A \in L(C^n, C^m)$.

1.2.1 Norm

Definition 1.2.1 *A mapping $\|\cdot\|$ is called a norm if and only if it satisfies the following properties:*

(i) $\|x\| \geq 0, \forall x \in R^n$; $\|x\| = 0$ *if and only if* $x = 0$;

(ii) $\|\alpha x\| = |\alpha|\|x\|, \forall \alpha \in R, x \in R^n$;

(iii) $\|x + y\| \leq \|x\| + \|y\|$, $\forall x, y \in R^n$.

Well-known examples of vector norm are as follows:

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|, \quad (l_\infty\text{-norm}) \tag{1.2.1}$$

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|, \quad (l_1\text{-norm}) \tag{1.2.2}$$

$$\|x\|_2 = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{1/2}, \quad (l_2\text{-norm}). \tag{1.2.3}$$

The above examples are particular cases of l_p-norm which is defined as

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}, \ (l_p\text{-norm}). \tag{1.2.4}$$

Another vector norm frequently used is the ellipsoid norm which is defined as

$$\|x\|_A = (x^T A x)^{1/2}, \tag{1.2.5}$$

where $A \in R^{n \times n}$ is a symmetric and positive definite matrix.

Similarly, we can define a matrix norm.

Definition 1.2.2 *Let $A, B \in R^{m \times n}$. A mapping $\|\cdot\| : R^{m \times n} \to R$ is said to be a matrix norm if it satisfies the properties*

(i) $\|A\| \geq 0, \forall A \in R^{m \times n}$; $\|A\| = 0$ *if and only if* $A = 0$;

(ii) $\|\alpha A\| = |\alpha|\|A\|, \forall \alpha \in R, A \in R^{m \times n}$;

(iii) $\|A + B\| \leq \|A\| + \|B\|$, $\forall A, B \in R^{m \times n}$.

Corresponding to the above vector l_p-norm, we have the matrix l_p-norm:

$$\|A\|_p = \sup_{x\neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \max_{\|x\|_p=1} \|Ax\|_p \tag{1.2.6}$$

which is said to be induced by, or subordinate to, the vector l_p-norm. In particular,

$$\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{m} |a_{ij}|, \quad \text{(maximum column norm)} \tag{1.2.7}$$

$$\|A\|_\infty = \max_{1\leq i\leq m} \sum_{j=1}^{n} |a_{ij}|, \quad \text{(maximum row norm)} \tag{1.2.8}$$

$$\|A\|_2 = \left(\lambda_{\max}(A^T A)\right)^{1/2}, \quad \text{(spectral norm.)} \tag{1.2.9}$$

Obviously, we have

$$\|A^{-1}\|_p = \frac{1}{\min_{x\neq 0} \frac{\|Ax\|_p}{\|x\|_p}}.$$

For an induced matrix norm, we always have $\|I\| = 1$, where I is an $n \times n$ identity matrix. More generally, for any vector norm $\|\cdot\|_\alpha$ on R^n and $\|\cdot\|_\beta$ on R^m, the matrix norm is defined by

$$\|A\|_{\alpha,\beta} = \sup_{x\neq 0} \frac{\|Ax\|_\beta}{\|x\|_\alpha}. \tag{1.2.10}$$

The most frequently used matrix norms also include the Frobenius norm

$$\|A\|_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2\right)^{1/2} = [\mathrm{tr}(A^T A)]^{1/2}, \tag{1.2.11}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a square matrix with $\mathrm{tr}(A) = \sum_{i=1}^{n} a_{ii}$. The trace satisfies

1. $\mathrm{tr}(\alpha A + \beta B) = \alpha \mathrm{tr}(A) + \beta \mathrm{tr}(B)$;
2. $\mathrm{tr}(A^T) = \mathrm{tr}(A)$;
3. $\mathrm{tr}(AB) = \mathrm{tr}(BA)$;
4. $\mathrm{tr}(A) = \sum_{i=1}^{n} \lambda_i$ if the eigenvalues of A are denoted by $\lambda_1, \cdots, \lambda_n$.

The weighted Frobenius norm and weighted l_2-norm are defined, respectively, as

$$\|A\|_{M,F} = \|MAM\|_F, \ \|A\|_{M,2} = \|MAM\|_2, \tag{1.2.12}$$

where M is an $n \times n$ symmetric and positive definite matrix.

Further, let $A \in R^{n\times n}$; if we define $\|x\|' = \|Px\|$ for all $x \in R^n$ and P an arbitrary nonsingular matrix, then

$$\|A\|' = \|PAP^{-1}\|. \tag{1.2.13}$$

The orthogonally invariant matrix norm is a class of important norms which satisfies, for $A \in R^{m\times n}$ and U an $m \times m$ orthogonal matrix, the identity

$$\|UA\| = \|A\|. \tag{1.2.14}$$

Clearly, the l_2-norm and the Frobenius norm are orthogonally invariant matrix norms.

A vector norm $\|\cdot\|$ and a matrix norm $\|\cdot\|'$ are said to be consistent if, for every $A \in R^{m\times n}$ and $x \in R^n$,

$$\|Ax\| \le \|A\|'\|x\|. \tag{1.2.15}$$

Obviously, the l_p-norm has this property, i.e.,

$$\|Ax\|_p \le \|A\|_p\|x\|_p. \tag{1.2.16}$$

More generally, for any vector norm $\|\cdot\|_\alpha$ on R^n and $\|\cdot\|_\beta$ on R^m we have

$$\|Ax\|_\beta \le \|A\|_{\alpha,\beta}\|x\|_\alpha, \tag{1.2.17}$$

where $\|A\|_{\alpha,\beta}$ is defined by

$$\|A\|_{\alpha,\beta} = \sup_{x\neq 0} \frac{\|Ax\|_\beta}{\|x\|_\alpha} \tag{1.2.18}$$

which is subordinate to the vector norm $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$.

Likewise, if a norm $\|\cdot\|$ satisfies

$$\|AB\| \le \|A\|\|B\|, \tag{1.2.19}$$

we say that the matrix norm satisfies the consistency condition (or submultiplicative property). It is easy to see that the Frobenius norm and the induced matrix norms satisfy the consistency condition, and we have

$$\|AB\|_F \le \min\{\|A\|_2\|B\|_F, \|A\|_F\|B\|_2\}. \tag{1.2.20}$$

Next, about the equivalence of norms, we have

Definition 1.2.3 *Let $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$ be two arbitrary norms on R^n. If there exist $\mu_1, \mu_2 > 0$, such that*

$$\mu_1\|x\|_\alpha \le \|x\|_\beta \le \mu_2\|x\|_\alpha,\ \forall x \in R^n, \tag{1.2.21}$$

we say that the norms $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$ are equivalent.

In particular, we have

$$\|x\|_2 \le \|x\|_1 \le \sqrt{n}\|x\|_2, \tag{1.2.22}$$
$$\|x\|_\infty \le \|x\|_2 \le \sqrt{n}\|x\|_\infty, \tag{1.2.23}$$
$$\|x\|_\infty \le \|x\|_1 \le n\|x\|_\infty, \tag{1.2.24}$$
$$\|x\|_\infty \le \|x\|_2 \le \|x\|_1, \tag{1.2.25}$$
$$\sqrt{\lambda}\|x\|_2 \le \|x\|_A \le \sqrt{\Lambda}\|x\|_2, \tag{1.2.26}$$

where λ and Λ are the smallest and the largest eigenvalues of A respectively. For $A \in R^{m\times n}$, we have

$$\|A\|_2 \le \|A\|_F \le \sqrt{n}\|A\|_2, \tag{1.2.27}$$
$$\max_{i,j}|a_{ij}| \le \|A\|_2 \le \sqrt{mn}\max_{i,j}|a_{ij}|, \tag{1.2.28}$$
$$\frac{1}{\sqrt{n}}\|A\|_\infty \le \|A\|_2 \le \sqrt{m}\|A\|_\infty, \tag{1.2.29}$$
$$\frac{1}{\sqrt{m}}\|A\|_1 \le \|A\|_2 \le \sqrt{n}\|A\|_1. \tag{1.2.30}$$

By use of norms, it is immediate to introduce the notation of distance. Let $x, y \in R^n$, the distance between two points x and y is defined by $\|x-y\|$. In particular, in the 2-norm, if $x = (x_1, \cdots, x_n)^T, y = (y_1, \cdots, y_n)^T$, then

$$\|x-y\|_2 = \sqrt{(x_1-y_1)^2 + \cdots + (x_n-y_n)^2}$$

which is just a direct generalization of distance in analytical geometry.

Obviously, by Definition 1.2.1, we have the following properties of distance:

1. $\|x-y\| \ge 0, \|x-y\| = 0$ if and only if $x = y$.

2. $\|x - z\| \leq \|x - y\| + \|y - z\|$.

3. $\|x - y\| = \|y - x\|$.

A vector sequence $\{x_k\}$ is said to be convergent to x^* if

$$\lim_{k\to\infty} \|x_k - x^*\| = 0. \tag{1.2.31}$$

A matrix sequence $\{A_k\}$ is said to be convergent to A if

$$\lim_{k\to\infty} \|A_k - A\| = 0. \tag{1.2.32}$$

Choice of norms is irrelevant since all norms in finite dimension space are equivalent.

Definition 1.2.4 *A sequence $\{x_k\} \subset R^n$ is said to be a Cauchy sequence if*

$$\lim_{m,l\to\infty} \|x_m - x_l\| = 0; \tag{1.2.33}$$

i.e., given $\epsilon > 0$, there is an integer N such that $\|x_m - x_l\| < \epsilon$ for all $m, l > N$.

In R^n, a sequence $\{x_k\}$ converges if and only if the sequence $\{x_k\}$ is a Cauchy sequence . However, in a normed space, a Cauchy sequence may not be convergent.

We conclude this subsection with several inequalities on norms.

(1) Cauchy-Schwarz inequality :

$$|x^T y| \leq \|x\|_2 \|y\|_2, \tag{1.2.34}$$

the equality holds if and only if x and y are linearly dependent.

(2) Let A be an $n \times n$ symmetric and positive definite matrix, then the inequality

$$|x^T A y| \leq \|x\|_A \|y\|_A \tag{1.2.35}$$

holds; the equality holds if and only if x and y are linearly dependent.

(3) Let A be an $n \times n$ symmetric and positive definite matrix, then the inequality

$$|x^T y| \leq \|x\|_A \|y\|_{A^{-1}} \tag{1.2.36}$$

holds; the equality holds if and only if x and $A^{-1}y$ are linearly dependent.

(4) Young inequality: Assume that real numbers p and q are each larger than 1, and $\frac{1}{p} + \frac{1}{q} = 1$. If x and y are also real numbers, then

$$xy \leq \frac{x^p}{p} + \frac{y^q}{q}, \tag{1.2.37}$$

and equality holds if and only if $x^p = y^q$.

Proof. Set $s = x^p$ and $t = y^q$. From the arithmetic-geometry inequality, we immediately have

$$xy = s^{1/p} t^{1/q} \leq \frac{s}{p} + \frac{t}{q} = \frac{x^p}{p} + \frac{y^q}{q}.$$

Further, the equality holds if and only if $s = t$, i.e., $x^p = y^q$. □

(5) Hölder inequality:

$$|x^T y| \leq \|x\|_p \|y\|_q = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \left(\sum_{i=1}^{n} |y_i|^q \right)^{1/q}, \tag{1.2.38}$$

where p and q are real numbers larger than 1 and satisfy $\frac{1}{p} + \frac{1}{q} = 1$.

Proof. If $x = 0$ or $y = 0$, the result is trivial. Now we assume that both x and y are not zero. From Young inequality, we have

$$\frac{|x_i y_i|}{\|x\|_p \|y\|_q} \leq \frac{1}{p} \frac{|x_i|^p}{\|x\|_p^p} + \frac{1}{q} \frac{|y_i|^q}{\|y\|_q^q}, \ i = 1, \cdots, n.$$

Taking the sum over i on both sides of the above inequality yields

$$\begin{aligned} & \frac{1}{\|x\|_p \|y\|_q} \sum_{i=1}^{n} |x_i y_i| \\ \leq \ & \frac{1}{p\|x\|_p^p} \sum_{i=1}^{n} |x_i|^p + \frac{1}{q\|y\|_q^q} \sum_{i=1}^{n} |y_i|^q \\ = \ & \frac{1}{p} + \frac{1}{q} \\ = \ & 1. \quad \square \end{aligned}$$

Multiplying $\|x\|_p \|y\|_q$ on both sides gives our result.

(6) Minkowski inequality:

$$\|x + y\|_p \leq \|x\|_p + \|y\|_p, \tag{1.2.39}$$

i.e.,

$$\left(\sum_{i=1}^{n}|x_i+y_i|^p\right)^{1/p} \le \left(\sum_{i=1}^{n}|x_i|^p\right)^{1/p} + \left(\sum_{i=1}^{n}|y_i|^p\right)^{1/p}, \tag{1.2.40}$$

where $p \ge 1$. The proof of this inequality will be given in §1.3.2 as an application of the convexity of a function.

1.2.2 Inverse and Generalized Inverse of a Matrix

In this subsection we collect some basic results of inverse and generalized inverse.

Theorem 1.2.5 *(Von-Neumann Lemma) Let $\|\cdot\|$ be a consistent matrix norm with $\|I\| = 1$. Let $E \in R^{n\times n}$. If $\|E\| < 1$, then $I - E$ is nonsingular, and*

$$(I-E)^{-1} = \sum_{k=0}^{\infty} E^k, \tag{1.2.41}$$

$$\|(I-E)^{-1}\| \le \frac{1}{1-\|E\|}. \tag{1.2.42}$$

If $A \in R^{n\times n}$ is nonsingular and $\|A^{-1}(B-A)\| < 1$, then B is nonsingular and satisfies

$$B^{-1} = \sum_{k=0}^{\infty}(I - A^{-1}B)^k A^{-1}, \tag{1.2.43}$$

and

$$\|B^{-1}\| \le \frac{\|A^{-1}\|}{1-\|A^{-1}(B-A)\|}. \tag{1.2.44}$$

Proof. Since $\|E\| < 1$, then

$$S_k \triangleq I + E + E^2 + \cdots + E^k$$

defines a Cauchy sequence , and hence S_k is convergent. So,

$$\sum_{k=0}^{\infty} E^k = \lim_{k\to\infty} S_k = (I-E)^{-1}$$

which proves (1.2.41)-(1.2.42).

Since A is nonsingular and $\|A^{-1}(B-A)\| = \|-(I-A^{-1}B)\| < 1$, by setting $E = I - A^{-1}B$ and using (1.2.41) and (1.2.42), we obtain immediately (1.2.43) and (1.2.44). □

This theorem indicates that the matrix B is invertible if B is sufficiently approximate to an invertible matrix A. The above theorem also can be written in the following form which sometimes is said to be the perturbation theorem:

Theorem 1.2.6 *Let $A, B \in R^{n\times n}$. Assume that A is invertible with $\|A^{-1}\| \leq \alpha$. If $\|A - B\| \leq \beta$ and $\alpha\beta < 1$, then B is also invertible, and*

$$\|B^{-1}\| \leq \frac{\alpha}{1-\alpha\beta}. \tag{1.2.45}$$

Let L and M be subspaces of R^n. The sum of two subspaces L and M is defined as

$$L + M = \{x = y + z \mid \ y \in L, z \in M\}. \tag{1.2.46}$$

The intersection of two subspaces L and M is defined as

$$L \cap M = \{x \mid x \in L \ \text{ and } \ x \in M\}. \tag{1.2.47}$$

Two subspaces L and M are orthogonal, denoted by $L \perp M$, if

$$< y, z >= 0, \quad \forall y \in L, \forall z \in M.$$

R^n is said to be a direct sum of L and M, denoted by

$$R^n = L \oplus M,$$

if and only if $R^n = L + M$ and $L \cap M = \{0\}$.

Let $R^n = L \oplus M$. If a linear operator $P : R^n \to R^n$ satisfies

$$Py = y, \forall y \in L; \ Pz = 0, \forall z \in M,$$

then P is called a projector of R^n onto the subspace L along the subspace M. Such a projector is denoted by $P_{L,M}$ or P. If $M \perp L$, then the above projector is called an orthogonal projector, denoted by P_L or P.

Normally, $C^{m\times n}$ denotes a set of all complex $m \times n$ matrices, $C_r^{m\times n}$ denotes a set of all complex $m \times n$ matrices with rank r. A^* denotes the conjugate transpose of a matrix A. For a real matrix, $R^{m\times n}$ and $R_r^{m\times n}$ have

similar meaning. Now we present some definitions and representations of the generalized inverse of a matrix A.

Let $A \in C^{m\times n}$. Then $A^+ \in C^{n\times m}$ is a Moore-Penrose generalized inverse of A if

$$AA^+A = A,\ A^+AA^+ = A^+,\ (AA^+)^* = AA^+,\ (A^+A)^* = A^+A, \quad (1.2.48)$$

or equivalently,

$$AA^+ = P_{R(A)},\ A^+A = P_{R(A^+)}, \quad (1.2.49)$$

where $P_{R(A)}$ and $P_{R(A^+)}$ are the orthogonal projectors on range $R(A)$ and $R(A^+)$ respectively.

If $A \in C_r^{m\times n}$ and A has the orthogonal decomposition

$$A = Q^*RP, \quad (1.2.50)$$

where Q and P are $m\times m$ and $n\times n$ unitary matrices respectively, $R \in C^{m\times n}$,

$$R = \begin{bmatrix} R_{11} & 0 \\ 0 & 0 \end{bmatrix},$$

where R_{11} is the $r \times r$ nonsingular upper triangular matrix, then

$$A^+ = P^*R^+Q, \quad (1.2.51)$$

where

$$R^+ = \begin{bmatrix} R_{11}^{-1} & 0 \\ 0 & 0 \end{bmatrix}.$$

Similarly, if $A \in C_r^{m\times n}$ has the singular value decomposition (SVD)

$$A = UDV^*, \quad (1.2.52)$$

where U and V are $m \times m$ and $n \times n$ unitary matrices respectively,

$$D = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \in C^{m\times n},$$

where $\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_r), \sigma_i > 0\ (i = 1, \cdots, r)$ are the nonzero singular values of A, then

$$A^+ = VD^+U^*, \quad (1.2.53)$$

where

$$D^+ = \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{bmatrix}.$$

An important role of the generalized inverse is that it offers the solution of general linear equations (including singular, rectangular, or inconsistent case). In the following we state this theorem and prove it by the singular value decomposition.

Theorem 1.2.7 *Let $A \in C^{m\times n}, b \in C^m$. Then $\bar{x} = A^+b$ is the unique solution of $Ax = b$, i.e.,*

$$\|\bar{x}\| \leq \|x\|, \forall x \in \{x \mid \|Ax - b\| \leq \|Az - b\|, \forall z \in C^n\}. \tag{1.2.54}$$

Such an $\bar{x}$ is called the minimal least-squares solution of $Ax = b$.

Proof. From the singular value decomposition (1.2.52), (1.2.54) is equivalent to

$$\min_{x\in R^n}\{\|V^*x\| \mid \|DV^*x - U^*b\| \leq \|DV^*z - U^*b\|, \forall z \in R^n\}$$

i.e., for $y = V^*x$,

$$\min_{y\in R^n}\{\|y\| \mid \|Dy - U^*b\| \leq \|D\hat{z} - U^*b\|, \ \forall \hat{z} \in R^n\}. \tag{1.2.55}$$

Since

$$\|Dy - U^*b\|^2 = \sum_{i=1}^{r}(\sigma_i y_i - (U^*b)_i)^2 + \sum_{i=r+1}^{m}((U^*b)_i)^2$$

which is minimized by any y with $y_i = (U^*b)_i/\sigma_i$, $(i = 1, \cdots, r)$ and $\|y\|$ is minimized by setting $y_i = 0\,(i = r+1, \cdots, m)$, then $y = D^+U^*b$ is the minimal least-squares solution of (1.2.55). Therefore $\bar{x} = VD^+U^*b = A^+b$ is the minimal least-squares solution of $Ax = b$. □

1.2.3 Properties of Eigenvalues

In this subsection we state, in brief, some properties of eigenvalues and eigenvectors that we will use in the text. We also summarize the definitions of positive definite, negative definite and indefinite symmetric matrices and their characterizations in terms of eigenvalues.

The eigenvalue problem of a matrix A is that

$$Ax = \lambda x,\ A \in R^{n\times n},\ x \neq 0, x \in R^n, \tag{1.2.56}$$

where λ is called an eigenvalue of A, x an eigenvector of A corresponding to λ, (λ, x) an eigen-pair of A.

The spectral radius of A is defined as

$$\rho(A) = \max_{1\leq i\leq n} |\lambda_i|.$$

Let $A \in R^{m\times n}$ have singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$, then

$$\|A\|_2 = \sigma_1,$$
$$\|A\|_F^2 = \sigma_1^2 + \cdots + \sigma_n^2.$$

In particular, if $A \in R^{n\times n}$ is symmetric with eigenvalues $\lambda_1, \cdots, \lambda_n$, then

$$\|A\|_2 = \max_{1\leq i\leq n} |\lambda_i|.$$

Then we immediately have that if A is nonsingular, the condition number of A is

$$\kappa(A) = \frac{\sigma_1}{\sigma_n};$$

in addition, if A is symmetric, then

$$\kappa(A) = \frac{\max_i |\lambda_i|}{\min_i |\lambda_i|}.$$

Let $A \in R^{n\times n}$ with eigenvalues $\lambda_1, \cdots, \lambda_n$. We have the following conclusions about the eigenvalues.

1. The eigenvectors corresponding to the distinct eigenvalues of A are independent.

2. A is diagonalizable if and only if, for each eigenvalue of A, its geometric multiplicity is equal to the algebraic multiplicity, i.e., the dimension of its corresponding eigenvectors is equal to the multiplicity of the eigenvalue.

3. Let $f(A)$ be a polynomial of A. If (λ, x) is an eigen-pair of A, then $(f(\lambda), x)$ is the eigen-pair of $f(A)$.

4. Let $B = PAP^{-1}$, where $P \in R^{n\times n}$ is a nonsingular transformation matrix. If (λ, x) is an eigen-pair of A, then (λ, Px) is the eigen-pair of B. This means that the similar transformation does not change the eigenvalues of a matrix.

Definition 1.2.8 *Let $A \in R^{n\times n}$ be symmetric. A is said to be positive definite if $v^T Av > 0, \forall v \in R^n, v \neq 0$. A is said to be positive semidefinite if $v^T Av \geq 0, \forall v \in R^n$. A is said to be negative definite or negative semidefinite if $-A$ is positive definite or positive semidefinite. A is said to be indefinite if it is neither positive semidefinite nor negative semidefinite.*

The main properties of a symmetric matrix are as follows. Let $A \in R^{n\times n}$ be symmetric. Then

(1) All eigenvalues of A are real.

(2) The eigenvectors corresponding to the distinct eigenvalues of A are orthogonal.

(3) A is orthogonally similar to a diagonal matrix, i.e., there exists an $n \times n$ orthogonal matrix Q such that

$$Q^{-1}AQ = Q^T AQ = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix},$$

where $\lambda_1, \cdots, \lambda_n$ are the eigenvalues of A. This means a symmetric matrix has an orthonormal eigenvector system.

The following properties are about symmetric positive definite, symmetric positive semidefinite, and so on.

Let $A \in R^{n\times n}$ be symmetric. Then A is positive definite if and only if all its eigenvalues are positive. A is positive semidefinite if and only if all its eigenvalues are nonnegative. A is negative definite or negative semidefinite if and only if all its eigenvalues are negative or nonpositive. A is indefinite if and only if it has both positive and negative eigenvalues. Furthermore, A is positive definite if and only if A has a unique Cholesky factorization $A = LDL^T$ with all positive diagonal elements of D.

The following is the definition of the Rayleigh quotient of a matrix and its properties.

Definition 1.2.9 *Let A be an $n \times n$ Hermitian matrix and $u \in C^n$. Then the Rayleigh quotient of A is defined by*

$$R_\lambda(u) = \frac{u^*Au}{u^*u},\ u \neq 0. \tag{1.2.57}$$

Theorem 1.2.10 *Let A be an $n \times n$ Hermitian matrix and $u \in C^n$. Then the Rayleigh quotient defined by (1.2.57) has the following basic properties:*

(i) Homogeneous Property:

$$R_\lambda(\alpha u) = R_\lambda(u),\ \alpha \neq 0. \tag{1.2.58}$$

(ii) Extreme Property:

$$\lambda_1 = \max_{\|u\|_2=1} u^*Au = \max_{u\neq 0} \frac{u^*Au}{u^*u}, \tag{1.2.59}$$

$$\lambda_n = \min_{\|u\|_2=1} u^*Au = \min_{u\neq 0} \frac{u^*Au}{u^*u}, \tag{1.2.60}$$

which show that the Rayleigh quotient has bounded property:

$$\lambda_n \leq R_\lambda(u) \leq \lambda_1. \tag{1.2.61}$$

(iii) Minimal Residual Property: for any $u \in C^n$,

$$\|(A - R_\lambda(u)I)u\| \leq \|(A - \mu I)u\|,\ \forall \text{ real number } \mu. \tag{1.2.62}$$

Proof. Property (i) is immediate from Definition 1.2.9. Now we consider Property (ii). By Property (i), we can consider the Rayleigh quotient on a unit sphere, i.e.,

$$R_\lambda(u) = u^*Au,\ \|u\|_2 = 1.$$

Let T be a unitary matrix such that $T^*AT = \Lambda$, where Λ is a diagonal matrix. Also let $u = Ty$, then

$$u^*Au = y^*\Lambda y = \sum_{i=1}^{n} \lambda_i |y_i|^2 \begin{cases} \geq \lambda_n \sum_{i=1}^n |y_i|^2, \\ \leq \lambda_1 \sum_{i=1}^n |y_i|^2. \end{cases}$$

Note that $\|u\|_2 = \|y\|_2 = 1$, hence the boundedness follows. Furthermore, when $y_1 = 1$ and $y_i = 0, i \neq 1, \lambda_1$ is the maximum; when $y_n = 1$ and $y_i = 0, i \neq n, \lambda_n$ is the minimum. This proves Property (ii).

To establish Property (iii), we define

$$s(u) = Au - R_\lambda(u)u, \ u \neq 0, \tag{1.2.63}$$

which implies that

$$Au = R_\lambda(u)u + s(u). \tag{1.2.64}$$

By Definition 1.2.9, we have $\langle s(u), u\rangle = \langle Au - R_\lambda(u)u, u\rangle = 0$ which means that the decomposition (1.2.64) is an orthogonal decomposition. Thus $R_\lambda(u)u$ is an orthogonal projection of Au on $L = \{u\}$, which shows that the residual defined by (1.2.63) has the minimal residual Property (iii). □

Next, we state some concepts of reducible and irreducible matrices which are useful in discussing invertibility and positive definiteness of a matrix.

Definition 1.2.11 *Let $A \in R^{n\times n}$. A is said to be reducible if there is a permutation matrix P such that*

$$PAP^T = \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix},$$

where B_{11} and B_{22} are square matrices; A is irreducible if it is not reducible.

Equivalently, A is reducible if and only if there is a nonempty subset of indices $J \subset \{1, \cdots, n\}$ such that

$$a_{kj} = 0, \ \forall k \in J, j \notin J.$$

Definition 1.2.12 *Let $A \in R^{n\times n}$. A is said to be diagonally dominant if*

$$\sum_{j=1, j\neq i}^{n} |a_{ij}| \leq |a_{ii}|, \ i = 1, \cdots, n. \tag{1.2.65}$$

A is said to be strictly diagonally dominant if strict inequality holds in (1.2.65) for all i. A is said to be irreducibly diagonally dominant if it is irreducible, diagonally dominant, and strict inequality holds in (1.2.65) for at least one i.

The above concepts give an important theorem which is called the Diagonal Dominant Theorem.

Theorem 1.2.13 *(Diagonal Dominant Theorem) Let $A \in R^{n\times n}$ be either strictly or irreducibly diagonal dominant. Then A is invertible.*

As a corollary of the above theorem, we state the Gerschgorin circle Theorem which gives an isolation property of eigenvalues.

Theorem 1.2.14 *Let $A \in C^{n\times n}$. Define the i-th circle as*

$$\mathcal{D}_i = \{\lambda \mid |\lambda - a_{ii}| \leq \sum_{j=1, j\neq i}^{n} |a_{ij}|\},\ i = 1, \cdots, n.$$

Then each eigenvalue of A lies in the union $S = \cup_{i=1}^{n} \mathcal{D}_i$. This also means that

$$\min_i \lambda_i \geq \min_i \{a_{ii} - \sum_{j=1, j\neq i}^{n} |a_{ij}|\}$$

and

$$\max_i \lambda_i \leq \max_i \{a_{ii} + \sum_{j=1, j\neq i}^{n} |a_{ij}|\}.$$

1.2.4 Rank-One Update

The rank-one update of matrices is often used in optimization. In this subsection we introduce inverse of rank-one update, determinant of rank-one update, chain of the eigenvalues of rank-one update, and updating matrix factorizations. Detailed proofs can be found in books on linear algebra or numerical linear algebra.

The following theorem due to Sherman and Morrison is wellknown.

Theorem 1.2.15 *Let $A \in R^{n\times n}$ be nonsingular and $u, v \in R^n$ be arbitrary. If*

$$1 + v^T A^{-1} u \neq 0, \tag{1.2.66}$$

then the rank-one update $A + uv^T$ of A is nonsingular, and its inverse is represented by

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1} u}. \tag{1.2.67}$$

An interesting generalization of the above theorem is

Theorem 1.2.16 *(Sherman-Morrison-Woodburg Theorem)*

Let A be an $n\times n$ nonsingular matrix, U, V $n\times m$ matrices. If $I + V^* A^{-1} U$ is invertible, then $A + UV^*$ is invertible, and

$$(A + UV^*)^{-1} = A^{-1} - A^{-1}U(I + V^* A^{-1} U)^{-1} V^* A^{-1}. \tag{1.2.68}$$

Consider the determinant of a rank-one update; we have

$$\det(I + uv^T) = 1 + u^T v. \tag{1.2.69}$$

In fact, assuming $u \neq 0$, we have that the eigenvectors of $I + uv^T$ are either orthogonal to v or parallel to u. If they are orthogonal to v, the corresponding eigenvalues are 1; otherwise the corresponding eigenvalue is $1 + u^T v$. Hence (1.2.69) follows.

Furthermore, for the determinant of rank-two update, we have the following result:

$$\begin{aligned} & \det(I + u_1 u_2^T + u_3 u_4^T) \\ = \; & (1 + u_1^T u_2)(1 + u_3^T u_4) - (u_1^T u_4)(u_2^T u_3). \end{aligned} \tag{1.2.70}$$

In fact, as long as we note that

$$I + u_1 u_2^T + u_3 u_4^T = (I + u_1 u_2^T)[I + (I + u_1 u_2^T)^{-1} u_3 u_4^T],$$

it follows from (1.2.69) and (1.2.67) that

$$\begin{aligned} & \det(I + u_1 u_2^T + u_3 u_4^T) \\ = \; & (1 + u_1^T u_2)[1 + u_4^T (I + u_1 u_2^T)^{-1} u_3] \\ = \; & (1 + u_1^T u_2) \left[1 + u_4^T \left(I - \frac{u_1 u_2^T}{1 + u_1^T u_2} \right) u_3 \right] \\ = \; & (1 + u_1^T u_2)(1 + u_3^T u_4) - (u_1^T u_4)(u_2^T u_3). \end{aligned}$$

By $\|A\|_F^2 = \mathrm{tr}(A^T A)$, where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix, it follows that the Frobenius norm of rank-one update $A + xy^T$ is

$$\|A + xy^T\|_F^2 = \|A\|_F^2 + 2y^T A^T x + \|x\|^2 \|y\|^2. \tag{1.2.71}$$

About the chain of the eigenvalues of rank-one update, we have the following theorem.

Theorem 1.2.17 *Let A be an $n \times n$ symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. Also let $\bar{A} = A + \sigma uu^T$ with eigenvalues $\bar{\lambda}_1 \geq \bar{\lambda}_2 \geq \cdots \geq \bar{\lambda}_n$, where $u \in R^n$. Then we have the conclusions:*

(i) if $\sigma > 0$, then

$$\bar{\lambda}_1 \geq \lambda_1 \geq \bar{\lambda}_2 \geq \lambda_2 \geq \cdots \geq \bar{\lambda}_n \geq \lambda_n.$$

(ii) if $\sigma < 0$, then

$$\lambda_1 \geq \bar{\lambda}_1 \geq \lambda_2 \geq \bar{\lambda}_2 \geq \cdots \geq \lambda_n \geq \bar{\lambda}_n.$$

Next, we discuss updating matrix factorizations which conclude updates of Cholesky factorization and orthogonal decomposition.

Let B and $\bar{B}$ be $n \times n$ symmetric and positive definite matrices,

$$\bar{B} = B + \alpha yy^T, \; B = LDL^T. \tag{1.2.72}$$

We can find the Cholesky factorization $\bar{B} = \bar{L}\bar{D}\bar{L}^T$ as follows:

$$\begin{aligned} \bar{B} &= B + \alpha yy^T \\ &= L(D + \alpha pp^T)L^T, \end{aligned} \tag{1.2.73}$$

where p solves $Lp = y$. Note that since $D + \alpha pp^T$ is a positive definite matrix with the Cholesky factorization $D + \alpha pp^T = \hat{L}\hat{D}\hat{L}^T$, we have

$$\bar{B} = L\hat{L}\hat{D}\hat{L}^T L^T = \bar{L}\bar{D}\bar{L}^T, \tag{1.2.74}$$

where $\bar{L} = L\hat{L}, \bar{D} = \hat{D}$. The following algorithm gives the steps for computing $\bar{L}$ and $\bar{D}$.

Algorithm 1.2.18 *(Cholesky Factorization of Rank-One Update)*

1. *Set* $\alpha_1 = \alpha, w^{(1)} = y$.
2. *For* $j = 1, 2, \cdots, n$, *compute*

$$\begin{aligned} &p_j = w_j^{(j)}, \\ &\bar{d}_j = d_j + \alpha_j p_j^2, \\ &\beta_j = p_j \alpha_j / \bar{d}_j, \\ &\alpha_{j+1} = d_j \alpha_j / \bar{d}_j, \\ &w_r^{(j+1)} = w_r^{(j)} - p_j l_{rj}, \; r = j+1, \cdots, n, \\ &\bar{l}_{rj} = l_{rj} + \beta_j w_r^{(j+1)}, \; r = j+1, \cdots, n. \; \square \end{aligned}$$

Similarly, for the negative rank-one update of Cholesky factorization , we have

$$\begin{aligned} \bar{B} &= B - yy^T = L(D - pp^T)L^T \\ &= L\hat{L}\hat{D}\hat{L}^T L^T = \bar{L}\bar{D}\bar{L}^T. \end{aligned} \tag{1.2.75}$$

Since, in this case, it is possible that the elements of $\bar{D}$ become zero or negative due to round-off error, this phenomenon must be taken into consideration. The following algorithm keeps all $\bar{d}_j$ $(j = 1, \cdots, n)$ positive.

Algorithm 1.2.19 *(Cholesky Factorization of Negative Rank-One Update)*

1. *Solve* $Lp = y$ *for* p. *Set* $t_{n+1} = 1 - p^T D^{-1} p$. *If* $t_{n+1} < \epsilon_M$, *set* $t_{n+1} = \epsilon_M$, *where* ϵ_M *is the relative precision of the computer.*

2. *For* $j = n, n-1, \cdots, 1$, *compute*

$$\begin{aligned}
&t_j = t_{j+1} + p_j^2/d_j, \\
&\bar{d}_j = d_j t_{j+1}/t_j, \\
&\beta_j = -p_j/(d_j t_{j+1}), \\
&w_j^{(j)} = p_j, \\
&\bar{l}_{rj} = l_{rj} + \beta_j w_r^{(j+1)},\ r = j+1, \cdots, n., \\
&w_r^{(j)} = w_r^{(j+1)} + p_j l_{rj},\ r = j+1, \cdots, n. \ \square
\end{aligned}$$

Furthermore, Algorithm 1.2.18 and Algorithm 1.2.19 about Cholesky factorization of rank-one update can be used to compute the Cholesky factorization of rank-two update. Consider

$$\bar{B} = B + vw^T + wv^T. \tag{1.2.76}$$

Setting

$$x = (v+w)/\sqrt{2},\ y = (v-w)/\sqrt{2} \tag{1.2.77}$$

yields

$$\bar{B} = B + xx^T - yy^T, \tag{1.2.78}$$

so, we can use Algorithm 1.2.18 and Algorithm 1.2.19 to get the Cholesky factorization of $\bar{B}$.

Below, we consider the special cases of rank-two update. Let B be an $n \times n$ symmetric positive definite matrix with Cholesky factorization $B = LDL^T$. Consider the case adding one row and one column to B:

$$\bar{B} = \begin{bmatrix} B & b \\ b^T & \theta \end{bmatrix}, \tag{1.2.79}$$

where $b \in R^n$ and θ is a number. If we set

$$\hat{B} = \begin{bmatrix} B & 0 \\ 0 & \theta \end{bmatrix}, \tag{1.2.80}$$

then we have

$$\bar{B} = \hat{B} + e_{n+1} \begin{pmatrix} b \\ 0 \end{pmatrix}^T + \begin{pmatrix} b \\ 0 \end{pmatrix} e_{n+1}^T. \tag{1.2.81}$$

So, we can use the above algorithm to compute Cholesky factors $\bar{L}$ and $\bar{D}$ of $\bar{B}$. In addition, we can show that $\bar{L}$ and $\bar{D}$ have the following forms:

$$\bar{L} = \begin{bmatrix} L & 0 \\ l^T & 1 \end{bmatrix}, \ \bar{D} = \begin{bmatrix} D & 0 \\ 0 & d \end{bmatrix}. \tag{1.2.82}$$

In fact, it is enough to consider

$$\begin{bmatrix} L & 0 \\ l^T & 1 \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & d \end{bmatrix} \begin{bmatrix} L^T & l \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} B & b \\ b^T & \theta \end{bmatrix} \tag{1.2.83}$$

and solve the equations obtained

$$\begin{aligned} &LDl = b, \\ &d = \theta - l^T Dl \end{aligned} \tag{1.2.84}$$

for l and d. Then we get $\bar{L}$ and $\bar{D}$ from (1.2.82).

Now we consider the case deleting the j-th row and j-th column from B. Let $B = LDL^T$ with the form

$$B = \begin{bmatrix} B_1 & \vdots & B_2 \\ \cdots & \cdot & \cdots \\ B_2^T & \vdots & B_3 \end{bmatrix} \leftarrow j\text{-th row.} \tag{1.2.85}$$

Define

$$\bar{B} = \begin{bmatrix} B_1 & B_2 \\ B_2^T & B_3 \end{bmatrix} \} \, n-1 \text{columns.} \tag{1.2.86}$$

The algebraic operations give

$$\bar{B} = \hat{L} D \hat{L}^T, \tag{1.2.87}$$

which is our desired result, where $\hat{L}$ is an $(n-1) \times n$ matrix obtained by deleting the j-th row from L.

In the above, we discussed the Cholesky factorization of rank-one update. Next, we handle the QR factorization of rank-one update. Let $A, \bar{A} \in R^{n \times n}, u, v \in R^n$,

$$A = QR, \ \bar{A} = A + uv^T. \tag{1.2.88}$$

Then we have

$$\bar{A} = QR + uv^T = Q(R + wv^T), \tag{1.2.89}$$

where $w = Q^T u$. Forming QR decomposition

$$R + wv^T = \tilde{Q}\tilde{R},$$

we have

$$\bar{A} = Q\tilde{Q}\tilde{R} \triangleq \bar{Q}\bar{R}, \tag{1.2.90}$$

where $\bar{Q} = Q\tilde{Q}, \bar{R} = \tilde{R}$.

Similarly, if $m \times n$ matrix A $(m < n)$ has an orthogonal decomposition

$$A = [L\ 0]Q, \tag{1.2.91}$$

where L is an $m \times m$ unit lower triangular matrix and Q is an $n \times n$ orthogonal matrix with $Q^T Q = I$, then we can obtain the LQ decomposition of

$$\bar{A} = A + xy^T \tag{1.2.92}$$

as follows.

$$\begin{aligned}
\bar{A} &= A + xy^T \\
&= [L\ 0]Q + xy^T \\
&= ([L\ 0] + xw^T)Q \quad (\text{where } w = Qy) \\
&= ([L\ 0] + xw^T)P^T PQ \quad (\text{where } P^T P = I) \\
&= ([H\ 0] + \alpha x e_1^T)PQ \quad (\text{where } Pw = \alpha e_1, H = LP^T) \\
&= [\bar{H}\ 0]PQ \\
&= [\bar{H}\ 0]\bar{P}\bar{P}^T PQ \quad (\text{where } \bar{P}\bar{P}^T = I) \\
&= [\bar{L}\ 0]\bar{P}^T PQ \quad (\text{where } [\bar{H}\ 0]\bar{P} = [\bar{L}\ 0]) \\
&= [\bar{L}\ 0]\bar{Q} \quad (\text{where } \bar{Q} = \bar{P}^T PQ).
\end{aligned} \tag{1.2.93}$$

1.2.5 Function and Differential

This subsection presents some materials of set theory and multivariable calculus background.

Give a point $x \in R^n$ and a $\delta > 0$. The δ-neighborhood of x is defined as

$$N_\delta(x) = \{y \in R^n \mid \|y - x\| < \delta\}.$$

Let $D \subset R^n$ and $x \in D$. The point x is said to be an interior point of D if there exists a δ-neighborhood of x such that $N_\delta(x) \subset D$. The set of all such points is called the interior of D and is denoted by $\text{int}(D)$. Obviously, $\text{int}(D) \subset D$. Furthermore, if $\text{int}(D) = D$, i.e., every point of D is the interior point of D, then D is an open set.

$x \in D \subset R^n$ is said to be an accumulation point if for each $\delta > 0, D \cap N_\delta(x) \neq \phi$, where ϕ is an empty set. It means that there exists a subsequence $\{x_{n_k}\} \subset D$, such that $x_{n_k} \to x$. The set of all such points is called the closure of D and is denoted by $\bar{D}$. Obviously, $D \subset \bar{D}$. Furthermore, if $D = \bar{D}$, i.e., every accumulation point of D is contained in D, then D is said to be closed. It is also clear that a set $D \subset R^n$ is closed if and only if its complement is open.

A set $D \subset R^n$ is said to be compact if it is bounded and closed. For every sequence $\{x_k\}$ in a compact set D, there exists a convergent subsequence with a limit in D.

A function $f : R^n \to R$ is said to be continuous at $\bar{x} \in R^n$ if, for any given $\epsilon > 0$, there exists $\delta > 0$ such that $\|x - \bar{x}\| < \delta$ implies $|f(x) - f(\bar{x})| < \epsilon$. It can also be written as follows: $\forall \epsilon > 0, \exists \delta > 0$, such that $\forall x \in N_\delta(\bar{x})$, we have $f(x) \in N_\epsilon(f(\bar{x}))$. If f is continuous at every point in an open set $D \subset R^n$, then f is said to be continuous on D.

A continuous function $f : R^n \to R$ is said to be continuously differentiable at $x \in R^n$, if $\left(\frac{\partial f}{\partial x_i}\right)(x)$ exists and is continuous, $i = 1, \cdots, n$. The gradient of f at x is defined as

$$\nabla f(x) = \left[\frac{\partial f}{\partial x_1}(x), \cdots, \frac{\partial f}{\partial x_n}(x)\right]^T. \tag{1.2.94}$$

If f is continuously differentiable at every point of an open set $D \subset R^n$, then f is said to be continuously differentiable on D and denoted by $f \in C^1(D)$.

A continuously differentiable function $f : R^n \to R$ is called twice continuously differentiable at $x \in R^n$ if $\frac{\partial^2 f}{\partial x_i \partial x_j}(x)$ exists and is continuous, $i = 1, \cdots, n$. The Hessian of f is defined as the $n \times n$ symmetric matrix with elements

$$[\nabla^2 f(x)]_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}(x),\ 1 \leq i, j \leq n.$$

If f is twice continuously differentiable at every point in an open set $D \subset R^n$, then f is said to be twice continuously differentiable on D and denoted by $f \in C^{(2)}(D)$.

Let $f : R^n \to R$ be continuously differentiable on an open set $D \subset R^n$. Then for $x \in D$ and $d \in R^n$, the directional derivative of f at x in the direction d is defined as

$$f'(x;d) \stackrel{def}{=} \lim_{\theta \to 0} \frac{f(x+\theta d) - f(x)}{\theta} = \nabla f(x)^T d, \tag{1.2.95}$$

where $\nabla f(x)$ is the gradient of f at x, an $n \times 1$ vector.

For any $x, x+d \in D$, if $f \in C^1(D)$, then

$$\begin{aligned} f(x+d) &= f(x) + \int_0^1 \nabla f(x+td)^T d\mathrm{d}t \\ &= f(x) + \int_x^{x+d} \nabla f(\xi)\mathrm{d}\xi. \end{aligned} \tag{1.2.96}$$

Thus,

$$f(x+d) = f(x) + \nabla f(\xi)^T d,\ \xi \in (x, x+d). \tag{1.2.97}$$

Similarly, for all $x, y \in D$, we have

$$f(y) = f(x) + \nabla f(x + t(y-x))^T (y-x),\ t \in (0,1), \tag{1.2.98}$$

or

$$f(y) = f(x) + \nabla f(x)^T (y-x) + o(\|y-x\|). \tag{1.2.99}$$

It follows from (1.2.98) that

$$|f(y) - f(x)| \le \|y-x\| \sup_{\xi \in L(x,y)} \|f'(\xi)\|, \tag{1.2.100}$$

where $L(x,y)$ denotes the line segment with endpoints x and y.

Let $f \in C^{(2)}(D)$. For any $x \in D, d \in R^n$, the second directional derivative of f at x in the direction d is defined as

$$f''(x;d) = \lim_{\theta \to 0} \frac{f'(x+\theta d;d) - f'(x;d)}{\theta}, \tag{1.2.101}$$

which equals $d^T \nabla^2 f(x) d$, where $\nabla^2 f(x)$ denotes the Hessian of f at x. For any $x, x+d \in D$, there exists $\xi \in (x, x+d)$ such that

$$f(x+d) = f(x) + \nabla f(x)^T d + \frac{1}{2} d^T \nabla^2 f(\xi) d, \tag{1.2.102}$$

or

$$f(x+d) = f(x) + \nabla f(x)^T d + \frac{1}{2} d^T \nabla^2 f(x) d + o(\|d\|^2). \tag{1.2.103}$$

Let $h : R^n \to R$, $g : R^m \to R$, $f : R^n \to R^m$. Let $f \in C^1$, $g \in C^1$, $h(x_0) = g(f(x_0))$. Then the chain rule is

$$h'(x_0) = g'(f(x_0))f'(x_0), \tag{1.2.104}$$

where $f'(x_0) = \left[\frac{\partial f_i(x_0)}{\partial x^j}\right]_{m\times n}$ is an $m \times n$ matrix. Also

$$h''(x_0) = \nabla f(x_0)^T \nabla^2 g[f(x_0)]\nabla f(x_0) + \sum_{i=1}^{m} \frac{\partial g[f(x_0)]}{\partial f_i}[f_i(x_0)]''. \tag{1.2.105}$$

Next, we discuss the calculus of vector-valued functions.

A continuous function $F : R^n \to R^m$ is continuously differentiable at $x \in R^n$ if each component function $f_i (i = 1, \cdots, m)$ is continuously differentiable at x. The derivative $F'(x) \in R^{m\times n}$ of F at x is called the Jacobian matrix of F at x,

$$F'(x) = J(x)$$

with components

$$[F'(x)]_{ij} = [J(x)]_{ij} = \frac{\partial f_i}{\partial x^j}(x),\ i = 1, \cdots, m; j = 1, \cdots, n.$$

If $F : R^n \to R^m$ is continuously differentiable in an open convex set $D \subset R^n$, then for any $x, x + d \in D$, we have

$$F(x + d) - F(x) = \int_0^1 J(x + td)d\mathrm{d}t = \int_x^{x+d} F'(\xi)\mathrm{d}\xi. \tag{1.2.106}$$

In many of our considerations, we shall wish to single out different types of continuities.

Definition 1.2.20 *$F : D \subset R^n \to R^m$ is Hölder continuous on D if there exist constants $\gamma \geq 0$ and $p \in (0, 1]$ so that for all $x, y \in D$,*

$$\|F(y) - F(x)\| \leq \gamma\|y - x\|^p. \tag{1.2.107}$$

If $p = 1$, then F is called Lipschitz continuous on D and γ is a Lipschitz constant.

$F : D \subset R^n \to R^m$ is Hölder continuous at $x \in D$ if (1.2.107) holds for any y in the neighborhood of x.

Definition 1.2.21 *$F : D \subset R^n \to R^m$ is hemi-continuous at $x \in D$ if, for any $d \in R^n$ and $\epsilon > 0$, there is a $\delta = \delta(\epsilon, d)$ so that whenever $|t| < \delta$ and $x + td \in D$,*

$$\|F(x + td) - F(x)\| < \epsilon. \tag{1.2.108}$$

We also can define the upper hemi-continuous and lower hemi-continuous at $x \in D$ if, instead of (1.2.108), we use, respectively, $F(x + td) < F(x) + \epsilon$ and $F(x + td) > F(x) - \epsilon$ for sufficiently small t.

The following two theorems establish the bounds of errors within which some standard models approximate the objective functions. For $F : R^n \to R^m$, Theorem 1.2.22 gives a bound of the error in linear model $F(x) + F'(x)d$ as an approximation to $F(x + d)$. Similarly, for $f : R^n \to R$, Theorem 1.2.23 gives a bound of errors with a quadratic model as an approximation to $f(x + d)$.

Theorem 1.2.22 *Let $F : R^n \to R^m$ be continuously differentiable in the open convex set $D \subset R^n$. Let F' be Lipschitz continuous at $x \in D$. Then for any $x + d \in D$, we have*

$$\|F(x + d) - F(x) - F'(x)d\| \le \frac{\gamma}{2}\|d\|^2. \tag{1.2.109}$$

Proof.

$$\begin{aligned} F(x + d) - F(x) - F'(x)d &= \int_0^1 F'(x + \alpha d)d\mathrm{d}\alpha - F'(x)d \\ &= \int_0^1 [F'(x + \alpha d) - F'(x)]d\mathrm{d}\alpha. \end{aligned}$$

Hence

$$\begin{aligned} \|F(x + d) - F(x) - F'(x)d\| &\le \int_0^1 \|F'(x + \alpha d) - F'(x)\|\|d\|\mathrm{d}\alpha \\ &\le \int_0^1 \gamma\|\alpha d\|\|d\|\mathrm{d}\alpha \\ &= \gamma\|d\|^2 \int_0^1 \alpha\mathrm{d}\alpha \\ &= \frac{\gamma}{2}\|d\|^2. \quad \square \end{aligned}$$

Theorem 1.2.23 *Let $f : R^n \to R$ be twice continuously differentiable in the open convex set $D \subset R^n$. Let $\nabla^2 f(x)$ be Lipschitz continuous at $x \in D$ with Lipschitz constant γ. Then for any $x + d \in D$, we have*

$$\left| f(x+d) - [f(x) + \nabla f(x)^T d + \frac{1}{2} d^T \nabla^2 f(x) d] \right| \leq \frac{\gamma}{6} \|d\|^3. \tag{1.2.110}$$

The proof of this theorem is left to readers as an exercise.

As a generalization of Theorem 1.2.22, we obtain

Theorem 1.2.24 *Let $F : R^n \to R^m$ be continuously differentiable in the open convex set $D \subset R^n$. Then for any $u, v, x \in D$, we have*

$$\begin{aligned} & \|F(u) - F(v) - F'(x)(u-v)\| \\ \leq & \left[\sup_{0 \leq t \leq 1} \|F'(v + t(u-v)) - F'(x)\| \right] \|u - v\|. \end{aligned} \tag{1.2.111}$$

Furthermore, assume that F' is Lipschitz continuous in D, then

$$\|F(u) - F(v) - F'(x)(u-v)\| \leq \gamma \sigma(u,v) \|u - v\| \tag{1.2.112}$$

and

$$\|F(u) - F(v) - F'(x)(u-v)\| \leq \gamma \frac{\|u - x\| + \|v - x\|}{2} \|u - v\|, \tag{1.2.113}$$

where $\sigma(u,v) = \max\{\|u - x\|, \|v - x\|\}$.

Proof. By (1.2.106) and the mean-value theorem of integration, we have

$$\begin{aligned} & \|F(u) - F(v) - F'(x)(u-v)\| \\ = & \left\| \int_0^1 [F'(v + t(u-v)) - F'(x)](u-v) \mathrm{d}t \right\| \\ \leq & \int_0^1 \|F'(v + t(u-v)) - F'(x)\| \|u - v\| \mathrm{d}t \\ \leq & \left[\sup_{0 \leq t \leq 1} \|F'(v + t(u-v)) - F'(x)\| \right] \|u - v\| \end{aligned}$$

which is (1.2.111). Also since F' is Lipschitz continuous in D, we proceed with the above inequality and get

$$\|F(u) - F(v) - F'(x)(u-v)\|$$

$$\begin{aligned} &\leq \gamma \int_0^1 \|v + t(u-v) - x\| \|u - v\| \mathrm{d}t \\ &\leq \gamma \sup_{0 \leq t \leq 1} \|v + t(u-v) - x\| \|u - v\| \\ &= \gamma \sigma(u, v) \|u - v\| \end{aligned}$$

which is (1.2.112). Similarly, we can derive (1.2.113) which is left as an exercise. □

The following theorem is useful, giving a relation between $\|F(u) - F(v)\|$ and $\|u - v\|$.

Theorem 1.2.25 *Let F and F' satisfy the conditions of Theorem 1.2.24. Assume that $[F'(x)]^{-1}$ exists. Then there exist $\epsilon > 0$ and $\beta > \alpha > 0$ such that for all $u, v \in D$, when $\max\{\|u - x\|, \|v - x\|\} \leq \epsilon$, we have*

$$\alpha \|u - v\| \leq \|F(u) - F(v)\| \leq \beta \|u - v\|. \tag{1.2.114}$$

Proof. By the triangle inequality and (1.2.112),

$$\begin{aligned} \|F(u) - F(v)\| &\leq \|F'(x)(u - v)\| + \|F(u) - F(v) - F'(x)(u - v)\| \\ &\leq (\|F'(x)\| + \gamma\sigma(u, v)) \|u - v\| \\ &\leq \left(\|F'(x)\| + \gamma\epsilon\right) \|u - v\|. \end{aligned}$$

Set $\beta = \|F'(x)\| + \gamma\epsilon$, we obtain the right inequality of (1.2.114). Similarly,

$$\begin{aligned} \|F(u) - F(v)\| &\geq \|F'(x)(u - v)\| - \|F(u) - F(v) - F'(x)(u - v)\| \\ &\geq \left[1/\|[F'(x)]^{-1}\| - \gamma\sigma(u, v)\right] \|u - v\| \\ &\geq [1/\|[F'(x)]^{-1}\| - \gamma\epsilon] \|u - v\|. \end{aligned}$$

Hence, if $\frac{1}{\|[F'(x)]^{-1}\|\gamma} > \epsilon$, the left inequality of (1.2.114) also holds with

$$\alpha = \frac{1}{\|[F'(x)]^{-1}\|} - \gamma\epsilon > 0. \quad \square$$

Corollary 1.2.26 *Let F and F' satisfy the conditions of Theorem 1.2.22. When u and v are sufficiently close to x, we have*

$$\limsup_{\omega \to 0} \frac{\|u - x\|}{\|v - x\|} \leq C \limsup_{\omega \to 0} \frac{\|F(u) - F(x)\|}{\|F(v) - F(x)\|}, \tag{1.2.115}$$

where $C = \|F'(x)\| \|F'(x)^{-1}\|$ is a constant and $\omega = \max\{\|u - x\|, \|v - x\|\}$.

Proof. By using Theorem 1.2.22, we have

$$\begin{aligned}\|F(v)-F(x)\| &\leq \|F'(x)(v-x)\| + \|F(v)-F(x)-F'(x)(v-x)\| \\ &\leq \|F'(x)\|\|v-x\| + O(\|v-x\|^2)\end{aligned}$$

and

$$\begin{aligned}\|F(u)-F(x)\| &\geq \|F'(x)(u-x)\| - \|F(u)-F(x)-F'(x)(u-x)\| \\ &\geq \|u-x\|/\|[F'(x)]^{-1}\| + O(\|u-x\|^2).\end{aligned}$$

Then

$$\frac{\|F(u)-F(x)\|}{\|F(v)-F(x)\|} \geq \frac{\|u-x\|/\|F'(x)^{-1}\| + O(\|u-x\|^2)}{\|F'(x)\|\|v-x\| + O(\|v-x\|^2)}.$$

Setting $C = \|F'(x)\|\|F'(x)^{-1}\|$ and taking limit give

$$C \limsup_{\omega\to 0} \frac{\|F(u)-F(x)\|}{\|F(v)-F(x)\|} \geq \limsup_{\omega\to 0} \frac{\|u-x\|}{\|v-x\|},$$

where $\omega = \max\{\|u-x\|, \|v-x\|\}$. □

We conclude this subsection with some remarks about differentiation of the vector-valued functions.

About the calculus of vector-valued functions, we would like to review Gateaux and Fréchet derivatives.

Definition 1.2.27 *Let $D \subset R^n$ be an open set. The function $F : D \subset R^n \to R^m$ is Gateaux- (or G-) differentiable at $x \in D$ if there exists a linear operator $A \in L(R^n, R^m)$ such that for any $d \in R^n$,*

$$\lim_{\alpha\to 0} \frac{1}{\alpha}\|F(x+\alpha d) - F(x) - \alpha A d\| = 0. \tag{1.2.116}$$

The linear operator A is denoted by $F'(x)$ and is called the G-derivative of F at x.

Definition 1.2.28 *Let $D \subset R^n$ be an open set. The function $F : R^n \to R^m$ is Fréchet- (or F-) differentiable at $x \in D$ if there is a linear operator $A \in L(R^n, R^m)$ such that for any $d \in R^n$,*

$$\lim_{d\to 0} \frac{\|F(x+d) - F(x) - Ad\|}{\|d\|} = 0. \tag{1.2.117}$$

The linear operator A is again denoted by $F'(x)$, and is called the F-derivative of F at x.

The F-differentiability can also be written as

$$F(x+d) - F(x) = F'(x)d + o(\|d\|).$$

Furthermore, if for any $u, v \in R^n$,

$$\lim_{\|u-v\|\to 0} \frac{\|F(u) - F(v) - F'(x)(u-v)\|}{\|u-v\|} = 0, \tag{1.2.118}$$

then F is called strongly F-differentiable at $x \in D$ and $F'(x)$ is called a strong F-derivative.

From the above two definitions, we know the following facts.

1. If $F : R^n \to R^m$ is continuous at $x \in R^n$, then F is hemi-continuous at x.

2. If $F : R^n \to R^m$ is G-differentiable at $x \in D$, then F is hemi-continuous at x.

3. If $F : R^n \to R^m$ is F-differentiable at $x \in D$, then F is continuous at x.

4. If F is F-differentiable at $x \in D$, then it is G-differentiable at x; however, the reverse is not true.

5. If F is G-differentiable and its G-derivative F' is continuous, then F is F-differentiable and the F-derivative is continuous. In this case, we say that F is continuously differentiable.

6. The G-derivative and F-derivative of F, if they exist, are equal and given by the Jacobian matrix

$$F'(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x) & \frac{\partial f_1}{\partial x_2}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_m}{\partial x_1}(x) & \frac{\partial f_m}{\partial x_2}(x) & \cdots & \frac{\partial f_m}{\partial x_n}(x) \end{bmatrix},$$

where $f_1, f_2, \cdots, f_m$ are components of F.

7. The mean-value theorem: Let $F : R^n \to R^m$ be G-differentiable in the open convex set $D \subset R^n$. Then we have the following forms of the mean-value theorem:

   (a) For any $x, y, z \in D$, there exist $t_1, t_2, \cdots, t_m \in [0, 1]$ such that

$$F(y) - F(x) = \begin{pmatrix} f_1'(x + t_1(y - x)) \\ f_2'(x + t_2(y - x)) \\ \vdots \\ f_m'(x + t_m(y - x)) \end{pmatrix} \tag{1.2.119}$$

   and

$$\|F(y) - F(x)\| \leq \sup_{0 \leq t \leq 1} \|F'(x + t(y - x))\| \|y - x\|.$$

   (b) For any $x, y, z \in D$,

$$\|F(y)-F(z)-F'(x)(y-z)\| \leq \sup_{0 \leq t \leq 1} \|F'(z+t(y-z))-F'(x)\| \|y-z\|. \tag{1.2.120}$$

   (c) Furthermore, if the G-derivative F' is hemi-continuous on D, then for any $x, y \in D$,

$$F(y) - F(x) = \int_0^1 F'(x + t(y - x))(y - x) \mathrm{d}t. \tag{1.2.121}$$

   (d) If assume also that $F'(x)$ is Hölder continuous on D, then for all $x, y \in D$,

$$\|F(y) - F(x) - F'(x)(y - x)\| \leq \frac{\gamma}{p + 1} \|y - x\|^{p+1}. \tag{1.2.122}$$

   If $p = 1$, it is just (1.2.109).

1.3 Convex Sets and Convex Functions

Convex sets and convex functions play an important role in the study of optimization. In this section, we introduce the fundamental concepts and results of convex sets and convex functions.

1.3.1 Convex Sets

Definition 1.3.1 *Let the set $S \subset R^n$. If, for any $x_1, x_2 \in S$, we have*

$$\alpha x_1 + (1-\alpha)x_2 \in S, \ \forall \alpha \in [0,1], \tag{1.3.1}$$

then S is said to be a convex set.

This definition indicates, in geometry, that for any two points $x_1, x_2 \in S$, the line segment joining x_1 and x_2 is entirely contained in S. It also states that S is path-connected, i.e., two arbitrary points in S can be linked by a continuous path.

It can be shown by induction that the set $S \subset R^n$ is convex if and only if for any $x_1, x_2, \cdots, x_m \in S$,

$$\sum_{i=1}^{m} \alpha_i x_i \in S, \tag{1.3.2}$$

where $\sum_{i=1}^{m} \alpha_i = 1, \alpha_i \geq 0, i = 1, \cdots, m$.

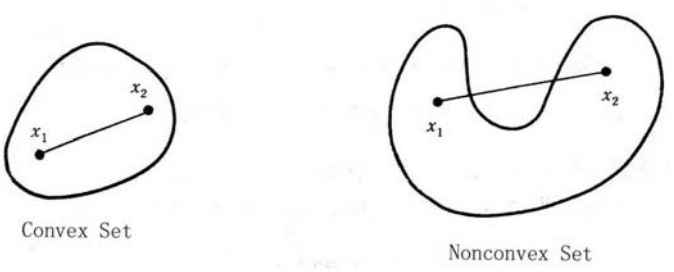


Figure 1.3.1 Convex set and nonconvex set

In (1.3.1), $x = \alpha x_1 + (1-\alpha)x_2$, where $\alpha \in [0,1]$, is called a convex combination of x_1 and x_2. In (1.3.2), $x = \sum_{i=1}^{m} \alpha_i x_i$ is called a convex combination of $x_1, \cdots, x_m$, where $\sum_{i=1}^{m} \alpha_i = 1, \alpha_i \geq 0, i = 1, \cdots, m$.

Example 1.3.2 *The hyperplane $H = \{x \in R^n \mid p^T x = \alpha\}$ is a convex set, where $p \in R^n$ is a nonzero vector referred to as the normal vector to the hyperplane, and α is a scalar.*

In fact, for any $x_1, x_2 \in H$ and each $\theta \in [0,1]$,

$$p^T[\theta x_1 + (1-\theta)x_2] = \alpha,$$

then $\theta x_1 + (1-\theta)x_2 \in H$.

In the hyperplane $H = \{x \in R^n \mid p^T x = \alpha\}$, if $\alpha = 0$, it can be reduced to a subspace of vectors that are orthogonal to p.

Similarly, the closed half space $H^- = \{x \in R^n \mid p^T x \leq \beta\}$ and $H^+ = \{x \in R^n \mid p^T x \geq \beta\}$ are closed convex sets. The open half space $(\overset{\circ}{H})^- = \{x \in R^n \mid p^T x < \beta\}$ and $(\overset{\circ}{H})^+ = \{x \in R^n \mid p^T x > \beta\}$ are open convex sets .

Example 1.3.3 *The ray $S = \{x \in R^n \mid x = x_0 + \lambda d, \lambda \geq 0\}$ is a convex set, where $d \in R^n$ is a nonzero vector, and $x_0 \in R^n$ is a fixed point.*

In fact, for any $x_1, x_2 \in S$ and each $\lambda \in [0,1]$, we have

$$x_1 = x_0 + \lambda_1 d,\ x_2 = x_0 + \lambda_2 d,$$

where $\lambda_1, \lambda_2 \in [0,1]$. Hence

$$\begin{aligned} \lambda x_1 + (1-\lambda)x_2 &= \lambda(x_0 + \lambda_1 d) + (1-\lambda)(x_0 + \lambda_2 d) \\ &= x_0 + [\lambda\lambda_1 + (1-\lambda)\lambda_2]d. \end{aligned}$$

Since $\lambda\lambda_1 + (1-\lambda)\lambda_2 \geq 0$, then $\lambda x_1 + (1-\lambda)x_2 \in S$.

The finite intersection of closed half spaces

$$S = \{x \in R^n \mid p_i^T x \leq \beta_i,\ i = 1, \cdots, m\},$$

is called a polyhedral set, where p_i is a nonzero vector, β_i a scalar. The polyhedral is a convex set .

Since an equality can be represented by two inequalities, the following sets are examples of polyhedral sets:

$$S = \{x \in R^n \mid Ax = b, x \geq 0\},$$

$$S = \{x \in R^n \mid Ax \geq 0, x \geq 0\}.$$

The theorems below state the algebraic properties and topological properties. That is the intersection of two convex sets is convex, the algebraic sum of two convex sets is convex, the interior of a convex set is convex, and the closure of a convex set is convex.

Theorem 1.3.4 *Let S_1 and S_2 be convex sets in R^n. Then*

1. *$S_1 \cap S_2$ is convex;*

2. *$S_1 \pm S_2 = \{x_1 \pm x_2 \mid x_1 \in S_1, x_2 \in S_2\}$ is convex.*

Proof. The proof is immediate from the definition of convex set and left to readers as an exercise. □

From this theorem, we know that the feasible regions in linear programming and quadratic programming are convex sets, because they are the intersection of a hyperplane and a half space.

Theorem 1.3.5 *Let $S \subset R^n$ be a convex set. Then*

1. *the interior int$S$ of S is a convex set;*

2. *the closure $\bar{S}$ of S is a convex set.*

Proof. 1) Let x and x' be in int$S$, and $x'' = \alpha x + (1-\alpha)x', \alpha \in (0,1)$. Choose $\delta > 0$ such that $B(x', \delta) \subset S$, where $B(x', \delta)$ is the δ-neighborhood of x'. It is easy to see that $\|x'' - x\|/\|x' - x\| = 1 - \alpha$. We know that $B(x'', (1-\alpha)\delta)$ is just the set $\alpha x + (1-\alpha)B(x', \delta)$ which is in S. Therefore $B(x'', (1-\alpha)\delta) \subset S$ which shows that $x'' \in$ int S.

2) Take $x, x' \in \bar{S}$. Select in S two sequences $\{x_k\}$ and $\{x'_k\}$ converging to x and x' respectively. Then, for $\alpha \in [0,1]$, we have

$$\begin{aligned} & \|[\alpha x_k + (1-\alpha)x'_k] - [\alpha x + (1-\alpha)x']\| \\ = \; & \|\alpha(x_k - x) + (1-\alpha)(x'_k - x')\| \\ \leq \; & \alpha\|x_k - x\| + (1-\alpha)\|x'_k - x'\|. \end{aligned}$$

Taking the limit yields

$$\lim_{k\to\infty} \|[\alpha x_k + (1-\alpha)x'_k] - [\alpha x + (1-\alpha)x']\| = 0,$$

which shows $\alpha x + (1-\alpha)x' \in \bar{S}$. □

Now we state some concepts related to convex sets.

Let $S \subset R^n$ be a nonempty set. We define the convex hull conv(S) as the intersection of all convex sets containing S, which is described as the set of

all convex combinations of the elements of S:

$$\begin{aligned} \text{conv}(S) &\triangleq \cap\{C \mid C \text{ is convex and contains } S\} \\ &= \{x \in R^n \mid x = \sum_{i=1}^{m} \alpha_i x_i, x_i \in S, \sum_{i=1}^{m} \alpha_i = 1, \\ &\quad \alpha_i \geq 0, i = 1, \cdots, m\}. \end{aligned} \tag{1.3.3}$$

We can see that conv(S) is the smallest convex set containing S.

A nonempty set $C \subset R^n$ is called a cone if it is closed under positive scalar multiplication, i.e., if $x \in C$ implies that $\lambda x \in C$ for all $\lambda > 0$. If, in addition, C is convex, then C is called a convex cone. $C \subset R^n$ is a convex cone if and only if it is closed under addition and positive scalar multiplication. The smallest convex cone containing convex S is

$$C = \{\lambda x \mid \lambda > 0, x \in S\}.$$

The following are examples of convex cones. For example, the nonnegative orthant of R^n

$$\{x = (\xi_1, \cdots, \xi_n) \,|\, \xi_1 \geq 0, \cdots, \xi_n \geq 0\},$$

positive orthant of R^n

$$\{x = (\xi_1, \cdots, \xi_n) \,|\, \xi_1 > 0, \cdots, \xi_n > 0\}$$

and the intersection of m half-spaces

$$\{x \in R^n \,|\, x^T b_i \leq 0,\, b_i \in R^n, i = 1, \cdots, m\}$$

are convex cones .

A specially important class of convex cones is polar cone. Let S be a nonempty set in R^n. The polar cone of S, denoted by S^*, is given by $\{p \,|\, p^T x \leq 0$ for all $x \in S\}$. It is easy to see from the above definition that the polar cone S^* of a nonempty set S has the following properties:

1. S^* is a closed convex cone.

2. $S \subset S^{**}$, where S^{**} is the polar cone of S^*. If S is a nonempty closed convex set, then $S^{**} = S$.

3. If S_1, S_2 are nonempty sets, then $S_1 \subset S_2$ implies $S_2^* \subset S_1^*$.

The normal and tangent cones play a special role in constrained optimization. Here we give their definitions below. Let S be a closed convex set. The normal cone of S at $\bar{x}$ is defined as

$$N(\bar{x}) = \{y \in R^n \mid \langle y, x - \bar{x}\rangle \leq 0, \forall x \in S\}. \tag{1.3.4}$$

The tangent cone of S at $\bar{x} \in S$ is the polar of the normal cone at $\bar{x}$, that is

$$\begin{aligned} T(\bar{x}) = (N(\bar{x}))^* &= \text{cl}\{\lambda(x - \bar{x}) \mid \lambda \geq 0, x \in S\} \\ &= \{d \mid d = \lim_{x \to \bar{x}} \lambda(x - \bar{x}),\ \lambda \geq 0, x \in S\}, \end{aligned} \tag{1.3.5}$$

where $\text{cl}\{S\}$ denotes the closure of S.

1.3.2 Convex Functions

Definition 1.3.6 *Let $S \subset R^n$ be a nonempty convex set. Let $f : S \subset R^n \to R$. If, for any $x_1, x_2 \in S$ and all $\alpha \in (0, 1)$, we have*

$$f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2), \tag{1.3.6}$$

then f is said to be convex on S. If the above inequality is true as a strict inequality for all $x_1 \neq x_2$, i.e.,

$$f(\alpha x_1 + (1 - \alpha)x_2) < \alpha f(x_1) + (1 - \alpha)f(x_2), \tag{1.3.7}$$

then f is called a strict convex function on S. If there is a constant $c > 0$ such that for any $x_1, x_2 \in S$,

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha)f(x_2) - \frac{1}{2}c\alpha(1-\alpha)\|x_1 - x_2\|^2, \tag{1.3.8}$$

then f is called a uniformly (or strongly) convex function on S.

If $-f$ is a convex (strictly convex, uniformly convex) function on S, then f is said to be a concave (strictly concave, uniformly concave) function.

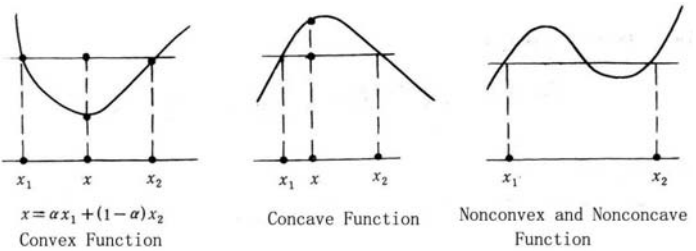


Figure 1.3.2 Convex function and concave function

Figure 1.3.2 gives examples of convex, concave, and neither convex nor concave functions. The geometrical interpretation of a convex function says that the function values are below the corresponding chord, that is, the values of a convex function at points on the line segment $\alpha x_1 + (1-\alpha)x_2$ are less than or equal to the height of the chord joining the points $(x_1, f(x_1))$ and $(x_2, f(x_2))$. It is obvious from the definition of convex function that a linear function $f(x) = a^T x + \beta$ is both a convex and concave function on R^n, where $a, x \in R^n, \beta \in R$.

The other basic and important examples of convex functions are indicator function, support function, norm and distance function.

Let $S \subset R^n$ be a nonempty subset; the indicator function $I_S : R^n \to R \cup \{+\infty\}$ is defined by

$$I_S(x) := \begin{cases} 0, & \text{if } x \in S, \\ +\infty, & \text{otherwise.} \end{cases} \tag{1.3.9}$$

Clearly, I_S is convex if and only if S is convex.

Let $S \subset R^n$ be a nonempty subset. The support function of S is defined by

$$\sigma_S(s) := \sup\{\langle s, x\rangle \mid x \in S\}. \tag{1.3.10}$$

This is a convex function.

It is easy to see that a norm on R^n is a convex function. If we define the distance function as

$$d_S(x) := \inf\{\|y - x\| \mid y \in S\},$$

where $S \subset R^n$ is a nonempty convex set and $\|\cdot\|$ is any norm on R^n, then d_S is a convex function.

A convex function can also be described by an epigraph. Now we first give the definition of the epigraph of f, and then show that f is convex if and only if its epigraph is a convex set.

Let $S \subset R^n$ be a nonempty set. A set $\{(x, f(x)) : x \in S\} \subset R^{n+1}$ describing the function f is said to be the graph of the function f. Related to the graph of f, there are the epigraph, which consists of points above the graph of f, and the hypograph, which consists of points below the graph of f.

Definition 1.3.7 *Let $S \subset R^n$ be a nonempty set. Let $f : S \subset R^n \to R$. The epigraph of f, denoted by epi$f$, is a subset of R^{n+1} defined by*

$$epif = \{(x, \alpha) \,|\, f(x) \leq \alpha, x \in S, \alpha \in R\}. \tag{1.3.11}$$

The hypograph of f, denoted by hyp$f$, is a subset of R^{n+1} defined by

$$hypf = \{(x, \alpha) \,|\, f(x) \geq \alpha, x \in S, \alpha \in R\}. \tag{1.3.12}$$

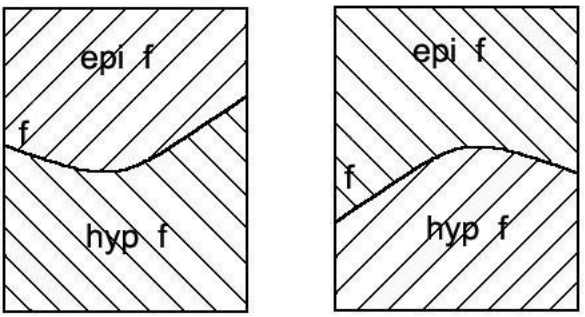


Figure 1.3.3 Epigraph and hypograph

The following theorem indicates the relation between convex function and convexity of epi$f$.

Theorem 1.3.8 *Let $S \subset R^n$ be a nonempty convex set. Let $f : S \subset R^n \to R$. Then f is convex if and only if epi$f$ is a convex set.*

Proof. Assume that f is convex. Let $x_1, x_2 \in S$ and $(x_1, \alpha_1), (x_2, \alpha_2)$ be in epi$f$. Then, it follows from Definition 1.3.6 and Definition 1.3.7 that

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2) \leq \lambda\alpha_1 + (1-\lambda)\alpha_2$$

for any $\lambda \in (0, 1)$. Since S is a convex set, $\lambda x_1 + (1-\lambda)x_2 \in S$. Hence $(\lambda x_1 + (1-\lambda)x_2, \lambda\alpha_1 + (1-\lambda)\alpha_2) \in$ epi f, which means epi$f$ is convex.

Conversely, assume that epi$f$ is convex, and let $x_1, x_2 \in S$ and $(x_1, f(x_1))$, $(x_2, f(x_2)) \in$ epi$f$. Then we have from the convexity of epi$f$ that

$$(\lambda x_1 + (1-\lambda)x_2, \lambda f(x_1) + (1-\lambda) f(x_2)) \in \text{epi}f, \text{ for } \lambda \in (0, 1).$$

This means

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

for each $\lambda \in (0,1)$. Hence f is convex. $\square$

The epigraph epi$f$ of a function f is an important concept and it is used often in convex programming. Here we would like to mention its properties. The epi$f$ has a closed relation with the lower semi-continuity (l.s.c.) of f which is also very important, because, for a function to have a minimum, a very basic requirement is lower semi-continuity. We may recall that a function f is lower semi-continuous if, for each $x \in R^n$,

$$\liminf_{y \to x} f(y) \geq f(x). \tag{1.3.13}$$

The following theorem gives an equivalent property between epi$f$ and l.s.c.

Theorem 1.3.9 *For $f : R^n \to R \cup \{+\infty\}$, the following three statements are equivalent:*

1. *f is lower semi-continuous on R^n;*
2. *epi$f$ is a closed set in $R^n \times R$;*
3. *the level sets $L_r(f) = \{x \in R^n \mid f(x) \leq r, r \in R\}$ are closed for all $r \in R$.*

Proof. (1) $\Rightarrow$ (2): Let $\{(y_k, r_k)\}$ be a sequence of epi$f$ converging to (x, r) for $k \to \infty$. Since $f(y_k) \leq r_k$ for all k, the (1.3.13) gives

$$r = \lim_{k \to \infty} r_k \geq \liminf_{y_k \to x} f(y_k) \geq f(x),$$

which indicates that $(x, r) \in$ epi$f$.

(2) $\Rightarrow$ (3): Construct the level set $L_r(f)$ which is the intersection of two closed sets epi$f$ and $(R^n \times \{r\})$. Obviously the intersection is closed.

(3) $\Rightarrow$ (1): Suppose that f is not lower semi-continuous at some x, which means there exists a sequence $\{y_k\}$ converging to x such that $\{f(y_k)\}$ converges to $\rho < f(x) \leq +\infty$. Take $r \in (\rho, f(x))$. When k tends large enough, we have $f(y_k) \leq r < f(x)$ which means that $L_r(f)$ does not contain its limit x. Hence $L_r(f)$ is not closed. $\square$

Using Theorem 1.3.9, we can give a definition of closed function.

Definition 1.3.10 *A function $f : R^n \to R \cup \{+\infty\}$ is said to be closed if it is lower semi-continuous everywhere, or if its epigraph is closed, or if its level sets are closed.*

Obviously, the indicator function I_S is closed if and only if S is closed. Also, epi$I_S = S \times R^+$. The support function σ_S is closed too.

Next, we give some properties of convex functions.

Theorem 1.3.11 *1. Let f be a convex function on a convex set $S \subset R^n$ and real number $\alpha \geq 0$, then αf is also a convex function on S.*

2. Let f_1, f_2 be convex functions on a convex set S, then $f_1 + f_2$ is also a convex function on S.

3. Let $f_1, f_2, \cdots, f_m$ be convex functions on a convex set S and real numbers $\alpha_1, \alpha_2, \cdots, \alpha_m \geq 0$, then $\sum_{i=1}^m \alpha_i f_i$ is also a convex function on S.

Proof. We only prove the second statement. The others are similar.

Let $x_1, x_2 \in S$ and $0 < \alpha < 1$, then

$$\begin{aligned} & f_1(\alpha x_1 + (1-\alpha)x_2) + f_2(\alpha x_1 + (1-\alpha)x_2) \\ \leq \; & \alpha[f_1(x_1) + f_2(x_1)] + (1-\alpha)[f_1(x_2) + f_2(x_2)]. \; \square \end{aligned}$$

Continuity is an important property of a convex function. However, it is not sure that a convex function whose domain is not open is continuous. The following theorem shows that a convex function is continuous on an open convex set or the interior of its domain.

Theorem 1.3.12 *Let $S \subset D$ be an open convex set. Let $f : D \subset R^n \to R$ be convex. Then f is continuous on S.*

Proof. Let x_0 be an arbitrary point in S. Since S is an open convex set, we can find $n+1$ points $x_1, \cdots, x_{n+1} \in S$ such that the interior of the convex hull

$$C = \{x \mid x = \sum_{i=1}^{n+1} \alpha_i x_i, \alpha_i \geq 0, \sum_{i=1}^{n+1} \alpha_i = 1\}$$

is not empty and $x_0 \in \text{int}C$.

Now let $\alpha = \max_{1\le i\le n+1} f(x_i)$, then

$$f(x) = f\left(\sum_{i=1}^{n+1} \alpha_i x_i\right) \le \sum_{i=1}^{n+1} \alpha_i f(x_i) \le \alpha,\ \forall x \in C, \tag{1.3.14}$$

so that f is bounded over C. Also, since $x_0 \in \text{int } C$, there is a $\delta > 0$ such that $B(x_0, \delta) \subset C$, where $B(x_0, \delta) = \{x|\ \|x - x_0\| \le \delta\}$. Hence for arbitrary $h \in B(0, \delta)$ and $\lambda \in [0, 1]$, we have

$$x_0 = \frac{1}{1+\lambda}(x_0 + \lambda h) + \frac{\lambda}{1+\lambda}(x_0 - h). \tag{1.3.15}$$

Since f is convex on C, then

$$f(x_0) \le \frac{1}{1+\lambda} f(x_0 + \lambda h) + \frac{\lambda}{1+\lambda} f(x_0 - h). \tag{1.3.16}$$

By (1.3.16) and (1.3.14), we have

$$f(x_0 + \lambda h) - f(x_0) \ge \lambda(f(x_0) - f(x_0 - h)) \ge -\lambda(\alpha - f(x_0)). \tag{1.3.17}$$

On the other hand,

$$f(x_0 + \lambda h) = f(\lambda(x_0 + h) + (1 - \lambda)x_0) \le \lambda f(x_0 + h) + (1 - \lambda) f(x_0),$$

which is

$$f(x_0 + \lambda h) - f(x_0) \le \lambda(f(x_0 + h) - f(x_0)) \le \lambda(\alpha - f(x_0)). \tag{1.3.18}$$

Therefore, (1.3.17) and (1.3.18) give

$$|f(x_0 + \lambda h) - f(x_0)| \le \lambda |f(x_0) - \alpha|. \tag{1.3.19}$$

Now, for given $\epsilon > 0$, choose $\delta' \le \delta$ so that $\delta' |f(x_0) - \alpha| \le \epsilon\delta$. Set $d = \lambda h$ with $\|h\| = \delta$, then $d \in B(0, \delta)$ and

$$|f(x_0 + d) - f(x_0)| \le \epsilon. \qquad \square$$

If a convex function is differentiable, we can describe the characterization of differential convex functions. The following theorem gives the first order characterization of differential convex functions.

Theorem 1.3.13 *Let $S \subset R^n$ be a nonempty open convex set and let $f : S \subset R^n \to R$ be a differentiable function. Then f is convex if and only if*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x), \ \forall x, y \in S. \tag{1.3.20}$$

Similarly, f is strictly convex on S if and only if

$$f(y) > f(x) + \nabla f(x)^T (y - x), \ \forall x, y \in S, y \neq x. \tag{1.3.21}$$

Furthermore, f is strongly (or uniformly) convex if and only if

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{1}{2} c \|y - x\|^2, \ \forall x, y \in S, \tag{1.3.22}$$

where $c > 0$ is a constant.

Proof. Necessity: Let $f(x)$ be a convex function, then for all α with $0 < \alpha < 1$,

$$f(\alpha y + (1 - \alpha)x) \leq \alpha f(y) + (1 - \alpha) f(x).$$

Hence,

$$\frac{f(x + \alpha(y - x)) - f(x)}{\alpha} \leq f(y) - f(x).$$

Setting $\alpha \to 0$ yields

$$\nabla f(x)^T (y - x) \leq f(y) - f(x).$$

Sufficiency: Assume that (1.3.20) holds. Pick any $x_1, x_2 \in S$ and set $x = \alpha x_1 + (1 - \alpha) x_2, 0 < \alpha < 1$. Then

$$\begin{aligned} f(x_1) &\geq f(x) + \nabla f(x)^T (x_1 - x), \\ f(x_2) &\geq f(x) + \nabla f(x)^T (x_2 - x). \end{aligned}$$

Hence

$$\begin{aligned} \alpha f(x_1) + (1 - \alpha) f(x_2) &\geq f(x) + \nabla f(x)^T (\alpha x_1 + (1 - \alpha) x_2 - x) \\ &= f(\alpha x_1 + (1 - \alpha) x_2), \end{aligned}$$

which indicates that $f(x)$ is a convex function.

Similarly, we can prove (1.3.21) and (1.3.22) by use of (1.3.20). For example, from the definition of the strictly convex, we have

$$f(x + \alpha(y - x)) - f(x) < \alpha(f(y) - f(x)).$$

Then, using (1.3.20) and the above inequality, we have

$$\langle \nabla f(x), \alpha(y-x) \rangle \leq f(x+\alpha(y-x)) - f(x) < \alpha(f(y)-f(x))$$

which is the required (1.3.21).

To obtain (1.3.22), it is enough to apply (1.3.20) to the function $f - \frac{1}{2}c\|\cdot\|^2$. □

Definition 1.3.6 of convex function indicates that the function value is below the chord, which means that the linear interpolation of the function values at two points is larger than the function value at the interpolation point. This theorem represents that the linear approximation based on a local derivative is a lower estimate, i.e., the convex function always lies above its tangent at any point. Such a tangent is called a supporting hyperplane of the convex function.

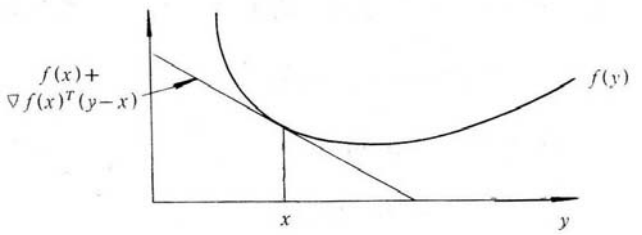


Figure 1.3.4 The first order characteristic of a convex function

Below, we consider the second order characteristic of a twice continuously differentiable convex function.

Theorem 1.3.14 *Let $S \subset R^n$ be a nonempty open convex set, and let $f : S \subset R^n \to R$ be twice continuously differentiable. Then*

1. *f is convex if and only if its Hessian matrix is positive semidefinite at each point in S.*

2. *f is strictly convex if its Hessian matrix is positive definite at each point in S.*

3. *f is uniformly convex if and only if its Hessian matrix is uniformly positive definite at each point in S, i.e., there exists a constant $m > 0$ such that*

$$m\|u\|^2 \leq u^T \nabla^2 f(x) u, \ \forall x \in S, \ u \in R^n.$$

Proof. We only prove the first case. The other two cases are analogous.

Sufficiency. Suppose that the Hessian matrix $\nabla^2 f(x)$ is positive semidefinite at each point $x \in S$. Consider $x, \bar{x} \in S$. By the mean-value theorem, we have

$$f(x) = f(\bar{x}) + \nabla f(\bar{x})^T (x - \bar{x}) + \frac{1}{2}(x - \bar{x})^T \nabla^2 f(\hat{x})(x - \bar{x}),$$

where $\hat{x} = \bar{x} + \theta(x - \bar{x}), \theta \in (0, 1)$. Noting that $\hat{x} \in S$, it follows from the assumption that

$$f(x) \geq f(\bar{x}) + \nabla f(\bar{x})^T (x - \bar{x}).$$

Hence f is a convex function by Theorem 1.3.13.

Necessity. Suppose that f is a convex function and let $\bar{x} \in S$. We need to prove $p^T \nabla^2 f(\bar{x}) p \geq 0, \forall p \in R^n$. Since S is open, then there exists $\delta > 0$ such that when $|\lambda| < \delta, \bar{x} + \lambda p \in S$. By Theorem 1.3.13,

$$f(\bar{x} + \lambda p) \geq f(\bar{x}) + \lambda \nabla f(\bar{x})^T p. \tag{1.3.23}$$

Also since $f(x)$ is twice differentiable at $\bar{x}$, then

$$f(\bar{x} + \lambda p) = f(\bar{x}) + \lambda \nabla f(\bar{x})^T p + \frac{\lambda^2}{2} p^T \nabla^2 f(\bar{x}) p + o(\|\lambda p\|^2). \tag{1.3.24}$$

Substituting (1.3.24) into (1.3.23) yields

$$\frac{1}{2}\lambda^2 p^T \nabla^2 f(\bar{x}) p + o(\|\lambda p\|^2) \geq 0.$$

Dividing by λ^2 and letting $\lambda \to 0$, it follows that

$$p^T \nabla^2 f(\bar{x}) p \geq 0. \square$$

Next, we would like to characterize a convex function with monotonicity which is very useful.

We first introduce a definition of monotone mapping.

Definition 1.3.15 *A mapping $F : D \subset R^n \to R^n$ is monotone on $D_0 \subset D$ if*

$$\langle F(x) - F(y), x - y \rangle \geq 0, \ \forall x, y \in D_0; \tag{1.3.25}$$

F is strictly monotone on D_0 if

$$\langle F(x) - F(y), x - y \rangle > 0, \ \forall x, y \in D_0, x \neq y; \tag{1.3.26}$$

F is uniformly (or strongly) monotone if there is a constant $c > 0$ so that

$$\langle F(x) - F(y), x - y\rangle \geq c\|x - y\|^2, \ \forall x, y \in D_0. \tag{1.3.27}$$

If we let $F = \nabla f$ in the above definition, we can get the following theorem which says that, for convex function f, its gradient ∇f is a monotone mapping.

Theorem 1.3.16 *Suppose that $f : D \subset R^n \to R$ is differentiable on an open set D, and that $S \subset D$ is a convex subset. Then f is convex on S if and only if its gradient ∇f is monotone, i.e.,*

$$\langle \nabla f(x) - \nabla f(y), x - y\rangle \geq 0, \ \forall x, y \in S; \tag{1.3.28}$$

and f is strictly convex on S if and only if its gradient ∇f is strictly monotone, i.e.,

$$\langle \nabla f(x) - \nabla f(y), x - y\rangle > 0, \ \forall x, y \in S, x \neq y; \tag{1.3.29}$$

finally, f is uniformly (or strongly) convex on S if and only if its gradient ∇f is uniformly monotone, i.e.,

$$\langle \nabla f(x) - \nabla f(y), x - y\rangle \geq c\|x - y\|^2, \tag{1.3.30}$$

where $c > 0$ is the constant of (1.3.8).

Proof. Let f be uniformly convex on S, then, by Theorem 1.3.13, for any $x, y \in S$, we have

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \frac{1}{2}c\|y - x\|^2, \tag{1.3.31}$$

$$f(x) \geq f(y) + \langle \nabla f(y), x - y\rangle + \frac{1}{2}c\|x - y\|^2, \tag{1.3.32}$$

and addition of these two inequalities shows that (1.3.30) holds.

Similarly, if f is convex, (1.3.31) and (1.3.32) hold with $c = 0$, and hence (1.3.28) holds. Moreover, if f is strictly convex, then (1.3.31) and (1.3.32) hold with $c = 0$ but with strict inequality for $x \neq y$. Hence the addition establishes (1.3.29).

Conversely, suppose that ∇f is monotone. For any fixed $x, y \in S$, the mean-value theorem (1.2.97) gives

$$f(y) - f(x) = \langle \nabla f(\xi), y - x\rangle, \tag{1.3.33}$$

where $\xi = x + t(y - x), t \in (0, 1)$. Then, it follows from (1.3.28) that

$$\langle \nabla f(\xi) - \nabla f(x), y - x \rangle = \frac{1}{t}[\nabla f(\xi) - \nabla f(x)]^T(\xi - x) \geq 0, \tag{1.3.34}$$

which, together with (1.3.33), gives

$$\begin{aligned} f(y) - f(x) &= \langle \nabla f(\xi) - \nabla f(x), y - x \rangle + \langle \nabla f(x), y - x \rangle \\ &\geq \langle \nabla f(x), y - x \rangle. \end{aligned} \tag{1.3.35}$$

The above inequality shows, by Theorem 1.3.13, that f is convex.

Similarly, if (1.3.29) holds, the same will be true in (1.3.35) with strict inequality and $x \neq y$, and thus f is strictly convex.

Finally, for uniform convexity, suppose (1.3.30) holds. Let $\phi(t) = f(x + t(y - x)) = f(u)$, where $u = x + t(y - x), t \in (0, 1)$. Noting that $\phi'(t) = \langle \nabla f(u), y - x \rangle$ and $\phi'(0) = \langle \nabla f(x), y - x \rangle$, then (1.3.30) means

$$\begin{aligned} \phi'(t) - \phi'(0) &= \langle \nabla f(u) - \nabla f(x), y - x \rangle = \frac{1}{t}\langle \nabla f(u) - \nabla f(x), u - x \rangle \\ &\geq \frac{1}{t}c\|u - x\|^2 = tc\|y - x\|^2. \end{aligned}$$

Hence,

$$\phi(1) - \phi(0) - \phi'(0) = \int_0^1 [\phi'(t) - \phi'(0)]\mathrm{d}t \geq \frac{1}{2}c\|y - x\|^2,$$

which, by the definition of ϕ, shows

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2}c\|y - x\|^2.$$

Therefore, we complete the proof. □

Combining Theorem 1.3.14 and 1.3.16, we immediately obtain the following theorem.

Theorem 1.3.17 *Let $S \subset R^n$ be a nonempty open convex set and f be a twice continuously differentiable function on S. Then*

1. *∇f is monotone on S if and only if $\nabla^2 f(x)$ is positive semidefinite for all $x \in S$.*

2. *If $\nabla^2 f(x)$ is positive definite for all $x \in S$, then ∇f is strictly monotone on S.*

3. *∇f is uniformly (or strongly) monotone on S if and only if $\nabla^2 f(x)$ is uniformly positive definite, i.e., there exists a number $c > 0$ so that*

$$d^T \nabla^2 f(x) d \geq c\|d\|^2, \ \forall x \in S, d \in R^n.$$

In the following, we are concerned with the level set which is closely related to a convex function and important to the minimization algorithm. The following theorem shows that the level set L_α corresponding to a convex function is convex.

Theorem 1.3.18 *Let $S \subset R^n$ be a nonempty convex set, f a convex function defined on S, α a real number. Then the level set $L_\alpha = \{x \mid x \in S, f(x) \leq \alpha\}$ is a convex set.*

Proof. Let $x_1, x_2 \in L_\alpha$, then $x_1, x_2 \in S, f(x_1) \leq \alpha, f(x_2) \leq \alpha$. Let $\lambda \in (0,1)$ and $x = \lambda x_1 + (1-\lambda)x_2$. Then from the convexity of S, we have $x \in S$. Also since f is convex,

$$f(x) \leq \lambda f(x_1) + (1-\lambda) f(x_2) \leq \lambda\alpha + (1-\lambda)\alpha = \alpha.$$

Hence $x \in L_\alpha$, which implies that L_α is a convex set. □

From Theorem 1.3.18 and Theorem 1.3.9, we know immediately that if f is a continuously convex function, then the level set L_α is a closed convex set. Furthermore, we also have

Theorem 1.3.19 *Let $f(x)$ be twice continuously differentiable on $S \subset R^n$, where S is a nonempty convex set. Suppose that there exists a number $m > 0$ such that*

$$u^T \nabla^2 f(x) u \geq m\|u\|^2, \ \forall x \in L(x_0), u \in R^n. \tag{1.3.36}$$

Then the level set $L(x_0) = \{x \in S \mid f(x) \leq f(x_0)\}$ is a bounded closed convex set.

Proof. By using Theorem 1.3.14, (1.3.36) implies that f is convex on $L(x_0)$, and then it follows from Theorem 1.3.18 that $L(x_0)$ is convex. Note that $f(x)$ is continuous, then $L(x_0)$ is a closed convex set for all $x_0 \in R^n$.

Now we prove the boundedness of $L(x_0)$. Using (1.3.36) and the fact that $L(x_0)$ is convex, we have for any $x, y \in L(x_0)$,

$$m\|y-x\|^2 \leq (y-x)^T\nabla^2 f(x+\alpha(y-x))(y-x).$$

Also by twice differentiability and the above inequality, we have

$$\begin{aligned} f(y) &= f(x)+\nabla f(x)^T(y-x) \\ &\quad +\int_0^1\int_0^t (y-x)^T\nabla^2 f(x+\alpha(y-x))(y-x)\mathrm{d}\alpha\mathrm{d}\ t \\ &\geq f(x)+\nabla f(x)^T(y-x)+\frac{1}{2}m\|y-x\|^2, \end{aligned}$$

where m is independent of x and y. Therefore for arbitrary $y \in L(x_0)$ and $y \neq x_0$,

$$\begin{aligned} f(y)-f(x_0) &\geq \nabla f(x_0)^T(y-x_0)+\frac{1}{2}m\|y-x_0\|^2 \\ &\geq -\|\nabla f(x_0)\|\|y-x_0\|+\frac{1}{2}m\|y-x_0\|^2. \end{aligned}$$

Noting that $f(y) \leq f(x_0)$, the above inequality implies

$$\|y-x_0\| \leq \frac{2}{m}\|\nabla f(x_0)\|,$$

which shows that the level set $L(x_0)$ is bounded. □

To conclude this subsection, we give a proof of Minkowski inequality which is an application of convexity of function.

Minkowski inequality:

$$\|x+y\|_p \leq \|x\|_p+\|y\|_p, \tag{1.3.37}$$

i.e.,

$$\left(\sum_{i=1}^n |x_i+y_i|^p\right)^{1/p} \leq \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}+\left(\sum_{i=1}^n |y_i|^p\right)^{1/p}, \tag{1.3.38}$$

where $p \geq 1$.

Proof. If x or y is the zero vector, the result is obvious. Now suppose that $x \neq 0$ and $y \neq 0$.

If $p = 1$, since $|x_i + y_i| \leq |x_i| + |y_i|, i = 1, \cdots, n$, then summing over i gives the result.

Now let $p > 1$ and consider the function

$$\phi(t) = t^p,\ t > 0.$$

Since

$$\phi''(t) = p(p-1)t^{p-2} > 0,$$

then $\phi(t)$ is strictly convex. Note that

$$\frac{\|x\|_p}{\|x\|_p + \|y\|_p} + \frac{\|y\|_p}{\|x\|_p + \|y\|_p} = 1,$$

it follows from the definition of convex function that

$$\begin{aligned} & \left(\frac{\|x\|_p}{\|x\|_p + \|y\|_p} \frac{|x_i|}{\|x\|_p} + \frac{\|y\|_p}{\|x\|_p + \|y\|_p} \frac{|y_i|}{\|y\|_p} \right)^p \\ \leq\ & \frac{\|x\|_p}{\|x\|_p + \|y\|_p} \left(\frac{|x_i|}{\|x\|_p} \right)^p + \frac{\|y\|_p}{\|x\|_p + \|y\|_p} \left(\frac{|y_i|}{\|y\|_p} \right)^p. \end{aligned} \tag{1.3.39}$$

Hence, using (1.3.39), we get

$$\begin{aligned} & \sum_{i=1}^n \left(\frac{|x_i + y_i|}{\|x\|_p + \|y\|_p} \right)^p \\ \leq\ & \sum_{i=1}^n \left(\frac{|x_i| + |y_i|}{\|x\|_p + \|y\|_p} \right)^p \\ =\ & \sum_{i=1}^n \left(\frac{\|x\|_p}{\|x\|_p + \|y\|_p} \frac{|x_i|}{\|x\|_p} + \frac{\|y\|_p}{\|x\|_p + \|y\|_p} \frac{|y_i|}{\|y\|_p} \right)^p \\ \leq\ & \sum_{i=1}^n \left(\frac{\|x\|_p}{\|x\|_p + \|y\|_p} \left(\frac{|x_i|}{\|x\|_p} \right)^p + \frac{\|y\|_p}{\|x\|_p + \|y\|_p} \left(\frac{|y_i|}{\|y\|_p} \right)^p \right) \\ \leq\ & \frac{\|x\|_p}{\|x\|_p + \|y\|_p} \sum_{i=1}^n \left(\frac{|x_i|}{\|x\|_p} \right)^p + \frac{\|y\|_p}{\|x\|_p + \|y\|_p} \sum_{i=1}^n \left(\frac{|y_i|}{\|y\|_p} \right)^p \\ =\ & \frac{\|x\|_p}{\|x\|_p + \|y\|_p} \frac{\|x\|_p^p}{\|x\|_p^p} + \frac{\|y\|_p}{\|x\|_p + \|y\|_p} \frac{\|y\|_p^p}{\|y\|_p^p} \\ =\ & 1, \end{aligned}$$

which implies that

$$\sum_{i=1}^{n} |x_i + y_i|^p \leq (\|x\|_p + \|y\|_p)^p.$$

Taking the p-th root gives our result. □

1.3.3 Separation and Support of Convex Sets

The separation and support of convex sets are important tools for research of optimality conditions. We first discuss the projection theorem which characterizes the projection and describes the sufficient and necessary condition for the distance between a closed convex set and a point not in the set to be minimal.

Theorem 1.3.20 *(Projection Theorem)*
Let $S \subset R^n$ be a nonempty closed convex set and $y \notin S$, then there exists a unique point $\bar{x} \in S$ with minimal distance from y, i.e.,

$$\|y - \bar{x}\| = \inf_{x \in S} \|y - x\|. \tag{1.3.40}$$

Furthermore, $\bar{x}$ is the minimal point of (1.3.40) if and only if

$$\langle y - \bar{x}, x - \bar{x} \rangle \leq 0, \ \forall x \in S, \tag{1.3.41}$$

or say that $\bar{x}$ is the projection $P_S(y)$ of y on S if and only if (1.3.41) holds.

Proof. Let

$$\inf\{\|y - x\| \,|\, x \in S\} = \gamma > 0. \tag{1.3.42}$$

There is a sequence $\{x_k\} \subset S$ so that $\|y - x_k\| \to \gamma$. In the following, we prove $\{x_k\}$ is a Cauchy sequence and hence there exists a limit $\bar{x} \in S$.

By the parallelogram law, we have

$$\begin{aligned} \|x_k - x_m\|^2 &= 2\|x_k - y\|^2 + 2\|x_m - y\|^2 - \|x_k + x_m - 2y\|^2 \\ &= 2\|x_k - y\|^2 + 2\|x_m - y\|^2 - 4\left\|\frac{x_k + x_m}{2} - y\right\|^2 \end{aligned} \tag{1.3.43}$$

Note that $(x_k + x_m)/2 \in S$, we have, from the definition of γ,

$$\left\|\frac{x_k + x_m}{2} - y\right\|^2 \geq \gamma^2.$$

Therefore,

$$\|x_k - x_m\|^2 \leq 2\|x_k - y\|^2 + 2\|x_m - y\|^2 - 4\gamma^2.$$

Taking k and m sufficiently large yields

$$\|x_k - x_m\| \to 0$$

which indicates that $\{x_k\}$ is a Cauchy sequence with limit $\bar{x}$. Since S is closed, then $\bar{x} \in S$. This shows there exists $\bar{x}$ such that $\|y - \bar{x}\| = \gamma$.

Next, we prove the uniqueness. Suppose that $\bar{x}, \bar{x}' \in S$ and satisfy

$$\|y - \bar{x}\| = \|y - \bar{x}'\| = \gamma. \tag{1.3.44}$$

Since S is convex, $(\bar{x} + \bar{x}')/2 \in S$. Then

$$\left\|y - \frac{\bar{x} + \bar{x}'}{2}\right\| \leq \frac{1}{2}\|y - \bar{x}\| + \frac{1}{2}\|y - \bar{x}'\| = \gamma. \tag{1.3.45}$$

If the strict inequality holds, we get a contradiction to (1.3.42). Then the equality holds in (1.3.45) and we have

$$y - \bar{x} = \lambda(y - \bar{x}'), \text{ for some } \lambda.$$

So, it follows from (1.3.44) that $|\lambda| = 1$. If $\lambda = -1$, we have $y = (\bar{x}+\bar{x}')/2 \in S$ which contradicts $y \notin S$. Therefore, $\lambda = 1$, that means $\bar{x} = \bar{x}'$.

Finally, we prove that the distance between $\bar{x} \in S$ and $y \notin S$ is minimal if and only if (1.3.41) holds.

Take x arbitrary in S and suppose (1.3.41) holds. Since

$$\begin{aligned} \|y - x\|^2 &= \|y - \bar{x} + \bar{x} - x\|^2 \\ &= \|y - \bar{x}\|^2 + \|\bar{x} - x\|^2 + 2(\bar{x} - x)^T(y - \bar{x}), \end{aligned}$$

then $\|y - x\|^2 \geq \|y - \bar{x}\|^2$ which is the desired sufficiency.

Conversely, let $\|y - x\|^2 \geq \|y - \bar{x}\|^2$, $\forall x \in S$. Since $\bar{x} + \lambda(x - \bar{x}) \in S$ with $\lambda \in (0, 1)$, then we have

$$\|y - \bar{x} - \lambda(x - \bar{x})\|^2 \geq \|y - \bar{x}\|^2.$$

Developing the square gives

$$\|y - \bar{x} - \lambda(x - \bar{x})\|^2 = \|y - \bar{x}\|^2 + \lambda^2\|x - \bar{x}\|^2 + 2\lambda(x - \bar{x})^T(\bar{x} - y).$$

Then we get

$$\lambda^2\|x - \bar{x}\|^2 + 2\lambda(x - \bar{x})^T(\bar{x} - y) \geq 0.$$

Dividing by λ and letting $\lambda \downarrow 0$, we obtain the result. □

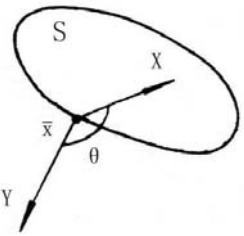


Figure 1.3.5 The angle-characterization of a projection

In fact, if we note that $\bar{x} = P_S(y)$ is the solution of

$$\min_{x \in S} \phi(x) \triangleq \frac{1}{2}\|x - y\|^2,$$

then, for the above minimization problem, one concludes from the optimality condition that

$$\langle \phi'(\bar{x}), x - \bar{x} \rangle \geq 0, \ \forall x \in S.$$

Since $\phi'(x) = x - y$, we have

$$\langle \bar{x} - y, x - \bar{x} \rangle \geq 0, \ \forall x \in S$$

which is just (1.3.41).

Remark: If S is an affine manifold (for example, a subspace), then $\bar{x} - x \in S$ whenever $x - \bar{x} \in S$. Therefore, (1.3.41) implies

$$\langle y - \bar{x}, x - \bar{x} \rangle = 0, \ \forall x \in S, \tag{1.3.46}$$

which is $(y - \bar{x}) \perp S$.

Now we can present the most fundamental separation theorem which separates a closed convex set and a point not in the set. This theorem is based on the above projection theorem.

Theorem 1.3.21 *Let $S \subset R^n$ be a nonempty closed convex set and $y \notin S$. Then there exist a nonzero vector p and a real number α such that*

$$p^T y > \alpha \ \textit{and} \ p^T x \leq \alpha, \ \forall x \in S, \tag{1.3.47}$$

i.e.,

$$p^T y > \sup\{p^T x, \ \forall x \in S\} \tag{1.3.48}$$

which says there exists a hyperplane $H = \{x \,|\, p^T x = \alpha\}$ that strictly separates y and S.

Proof. Since S is a nonempty closed convex set and $y \notin S$, then, by Theorem 1.3.20, there exists a unique point $\bar{x} \in S$ such that

$$(x - \bar{x})^T (y - \bar{x}) \leq 0, \ \forall x \in S.$$

Set $p = y - \bar{x} \neq 0$, then

$$\begin{aligned} 0 &\geq (y - \bar{x})^T (y - \bar{x} + x - y) \\ &= p^T x - p^T y + \|p\|^2. \end{aligned}$$

Hence

$$p^T y \geq p^T x + \|p\|^2, \ \forall x \in S.$$

Set $\alpha = \sup\{p^T x \,|\, x \in S\}$, and we get our result. □

As a consequence of Theorem 1.3.21, we immediately obtain Farkas' Lemma which has been used extensively in the derivation of optimality conditions.

Theorem 1.3.22 *(Farkas' Lemma) Let $A \in R^{m \times n}$ and $c \in R^n$. Then exactly one of the following two systems has a solution:*

$$\textit{System 1} \qquad Ax \leq 0, \ c^T x > 0, \tag{1.3.49}$$

$$\textit{System 2} \qquad A^T y = c, \ y \geq 0. \tag{1.3.50}$$

Proof. Suppose that there is a solution for (1.3.50), that is, there exists $y \geq 0$ such that $A^T y = c$. Let x satisfy $Ax \leq 0$, it follows from $y \geq 0$ that

$$c^T x = y^T A x \leq 0,$$

which shows that (1.3.49) has no solution.

Now suppose (1.3.50) has no solution. Let

$$S = \{x \,|\, x = A^T y, y \geq 0\},$$

which is a polyhedral set, and hence it is a nonempty closed convex set and $c \notin S$. By Theorem 1.3.21, there exist $p \in R^n$ and $\alpha \in R$ such that

$$p^T c > \alpha \text{ and } p^T x \leq \alpha, \ \forall x \in S.$$

Since $0 \in S, \alpha \geq p^T 0 = 0$. Then $p^T c > 0$. Also note that

$$\alpha \geq p^T x = p^T A^T y = y^T A p, \ \forall y \geq 0$$

and that y could be made arbitrarily large, thus it follows that $Ap \leq 0$. So, there is a vector $p \in R^n$ which is a solution of (1.3.49), and the proof is complete. □

In order to discuss the separation between two convex sets, we need the following definition and theorem of a supporting hyperplane.

Definition 1.3.23 *Let $S \subset R^n$ be a nonempty set, $p \in R^n$, and $\bar{x} \in \partial S$, where ∂S denotes the boundary of S. If either*

$$S \subset H^+ = \{x \in S \,|\, p^T(x - \bar{x}) \geq 0\} \tag{1.3.51}$$

or

$$S \subset H^- = \{x \in S \,|\, p^T(x - \bar{x}) \leq 0\}, \tag{1.3.52}$$

then the hyperplane $H = \{x \in S \,|\, p^T(x - \bar{x}) = 0\}$ is called a supporting hyperplane of S at $\bar{x}$. If, in addition, $S \not\subset H$, then H is called a proper supporting hyperplane of S at $\bar{x}$.

The following theorem shows that a convex set has a supporting hyperplane at each boundary point (see Figure 1.3.6).

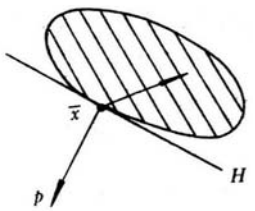


Figure 1.3.6 Supporting hyperplane

Theorem 1.3.24 *Let $S \subset R^n$ be a nonempty convex set and $\bar{x} \in \partial S$. Then, there exists a hyperplane supporting S at $\bar{x}$; that is, there exists a nonzero vector p such that*

$$p^T(x - \bar{x}) \leq 0, \ \forall x \in \bar{S}, \tag{1.3.53}$$

where $\bar{S}$ denotes the closure of S.

Proof. Since $\bar{x} \in \partial S$, there exists a sequence $\{y_k\} \not\subset \bar{S}$ so that $y_k \to \bar{x}, (k \to \infty)$. By Theorem 1.3.21, corresponding to each y_k, there exists $p_k \in R^n$ with $\|p_k\| = 1$, such that

$$p_k^T y_k > p_k^T x, \ \forall x \in \bar{S}. \tag{1.3.54}$$

Since $\{p_k\}$ is bounded, there is a convergent subsequence $\{p_k\}_{\mathcal{K}}$ with limit p and $\|p\| = 1$. For this subsequence, (1.3.54) holds, that is

$$p_{k_j}^T y_{k_j} > p_{k_j}^T x, \ \forall x \in \bar{S}.$$

Fix $x \in \bar{S}$ and take the limit as $k \in \mathcal{K}$ and $k \to \infty$, we have $p^T \bar{x} \geq p^T x$, $\forall x \in \bar{S}$, which is our desired result. □

By use of Theorem 1.3.21 and Theorem 1.3.24, the following corollary is obvious.

Corollary 1.3.25 *Let $S \subset R^n$ be a nonempty convex set and $\bar{x} \notin S$. Then there exists a nonzero vector p such that*

$$p^T(x - \bar{x}) \leq 0, \ \forall x \in \bar{S}. \tag{1.3.55}$$

Proof. Let $\bar{x} \notin S$; there are two cases. If $\bar{x} \notin \bar{S}$, the conclusion is immediate from Theorem 1.3.21. If $\bar{x} \in \partial S$, the corollary reduces to Theorem 1.3.24. □

Now, we are going to discuss the separation theorems of two convex sets which include separation theorem, strict separation theorem and strong separation theorem.

Definition 1.3.26 *Let $S_1, S_2 \subset R^n$ be nonempty convex sets. If*

$$p^T x \geq \alpha, \forall x \in S_1 \text{ and } p^T x \leq \alpha, \forall x \in S_2, \tag{1.3.56}$$

then the hyperplane $H = \{x \,|\, p^T x = \alpha\}$ is said to separate S_1 and S_2. If, in addition, $S_1 \cup S_2 \not\subset H$, then H is said to properly separate S_1 and S_2. If

$$p^T x > \alpha, \forall x \in S_1 \text{ and } p^T x < \alpha, \forall x \in S_2, \tag{1.3.57}$$

then H is said to strictly separate S_1 and S_2. If

$$p^T x \geq \alpha + \epsilon, \ \forall x \in S_1 \text{ and } p^T x \leq \alpha, \forall x \in S_2, \tag{1.3.58}$$

then H is said to strongly separate S_1 and S_2, where $\epsilon > 0$.

Theorem 1.3.27 *(Separation Theorem)*
Let $S_1, S_2 \subset R^n$ be nonempty convex sets. If $S_1 \cap S_2 = \phi$, then there exists a hyperplane separating S_1 and S_2, that is, there exists a nonzero vector $p \in R^n$ such that

$$p^T x_1 \leq p^T x_2, \ \forall x_1 \in \bar{S}_1, x_2 \in \bar{S}_2. \tag{1.3.59}$$

Proof. Let

$$S = S_1 - S_2 = \{x_1 - x_2 \,|\, x_1 \in S_1, x_2 \in S_2\}.$$

Note that S is a nonempty convex set and that $0 \notin S$ (otherwise, if $0 \in S$, then we have $x_1 - x_2 = 0$ and $x_1 = x_2 \in S_1 \cap S_2$ which implies $S_1 \cap S_2 \neq \phi$). Hence, by Corollary 1.3.25, there exists a nonzero vector p such that

$$p^T x \leq p^T 0 = 0, \ \forall x \in \bar{S},$$

which implies that

$$p^T x_1 \leq p^T x_2, \forall x_1 \in \bar{S}_1, x_2 \in \bar{S}_2.$$

Then we complete the proof. □

Note that (1.3.59) also can be written as

$$\sup\{p^T x \,|\, x \in S_1\} \leq \inf\{p^T x \,|\, x \in S_2\}. \tag{1.3.60}$$

Theorem 1.3.28 *(Strong Separation Theorem)*
Let S_1 and S_2 be two closed convex sets on R^n, and suppose that S_2 is bounded. If $S_1 \cap S_2 = \phi$, then there exists a hyperplane that strongly separates S_1 and S_2, that is, there exist a nonzero vector p and $\epsilon > 0$ such that

$$\inf\{p^T x \,|\, x \in S_2\} \geq \sup\{p^T x \,|\, x \in S_1\} + \epsilon. \tag{1.3.61}$$

Proof. Let $S = S_1 - S_2$. Note that S is convex and $0 \notin S$. We first prove that S is closed. Let $\{x_k\} \subset S, x_k \to x$. By the definition of S, $x_k = y_k - z_k, y_k \in S_1, z_k \in S_2$. Since S_2 is compact, there exists a convergent subsequence $\{z_k\}_{\mathcal{K}}, z_k \to z, z \in S_2, k \in \mathcal{K}$. Since

$$y_k - z_k \to x, z_k \to z, \ \forall k \in \mathcal{K},$$

then $y_k \to y$. Also since S_1 is closed, $y \in S_1$. Therefore,

$$x = y - z,\ y \in S_1, z \in S_2,$$

which means $x \in S$ and S is closed.

Now we have that S is a closed convex set and $0 \notin S$. Then, by Theorem 1.3.21, there exist nonzero vector p and real number α, such that

$$p^T x \leq \alpha,\ \forall x \in S \text{ and } p^T 0 > \alpha.$$

Hence, $\alpha < 0$. Using the definition of S yields

$$p^T z \geq p^T y - \alpha,\ \forall y \in S_1, z \in S_2,$$

which, by setting $\epsilon = -\alpha > 0$, is

$$\inf\{p^T z \,|\, z \in S_2\} \geq \sup\{p^T y \,|\, y \in S_1\} + \epsilon. \quad \square$$

Similarly, we can obtain the following strict separation theorem.

Theorem 1.3.29 *(Strict Separation Theorem)*
Let S_1 and S_2 be two closed convex sets on R^n, and suppose that S_2 is bounded. If $S_1 \cap S_2 = \phi$, there exists a nonzero vector p such that

$$\inf\{p^T x \,|\, x \in S_2\} > \sup\{p^T x \,|\, x \in S_1\}. \tag{1.3.62}$$

Proof. The result (1.3.62) is immediate from (1.3.61). $\square$

1.4 Optimality Conditions for Unconstrained Optimization

In this section we consider the unconstrained optimization problem

$$\min f(x),\ x \in R^n \tag{1.4.1}$$

and present its optimality conditions which include first-order conditions and second-order conditions.

In general, we have two types of minimizers: local minimizer and global minimizer. In the following, we give their exact definitions.

Definition 1.4.1 *A point x^* is called a local minimizer if there exists $\delta > 0$ such that $f(x^*) \leq f(x)$ for all $x \in R^n$ satisfying $\|x - x^*\| < \delta$.*

A point x^* is called a strict local minimizer if there exists $\delta > 0$ such that $f(x^*) < f(x)$ for all $x \in R^n$ with $x \neq x^*$ and $\|x - x^*\| < \delta$.

Definition 1.4.2 *A point x^* is called a global minimizer if $f(x^*) \leq f(x)$ for all $x \in R^n$. A point x^* is called a strict global minimizer if $f(x^*) < f(x)$ for all $x \in R^n$ with $x \neq x^*$.*

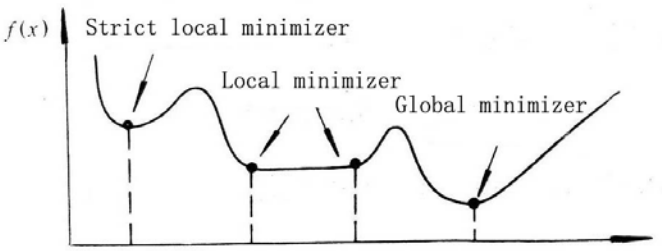


Figure 1.4.1 Types of minimal points

Note that, in practice, most algorithms are able to find only a local minimizer that is not a global minimizer. Normally, finding a global minimizer is a difficult task. In many practical applications, we are content with getting a local minimizer. In addition, many global optimization algorithms proceed by solving a sequence of local optimization algorithms. Hence, in this book, our focus is on the model, property, convergence and computation of local optimization algorithms. Usually, in the book, the minimizer refers to the local minimizer. However, if the problem is a convex programming problem, all local minimizers are also global minimizers.

The descent direction given in the following definition is an important concept.

Definition 1.4.3 *Let $f : R^n \to R$ be differentiable at $x \in R^n$. If there exists a vector $d \in R^n$ such that*

$$\langle \nabla f(x), d \rangle < 0, \tag{1.4.2}$$

then d is called a descent direction of f at x.

By means of Taylor's expansion,

$$f(x_k + td) = f(x_k) + t\nabla f(x_k)^T d + o(t),$$

then it is easy to see that

$$\exists \delta > 0 \text{ such that } f(x_k + td) < f(x_k),\ \forall t \in (0, \delta)$$

if and only if d is a descent direction of f at x_k.

Now we discuss the first-order optimality condition.

Theorem 1.4.4 *(First-Order Necessary Condition)*
Let $f : D \subset R^n \to R$ be continuously differentiable on an open set D. If $x^* \in D$ is a local minimizer of (1.4.1), then

$$\nabla f(x^*) = 0. \tag{1.4.3}$$

Proof. [proof I] Let x^* be a local minimizer. Consider the sequence

$$x_k = x^* - \alpha_k \nabla f(x^*),\ \alpha_k > 0.$$

By Taylor's expansion, for k sufficiently large, we have

$$0 \le f(x_k) - f(x^*) = -\alpha_k \nabla f(\eta_k)^T \nabla f(x^*),$$

where η_k is a convex combination of x_k and x^*. Dividing by α_k and taking the limit, it follows from $f \in C^1$ that

$$0 \le -\|\nabla f(x^*)\|^2$$

which means $\nabla f(x^*) = 0$. □

[proof II] (By contradiction). Suppose that $\nabla f(x^*) \neq 0$. Taking $d = -\nabla f(x^*)$ yields

$$d^T \nabla f(x^*) = -\|\nabla f(x^*)\|^2 < 0.$$

So, d is a descent direction and there exists $\delta > 0$ such that

$$f(x^* + \alpha d) < f(x^*),\ \forall \alpha \in (0, \delta)$$

which contradicts the assumption that x^* is a local minimizer. □

[proof III] Let x^* be a local minimizer, then there exists $\delta > 0$ so that $f(x) \geq f(x^*)$ for any x with $\|x - x^*\| < \delta$. By Taylor's expansion,

$$f(x) = f(x^*) + \nabla f(x^*)^T(x - x^*) + o(\|x - x^*\|) \geq f(x^*).$$

Dividing $\|x - x^*\|$ and letting $x \to x^*$ yield

$$\nabla f(x^*)^T \frac{(x - x^*)}{\|x - x^*\|} \geq 0.$$

Setting $s = (x - x^*)/\|x - x^*\|$, the above inequality is

$$\nabla f(x^*)^T s \geq 0, \ \forall s \text{ with } \|s\| = 1.$$

Choosing $s = \pm e_i, (i = 1, \cdots, n)$, we obtain $\nabla f(x^*) = 0$. □

Theorem 1.4.4 says that if x^* is a local minimizer, f has a zero slope at x^*. The following theorem indicates that if x^* is a local minimizer, f has nonnegative curvature at x^* besides zero slope.

Theorem 1.4.5 *(Second-Order Necessary Condition)*
Let $f : D \subset R^n \to R$ be twice continuously differentiable on an open set D. If x^* is a local minimizer of (1.4.1), then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive semidefinite.

Proof. [proof I] We have known from Theorem 1.4.4 that $\nabla f(x^*) = 0$, hence we only need to prove that $\nabla^2 f(x^*)$ is positive semidefinite. Consider the sequence

$$x_k = x^* + \alpha_k d, \ \alpha_k > 0,$$

where d is arbitrary. Since $f \in C^2$ and $\nabla f(x^*) = 0$, then by Taylor's expansion, we have for k sufficiently large that

$$0 \leq f(x_k) - f(x^*) = \frac{1}{2}\alpha_k^2 d^T \nabla^2 f(\eta_k) d,$$

where η_k is a convex combination of x_k and x^*. Dividing by $\frac{1}{2}\alpha_k^2$ and taking the limit, we get

$$d^T \nabla^2 f(x^*) d \geq 0, \ \forall d \in R^n.$$

Hence we complete the proof. □

[proof II] (By contradiction). Suppose that $\nabla^2 f(x^*)$ is not positive semidefinite, then we can choose $d \in R^n$ such that $d^T \nabla^2 f(x^*) d < 0$. Since $f \in C^2$,

there exists $\delta > 0$ and we can choose $\epsilon > 0$ such that $x^* + \epsilon d \in B(x^*, \delta)$ and $d^T \nabla^2 f(x^* + \epsilon d) d < 0$.

By use of $\nabla f(x^*) = 0$, it follows that

$$f(x^* + \epsilon d) = f(x^*) + \frac{1}{2}\epsilon^2 d^T \nabla^2 f(x^* + \theta\epsilon d) d,$$

where $0 \leq \theta \leq 1$. Therefore $f(x^* + \epsilon d) < f(x^*)$. This contradicts the assumption that x^* is a local minimizer. □

Next, we describe the second-order sufficient condition.

Theorem 1.4.6 *(Second-Order Sufficient Condition)*

Let $f : D \subset R^n \to R$ be twice continuously differentiable on an open set D. If $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite, then $x^* \in D$ is a strict local minimizer.

Proof. [proof I] Assume that $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite. By Taylor's expansion, for any vector $d \in R^n$ such that $x^* + d$ lies in a neighborhood of x^* in which $\nabla^2 f(x^* + d)$ is positive definite, we have

$$f(x^* + d) = f(x^*) + \frac{1}{2} d^T \nabla^2 f(x^* + \theta d) d,$$

where $\theta \in (0, 1)$. Then we can choose $\delta > 0$ such that $x^* + d \in B(x^*, \delta)$ and $d^T \nabla^2 f(x^* + \theta d) d > 0$. Therefore,

$$f(x^* + d) > f(x^*)$$

which shows our result. □

[proof II] (By contradiction). Assume that x^* is not a strict local minimizer, then there exists a sequence $\{x_k\} \subset D$ with $x_k \neq x^*$, $\forall k$, such that $f(x_k) \leq f(x^*)$ for k sufficiently large. By use of Taylor's expansion,

$$\begin{aligned} 0 &\geq f(x_k) - f(x^*) \\ &= \nabla f(x^*)^T (x_k - x^*) + \frac{1}{2}(x_k - x^*)^T \nabla^2 f(\eta_k)(x_k - x^*), \end{aligned}$$

where η_k is a convex combination of x_k and x^*. Using $\nabla f(x^*) = 0$, dividing by $\frac{1}{2}\|x_k - x^*\|^2$ and taking the limit, we have

$$0 \geq \bar{e}^T \nabla^2 f(x^*) \bar{e}, \tag{1.4.4}$$

where $\bar{e}$ is the accumulation point of the uniformly bounded sequence $\{(x_k - x^*)/\|x_k - x^*\|\}$ and $\|\bar{e}\| = 1$. Obviously, (1.4.4) contradicts the positive definiteness of $\nabla^2 f(x^*)$. $\square$

Definition 1.4.7 *A point $x^* \in R^n$ is said to be a stationary (or critical) point for the differentiable f if $\nabla f(x^*) = 0$.*

Theorem 1.4.4 tells us that if x^* is a local minimizer, then it is a stationary point. However, the converse is not true. If x^* is a stationary point, it is possible for x^* to be a local minimizer or maximizer, it is also possible for x^* to not be an extreme point. If a stationary point x^* is neither minimizer nor maximizer, it is called a saddle point. Therefore, a stationary point need not be a local minimizer. But if the objective function which is differentiable is convex, its stationary points are the local minimizers and the global minimizers.

Theorem 1.4.8 below says, for a convex function, that its local minimizer is also a global minimizer. Theorem 1.4.9 says, for a differentiable convex function, that its stationary point is also a global minimizer.

Theorem 1.4.8 *Let $S \subset R^n$ be a nonempty convex set and $f : S \subset R^n \to R$. Let $x^* \in S$ be a local minimizer for $\min_{x \in S} f(x)$.*

1. *If f is convex, then x^* is also a global minimizer.*

2. *If f is strictly convex, then x^* is a unique global minimizer.*

Proof. (1) Let f be convex and x^* be a local minimizer, then there exists a δ-neighborhood $B(x^*, \delta)$ such that

$$f(x) \geq f(x^*), \ \forall x \in S \cap B(x^*, \delta). \tag{1.4.5}$$

By contradiction, suppose that x^* is not a global minimizer. Then we can find some $\hat{x} \in S$ such that $f(\hat{x}) < f(x^*)$. By convexity of f, we have for $\alpha \in (0, 1)$,

$$\begin{aligned} f(\alpha\hat{x} + (1-\alpha)x^*) &\leq \alpha f(\hat{x}) + (1-\alpha) f(x^*) \\ &< \alpha f(x^*) + (1-\alpha) f(x^*) \\ &= f(x^*). \end{aligned} \tag{1.4.6}$$

But for sufficiently small $\alpha > 0$, $\alpha\hat{x} + (1-\alpha)x^* \in S \cap B(x^*, \delta)$. Therefore, (1.4.6) contradicts (1.4.5). This contradiction proves the first conclusion.

(2) From part (1) we have that x^* is a global minimizer because strict convexity means convexity. Therefore, it is enough to prove the uniqueness.

By contradiction, suppose that x^* is not the unique global minimizer, so that we can find $x \in S, x \neq x^*$, such that $f(x) = f(x^*)$.

By strict convexity of f,

$$f(\frac{1}{2}x + \frac{1}{2}x^*) < \frac{1}{2}f(x) + \frac{1}{2}f(x^*) = f(x^*). \tag{1.4.7}$$

Note from the convexity of S that $\frac{1}{2}x + \frac{1}{2}x^* \in S$. Therefore, (1.4.7) contradicts the fact that x^* is a global minimizer. □

Theorem 1.4.9 *Let $f : R^n \to R$ be a differentiable convex function. Then x^* is a global minimizer if and only if $\nabla f(x^*) = 0$.*

Proof. Sufficiency. Let f be a differentiable convex function in R^n and $\nabla f(x^*) = 0$, then

$$f(x) \geq f(x^*) + \nabla f(x^*)(x - x^*) = f(x^*), \ \forall x \in R^n$$

which indicates that x^* is a global minimizer of f.

Necessity. It is obvious because the global minimizer is also a local minimizer, and is also a stationary point. □

The optimality conditions of constrained optimization will be discussed in Chapter 8.

1.5 Structure of Optimization Methods

Usually, the optimization method is an iterative one for finding the minimizer of an optimization problem. The basic idea is that, given an initial point $x_0 \in R^n$, one generates an iterate sequence $\{x_k\}$ by means of some iterative rule, such that when $\{x_k\}$ is a finite sequence, the last point is the optimal solution of the model problem; when $\{x_k\}$ is infinite, it has a limit point which is the optimal solution of the model problem. A typical behavior of an algorithm which is regarded as acceptable is that the iterates x_k move steadily towards the neighborhood of a local minimizer x^*, and then rapidly converge to the point x^*. When a given convergence rule is satisfied, the

iteration will be terminated. In general, the most natural stopping criterion is

$$\|\nabla f(x_k)\| \leq \delta, \tag{1.5.1}$$

where δ is a prescribed tolerance. If (1.5.1) is satisfied, it implies that the gradient vector $\nabla f(x_k)$ tends to zero and the iterate sequence $\{x_k\}$ converges to a stationary point.

Let x_k be the k-th iterate, d_k k-th direction, α_k k-th steplength factor. Then the k-th iteration is

$$x_{k+1} = x_k + \alpha_k d_k. \tag{1.5.2}$$

We can see from (1.5.2) that the different stepsize α_k and different direction d_k form different methods. In Chapter 2 we will discuss several methods to determine α_k. In Chapter 3 we will present various methods to find search directions d_k. Most optimization methods are so-called descent methods in the sense that f satisfies at each iteration

$$f(x_{k+1}) = f(x_k + \alpha_k d_k) < f(x_k),$$

in which d_k is a descent direction defined by Definition 1.4.3.

The basic scheme of optimization methods is as follows.

Algorithm 1.5.1 *(Basic Scheme)*

Step 0. (Initial step) Given initial point $x_0 \in R^n$ and the tolerance $\epsilon > 0$.

Step 1. (Termination criterion) If $\|\nabla f(x_k)\| \leq \epsilon$, stop.

Step 2. (Finding the direction) According to some iterative scheme, find d_k which is a descent direction.

Step 3. (Line search) Determine the stepsize α_k such that the objective function value decreases, i.e.,

$$f(x_k + \alpha_k d_k) < f(x_k).$$

Step 4. (Loop) Set $x_{k+1} = x_k + \alpha_k d_k$, $k := k + 1$, and loop to Step 1. □

Convergence rate, which is a local characterization of an algorithm, can measure the effectiveness of an optimization method. We now give a brief description associated with different types of convergence rate. More details can be found in Ortega and Rheinboldt (1970).

Let the iterate sequence $\{x_k\}$ generated by an algorithm converge to x^* in some norm, i.e.,

$$\lim_{k\to\infty} \|x_k - x^*\| = 0. \tag{1.5.3}$$

If there are real number $\alpha \geq 1$ and a positive constant β which is independent of the iterative number k, such that

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^\alpha} = \beta, \tag{1.5.4}$$

we say that $\{x_k\}$ has α-order of Q-convergence rate, where Q-convergence rate means Quotient-convergence rate. In particular,

1. when $\alpha = 1$ and $\beta \in (0, 1)$, the sequence $\{x_k\}$ is said to converge Q-linearly;

2. when $\alpha = 1$ and $\beta = 0$, or $1 < \alpha < 2$ and $\beta > 0$, the sequence $\{x_k\}$ is said to converge Q-superlinearly;

3. when $\alpha = 2$, we say that $\{x_k\}$ has Q-quadratic convergence rate.

The primary motivation for introducing the Q-convergence rate is to compare the speed of convergence of different iterations. It is not difficult to see that the convergence rate depends on α and (more weakly) on β. Suppose that there are two sequences $\{x_k\}$ and $\{x'_k\}$ and that their Q-order and Q-factor are respectively $\{\alpha, \beta\}$ and $\{\alpha', \beta'\}$. If $\alpha > \alpha'$, then the sequence with Q-α order converges faster than the sequence with Q-α' order. For example, a quadratically convergent sequence will eventually converge faster than linearly and superlinearly convergent sequences. When $\alpha = \alpha'$, i.e., their Q-order of convergence rate is the same, if $\beta < \beta'$, then the sequence $\{x_k\}$ is faster than $\{x'_k\}$.

Mainly, we are concerned with Q-linear, Q-superlinear and Q-quadratic convergence. Usually, if the convergence rate of an algorithm is Q-superlinear or Q-quadratic, we say that it has rapid convergence rate. For example, quasi-Newton methods converge Q-superlinearly, and Newton's method converges Q-quadratically.

Another measure of the convergence rate which is weaker than Q-convergence rate is R-convergence rate which means Root-convergence rate.

Let $\{x_k\} \subset R^n$ be any sequence that converges to x^*. Let

$$R_p = \begin{cases} \limsup_{k\to\infty} \|x_k - x^*\|^{1/k}, & \text{if } p = 1; \\ \limsup_{k\to\infty} \|x_k - x^*\|^{1/p^k}, & \text{if } p > 1. \end{cases}$$

If $R_1 = 0$, $\{x_k\}$ is said to be R-superlinearly convergent to x^*.

If $0 < R_1 < 1$, $\{x_k\}$ is said to be R-linearly convergent to x^*.

If $R_1 = 1$, $\{x_k\}$ is said to be R-sublinearly convergent to x^*.

Similarly, if $R_2 = 0, 0 < R_2 < 1, R_2 \geq 1$ respectively, then $\{x_k\}$ is said to be R-superquadratically, R-quadratically, and R-subquadratically convergent to x^* respectively.

The above R-rate of convergence can also be stated as follows:

The sequence $\{x_k\}$ is said to be R-linearly convergent if there is a sequence of nonnegative scalars $\{q_k\}$ such that

$$\|x_k - x^*\| \leq q_k \quad \text{for all } k, \text{ and } \{q_k\} \text{ converges } Q\text{-linearly to zero.}$$

Similarly, the sequence $\{x_k\}$ is said to be R-superlinearly convergent if $\{q_k\}$ converges Q-superlinearly to zero; the sequence $\{x_k\}$ is said to be R-quadratically convergent if $\{q_k\}$ converges Q-quadratically to zero.

Similar to Q-rate of convergence, R-rate of convergence also depends on R-order p and R-factor R_p. The higher the R-order is, the faster the corresponding sequence converges. When the R-order is the same, the smaller the R-factor is, the faster the corresponding sequence converges.

Throughout this book we mainly discuss Q-convergence rate. Hence, if there is not specific indication, the convergence rate refers to Q-convergence rate.

As indicated above, usually an algorithm with superlinear or quadratic rate is said to be desirable. However, it must be appreciated that the theoretical results of the convergence and convergence rate are not a guarantee of good performance. Not only do these results themselves fall short of guarantee of good behavior, but also they neglect computer round-off errors which may be crucial. In addition, these results often impose certain restrictions on $f(x)$ which may not be easy to verify, and in some cases (for example, in the convex case), these conditions may not be satisfied in practice. Thus,

the development of an optimization method also relies on numerical experimentation. The ideal is a good selection of experimental testing backed up by the proofs of convergence and convergence rate.

We have known from the above discussion that the convergence rate measures the local behavior of an algorithm and is used in local analysis. The theorem below gives a characterization of superlinear convergence which is useful for constructing termination criteria.

Theorem 1.5.2 *If the sequence $\{x_k\}$ converges Q-superlinearly to x^*, then*

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x_k\|}{\|x_k - x^*\|} = 1. \tag{1.5.5}$$

However, in general, the converse is not true.

Proof. For a given integer $k \geq 0$,

$$\begin{aligned} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} &= \frac{\|(x_{k+1} - x_k) + (x_k - x^*)\|}{\|x_k - x^*\|} \\ &\geq \left| \frac{\|x_{k+1} - x_k\|}{\|x_k - x^*\|} - \frac{\|x_k - x^*\|}{\|x_k - x^*\|} \right|. \end{aligned}$$

It follows from the definition of Q-superlinear convergence that

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

To show that the converse is not true, we give a counter-example. In normed space $\{R, |\cdot|\}$, define a sequence $\{x_k\}$ as follows:

$$\begin{aligned} x_{2i-1} &= \frac{1}{i!} \quad (i = 1, 2, \cdots), \\ x_{2i} &= 2x_{2i-1} \quad (i = 1, 2 \cdots). \end{aligned}$$

Obviously, $x^* = 0$. We have

$$\frac{|x_{k+1} - x_k|}{|x_k - x^*|} = \begin{cases} 1, & k = 2i - 1, i \geq 1, \\ 1 - \frac{1}{2(i+1)}, & k = 2i, i \geq 1. \end{cases}$$

So, $\{x_k\}$ satisfies (1.5.5) but does not converge Q-superlinearly to x^*. □

This theorem shows that if an algorithm is convergent Q-superlinearly, instead of $\|x_k - x^*\|$, we can use $\|x_{k+1} - x_k\|$ to give a termination criterion, and the estimation will be improved as k increases.

Finally, we discuss some termination criteria which are used frequently in practice. In order to guarantee convergence of an algorithm, we require

$$|f(x_k) - f(x^*)| \leq \epsilon \text{ or } \|x_k - x^*\| \leq \epsilon,$$

where the parameter ϵ is user-supplied. Unfortunately, these are not practicable since they need the information of the solution x^*.

Instead, we often use the following termination criteria:

$$\|\nabla f(x_k)\| \leq \epsilon_3, \tag{1.5.6}$$
$$\|x_{k+1} - x_k\| \leq \epsilon_1, \tag{1.5.7}$$
$$f(x_k) - f(x_{k+1}) \leq \epsilon_1. \tag{1.5.8}$$

Normally, when an algorithm can be expected to converge rapidly, it is suggested to use (1.5.7) or (1.5.8). When an algorithm has first-order derivative information and can be expected to converge less rapidly, a test based on (1.5.6) may be appropriate.

Himmeblau [174] suggested that it is suitable to use (1.5.7) together with (1.5.8) as follows:

When $\|x_k\| > \epsilon_2$ and $|f(x_k)| > \epsilon_2$, use

$$\frac{\|x_{k+1} - x_k\|}{\|x_k\|} \leq \epsilon_1, \ \frac{|f(x_k) - f(x_{k+1})|}{|f(x_k)|} \leq \epsilon_1; \tag{1.5.9}$$

otherwise, use

$$\|x_{k+1} - x_k\| \leq \epsilon_1, \ |f(x_k) - f(x_{k+1})| \leq \epsilon_1. \tag{1.5.10}$$

He also suggested using (1.5.9)-(1.5.10) together with (1.5.6).

In general, take $\epsilon_1 = \epsilon_2 = 10^{-5}, \epsilon_3 = 10^{-4}$.

Exercises

1. Let A be an $n \times n$ nonsingular matrix. Prove that $\|Ax\| \geq \|x\|/\|A^{-1}\|$.

2. Prove the equivalence (1.2.22)-(1.2.26) of vector norms.

3. Prove Cauchy-Schwarz inequality (1.2.34). Further, prove inequality (1.2.35).

4. Prove (1.2.45).

5. Let $A = UDV^*$ be the singular value decomposition. Prove that $A^+ = VD^+U^*$, where D^+ is defined in (1.2.54).

6. Prove Sherman-Morrison formula (1.2.67) and Sherman-Morrison-Woodburg formula (1.2.68).

7. Show Theorem 1.2.6 (Von-Neumann Lemma).

8. Prove (1.2.69) and (1.2.70).

9. Prove that a function that is Fréchet differentiable must be Gateaux differentiable, but the converse is not true.

10. Prove Theorem 1.2.23.

11. Show that the intersection of finitely many convex sets is a convex set.

12. Show by induction that the set $S \subset R^n$ is convex if and only if for any $x_1, x_2, \cdots, x_m \in S$,

$$\sum_{i=1}^{m} \alpha_i x_i \in S,$$

where $\sum_{i=1}^{m} \alpha_i = 1, \alpha_i \geq 0, i = 1, \cdots, m$. That means a convex combination of arbitrarily finitely many points of a convex set still belongs to the convex set.

13. Let $A \in R^{m\times n}, b \in R^m$. Show, by definition, that

$$S = \{x \in R^n \mid Ax = b, x \geq 0\}$$

is a convex set.

14. Let

$$D_1 = \{x|\ x_1 + x_2 \leq 1,\ \ x_1 \geq 0\}, \quad D_2 = \{x|\ x_1 - x_2 \geq 0,\ \ x_1 \leq 0\}.$$

Set $D = D_1 \cup D_2$. Show that D is not necessarily convex though both D_1 and D_2 are convex. This means that the union of convex sets is not necessarily a convex set.

15. Write the convex hull of the set $S = \{(0,0)^T, (1,0)^T, (0,1)^T\}$.

16. Let $S \subseteq R^n$. Prove that the following two statements are equivalent.
(1) The convex hull is the set of all convex combinations of arbitrarily finitely many elements of S as defined in (1.3.3).
(2) The convex hull $\text{conv}(S)$ is the intersection of all convex sets containing S.

17. Let $f_i(x), i = 1, 2, \cdots, m$, be convex functions defined on convex set $D \subset R^n$. Show that the function

$$g(x) = \sum_{i=1}^{m} \alpha_i f_i(x)$$

is also a convex function on D, where $\sum_{i=1}^{m} \alpha_i = 1, \alpha_i \geq 0, i = 1, 2, \cdots, m$. This means that the convex combination of convex functions is a convex function.

18. Discriminate convexity of the following functions:
(1) $f(x_1, x_2) = x_1 e^{-(x_1+x_2)}$;
(2) $f(x_1, x_2, x_3) = x_1^2 + 3x_2^2 + 9x_3^2 - 2x_1x_2 + 6x_2x_3 + 2x_3x_1$.

19. Prove Theorem 1.3.11.

20. State the first-order and second-order optimality conditions for unconstrained optimization and outline their proofs.

Chapter 2

Line Search

2.1 Introduction

Line search, also called one-dimensional search, refers to an optimization procedure for univariable functions. It is the base of multivariable optimization. As stated before, in multivariable optimization algorithms, for given x_k, the iterative scheme is

$$x_{k+1} = x_k + \alpha_k d_k. \tag{2.1.1}$$

The key is to find the direction vector d_k and a suitable step size α_k. Let

$$\phi(\alpha) = f(x_k + \alpha d_k). \tag{2.1.2}$$

So, the problem that departs from x_k and finds a step size in the direction d_k such that

$$\phi(\alpha_k) < \phi(0)$$

is just line search about α.

If we find α_k such that the objective function in the direction d_k is minimized, i.e.,

$$f(x_k + \alpha_k d_k) = \min_{\alpha>0} f(x_k + \alpha d_k),$$

or

$$\phi(\alpha_k) = \min_{\alpha>0} \phi(\alpha),$$

such a line search is called exact line search or optimal line search, and α_k is called optimal step size. If we choose α_k such that the objective function has acceptable descent amount, i.e., such that the descent $f(x_k) - f(x_k + \alpha_k d_k) >$

0 is acceptable by users, such a line search is called inexact line search, or approximate line search, or acceptable line search.

Since, in practical computation, theoretically exact optimal step size generally cannot be found, and it is also expensive to find almost exact step size, therefore the inexact line search with less computation load is highly popular.

The framework of line search is as follows. First, determine or give an initial search interval which contains the minimizer; then employ some section techniques or interpolations to reduce the interval iteratively until the length of the interval is less than some given tolerance.

Next, we give a notation about the search interval and a simple method to determine the initial search interval.

Definition 2.1.1 *Let* $\phi : R \to R, \alpha^* \in [0, +\infty)$*, and*

$$\phi(\alpha^*) = \min_{\alpha \geq 0} \phi(\alpha).$$

If there exists a closed interval $[a, b] \subset [0, +\infty)$ *such that* $\alpha^* \in [a, b]$*, then* $[a, b]$ *is called a search interval for one-dimensional minimization problem* $\min_{\alpha \geq 0} \phi(\alpha)$*. Since the exact location of the minimum of* ϕ *over* $[a, b]$ *is not known, this interval is also called the interval of uncertainty.*

A simple method to determine an initial interval is called the forward-backward method. The basic idea of this method is as follows. Given an initial point and an initial steplength, we attempt to determine three points at which their function values show "high–low–high" geometry. If it is not successful to go forward, we will go backward. Concretely, given an initial point α_0 and a steplength $h_0 > 0$. If

$$\phi(\alpha_0 + h_0) < \phi(\alpha_0),$$

then, next step, depart from $\alpha_0 + h_0$ and continue going forward with a larger step until the objective function increases. If

$$\phi(\alpha_0 + h_0) > \phi(\alpha_0),$$

then, next step, depart from α_0 and go backward until the objective function increases. So, we will obtain an initial interval which contains the minimum α^*.

Algorithm 2.1.2 *(Forward-Backward Method)*

Step 1. *Given* $\alpha_0 \in [0, \infty), h_0 > 0$, *the multiple coefficient* $t > 1$ *(Usually* $t = 2$*). Evaluate* $\phi(\alpha_0), k := 0$.

Step 2. *Compare the objective function values. Set* $\alpha_{k+1} = \alpha_k + h_k$ *and evaluate* $\phi_{k+1} = \phi(\alpha_{k+1})$. *If* $\phi_{k+1} < \phi_k$, *go to Step 3; otherwise, go to Step 4.*

Step 3. *Forward step. Set* $h_{k+1} := th_k, \alpha := \alpha_k, \alpha_k := \alpha_{k+1}, \phi_k := \phi_{k+1}, k := k + 1$, *go to Step 2.*

Step 4. *Backward step. If* $k = 0$, *invert the search direction. Set* $h_k := -h_k, \alpha_k := \alpha_{k+1}$, *go to Step 2; otherwise, set*

$$a = \min\{\alpha, \alpha_{k+1}\},\ b = \max\{\alpha, \alpha_{k+1}\},$$

output $[a, b]$ *and stop.* □

The methods of line search presented in this chapter use the unimodality of the function and interval. The following definitions and theorem introduce their concepts and properties.

Definition 2.1.3 *Let* $\phi : R \to R, [a, b] \subset R$. *If there is* $\alpha^* \in [a, b]$ *such that* $\phi(\alpha)$ *is strictly decreasing on* $[a, \alpha^*]$ *and strictly increasing on* $[\alpha^*, b]$, *then* $\phi(\alpha)$ *is called a unimodal function on* $[a, b]$. *Such an interval* $[a, b]$ *is called a unimodal interval related to* $\phi(\alpha)$.

The unimodal function can also be defined as follows.

Definition 2.1.4 *If there exists a unique* $\alpha^* \in [a, b]$, *such that for any* $\alpha_1, \alpha_2 \in [a, b], \alpha_1 < \alpha_2$, *the following statements hold:*

if $\alpha_2 < \alpha^*$, *then* $\phi(\alpha_1) > \phi(\alpha_2)$;

if $\alpha_1 > \alpha^*$, *then* $\phi(\alpha_1) < \phi(\alpha_2)$;

then $\phi(\alpha)$ *is the unimodal function on* $[a, b]$.

Note that, first, the unimodal function does not require the continuity and differentiability of the function; second, using the property of the unimodal function, we can exclude portions of the interval of uncertainty that do not

contain the minimum, such that the interval of uncertainty is reduced. The following theorem shows that if the function ϕ is unimodal on $[a, b]$, then the interval of uncertainy could be reduced by comparing the function values of ϕ at two points within the interval.

Theorem 2.1.5 *Let $\phi : R \to R$ be unimodal on $[a, b]$. Let $\alpha_1, \alpha_2 \in [a, b]$, and $\alpha_1 < \alpha_2$. Then*

1. *if $\phi(\alpha_1) \leq \phi(\alpha_2)$, then $[a, \alpha_2]$ is a unimodal interval related to ϕ;*
2. *if $\phi(\alpha_1) \geq \phi(\alpha_2)$, then $[\alpha_1, b]$ is a unimodal interval related to ϕ.*

Proof. From the Definition 2.1.3, there exists $\alpha^* \in [a, b]$ such that $\phi(\alpha)$ is strictly decreasing over $[a, \alpha^*]$ and strictly increasing over $[\alpha^*, b]$. Since $\phi(\alpha_1) \leq \phi(\alpha_2)$, then $\alpha^* \in [a, \alpha_2]$ (see Figure 2.1.1). Since $\phi(\alpha)$ is unimodal on $[a, b]$, it is also unimodal on $[a, \alpha_2]$. Therefore $[a, \alpha_2]$ is a unimodal interval related to $\phi(\alpha)$ and the proof of the first part is complete.

The second part of the theorem can be proved similarly. □

This theorem indicates that, for reducing the interval of uncertainty, we must at least select two observations, evaluate and compare their function values.

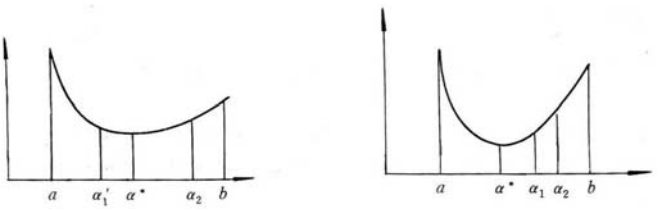


Figure 2.1.1 Properties of unimodal interval and unimodal function

2.2 Convergence Theory for Exact Line Search

The general form of an unconstrained optimization algorithm is as follows.

Algorithm 2.2.1 *(General Form of Unconstrained Optimization)*

Initial Step: Given $x_0 \in R^n, 0 \leq \epsilon \ll 1$.

k-th Step: Compute the descent direction d_k;
Compute the step size α_k, such that

$$f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha d_k); \tag{2.2.1}$$

Set

$$x_{k+1} = x_k + \alpha_k d_k; \tag{2.2.2}$$

If $\|\nabla f(x_{k+1})\| \leq \epsilon$, stop; otherwise, repeat the above steps. □

Set

$$\phi(\alpha) = f(x_k + \alpha d_k), \tag{2.2.3}$$

obviously we have from the algorithm that

$$\phi(0) = f(x_k),\ \phi(\alpha) \leq \phi(0).$$

In fact, (2.2.1) is to find the global minimizer of $\phi(\alpha)$ which is rather difficult. Instead, we look for the first stationary point, i.e., take α_k such that

$$\alpha_k = \min\{\alpha \geq 0 \mid \nabla f(x_k + \alpha d_k)^T d_k = 0\}. \tag{2.2.4}$$

Since, by (2.2.1) and (2.2.4), we find the exact minimizer and the stationary point of $\phi(\alpha)$ respectively, we say that (2.2.1) and (2.2.4) are exact line searches.

Let $\langle d_k, -\nabla f(x_k)\rangle$ denote the angle between d_k and $-\nabla f(x_k)$, we have

$$\cos\langle d_k, -\nabla f(x_k)\rangle = -\frac{d_k^T \nabla f(x_k)}{\|d_k\|\|\nabla f(x_k)\|}. \tag{2.2.5}$$

The following theorem gives a bound of descent in function values for each iteration in exact line search.

Theorem 2.2.2 *Let $\alpha_k > 0$ be the solution of (2.2.1). Let $\|\nabla^2 f(x_k + \alpha d_k)\| \leq M\ \forall \alpha > 0$, where M is some positive number. Then*

$$f(x_k) - f(x_k + \alpha_k d_k) \geq \frac{1}{2M}\|\nabla f(x_k)\|^2 \cos^2\langle d_k, -\nabla f(x_k)\rangle. \tag{2.2.6}$$

Proof. From the assumptions we have that

$$f(x_k + \alpha d_k) \leq f(x_k) + \alpha d_k^T \nabla f(x_k) + \frac{\alpha^2}{2} M \|d_k\|^2, \ \forall \alpha > 0. \tag{2.2.7}$$

Set $\bar{\alpha} = -d_k^T \nabla f(x_k)/(M\|d_k\|^2)$; it follows from the assumptions, (2.2.7) and (2.2.5) that

$$\begin{aligned} f(x_k) - f(x_k + \alpha_k d_k) &\geq f(x_k) - f(x_k + \bar{\alpha} d_k) \\ &\geq -\bar{\alpha} d_k^T \nabla f(x_k) - \frac{\bar{\alpha}^2}{2} M \|d_k\|^2 \\ &= \frac{1}{2} \frac{(d_k^T \nabla f(x_k))^2}{M\|d_k\|^2} \\ &= \frac{1}{2M} \|\nabla f(x_k)\|^2 \frac{(d_k^T \nabla f(x_k))^2}{\|d_k\|^2 \|\nabla f(x_k)\|^2} \\ &= \frac{1}{2M} \|\nabla f(x_k)\|^2 \cos^2\langle d_k, -\nabla f(x_k)\rangle. \square \end{aligned}$$

Now we are in position to state the convergence property of general unconstrained optimization algorithms with exact line search. The following two theorems state the convergence by different forms.

Theorem 2.2.3 *Let $f(x)$ be a continuously differentiable function on an open set $D \subset R^n$, assume that the sequence from Algorithm 2.2.1 satisfies $f(x_{k+1}) \leq f(x_k) \forall k$ and $\nabla f(x_k)^T d_k \leq 0$. Let $\bar{x} \in D$ be an accumulation point of $\{x_k\}$ and K_1 be an index set with $K_1 = \{k \mid \lim_{k\to\infty} x_k = \bar{x}\}$. Also assume that there exists $M > 0$ such that $\|d_k\| < M, \forall k \in K_1$. Then, if $\bar{d}$ is any accumulation point of $\{d_k\}$, we have*

$$\nabla f(\bar{x})^T \bar{d} = 0. \tag{2.2.8}$$

Furthermore, if $f(x)$ is twice continuously differentiable on D, then

$$\bar{d}^T \nabla^2 f(\bar{x}) \bar{d} \geq 0. \tag{2.2.9}$$

Proof. It is enough to prove (2.2.8) because the proof of (2.2.9) is similar.

Let $K_2 \subset K_1$ be an index set with $\bar{d} = \lim_{k\in K_2} d_k$. If $\bar{d} = 0$, (2.2.8) is trivial. Otherwise, we consider the following two cases.

(i) There exists an index set $K_3 \subset K_2$ such that $\lim_{k\in K_3} \alpha_k = 0$. Since α_k is an exact step size, then $\nabla f(x_k + \alpha_k d_k)^T d_k = 0$. Since $\|d_k\|$ is uniformly bounded above and $\alpha_k \to 0$, taking the limit yields

$$\nabla f(\bar{x})^T \bar{d} = 0.$$

(ii) Case of $\liminf_{k\in K_2} \alpha_k = \bar{\alpha} > 0$. Let $K_4 \subset K_2$ be an index set of k with $\alpha_k \geq \bar{\alpha}/2, \forall k \in K_4$. Now assume that the conclusion (2.2.8) is not true, then we have

$$\nabla f(\bar{x})^T \bar{d} < -\delta < 0.$$

So, there exist a neighborhood $N(\bar{x})$ of $\bar{x}$ and an index set $K_5 \subset K_4$ such that when $x \in N(\bar{x})$ and $k \in K_5$,

$$\nabla f(x)^T d_k \leq -\delta/2 < 0.$$

Let $\hat{\alpha}$ be a sufficiently small positive number, such that for all $0 \leq \alpha \leq \hat{\alpha}$ and all $k \in K_5$, $x_k + \alpha d_k \in N(\bar{x})$. Take $\alpha^* = \min(\bar{\alpha}/2, \hat{\alpha})$, then from the non-increasing property of the algorithm, exact line search and Taylor's expansion, we have

$$\begin{aligned}
f(\bar{x}) - f(x_0) &= \sum_{k=0}^{\infty} [f(x_{k+1}) - f(x_k)] \\
&\leq \sum_{k\in K_5} [f(x_{k+1}) - f(x_k)] \\
&\leq \sum_{k\in K_5} [f(x_k + \alpha^* d_k) - f(x_k)] \qquad (2.2.10) \\
&= \sum_{k\in K_5} \nabla f(x_k + \tau_k d_k)^T \alpha^* d_k \qquad (2.2.11) \\
&\leq \sum_{k\in K_5} -\left(\frac{\delta}{2}\right) \alpha^* \\
&= -\infty,
\end{aligned}$$

where $0 \leq \tau_k \leq \alpha^*$. The above contradiction shows that (2.2.8) also holds for case (ii).

The proof of (2.2.9) is similar. It is enough to note using the second-order form of the Taylor expansion instead of the first-order form in (2.2.11). In fact, from (2.2.10) we have

$$f(\bar{x}) - f(x_0)$$

$$
\begin{aligned}
&\leq \sum_{k\in K_5}[f(x_k+\alpha^* d_k)-f(x_k)]\\
&= \sum_{k\in K_5}\left[\nabla f(x_k)^T(\alpha^* d_k)+\frac{(\alpha^*)^2}{2}d_k^T\nabla^2 f(x_k+\tau_k d_k)d_k\right] \text{ for } 0\leq\tau_k\leq\alpha^*\\
&\leq \sum_{k\in K_5}\frac{(\alpha^*)^2}{2}d_k^T\nabla^2 f(x_k+\tau_k d_k)d_k \text{ for } 0\leq\tau_k\leq\alpha^*\\
&\leq \sum_{k\in K_5}\left[-\frac{1}{2}\left(\frac{\delta}{2}\right)(\alpha^*)^2\right]\\
&= -\infty. \qquad (2.2.12)
\end{aligned}
$$

We also get a contradiction which proves (2.2.9). □

Theorem 2.2.4 *Let $\nabla f(x)$ be uniformly continuous on the level set $L=\{x\in R^n \mid f(x)\leq f(x_0)\}$. Let also the angle θ_k between $-\nabla f(x_k)$ and the direction d_k generated by Algorithm 2.2.1 is uniformly bounded away from 90°, i.e., satisfies*

$$\theta_k\leq\frac{\pi}{2}-\mu, \text{ for some } \mu>0. \qquad (2.2.13)$$

Then $\nabla f(x_k)=0$ for some k; or $f(x_k)\to-\infty$; or $\nabla f(x_k)\to 0$.

Proof. Assume that, for all k, $\nabla f(x_k)\neq 0$ and $f(x_k)$ is bounded below. Since $\{f(x_k)\}$ is monotonic descent, its limit exists. Therefore

$$f(x_k)-f(x_{k+1})\to 0. \qquad (2.2.14)$$

Assume, by contradiction, that $\nabla f(x_k)\to 0$ does not hold. Then there exists $\epsilon>0$ and a subset K, such that $\|\nabla f(x_k)\|\geq\epsilon\forall k\in K$. Therefore

$$-\nabla f(x_k)^T d_k/\|d_k\| = \|\nabla f(x_k)\|\cos\theta_k\geq\epsilon\sin\mu\stackrel{\Delta}{=}\epsilon_1. \qquad (2.2.15)$$

Note that

$$
\begin{aligned}
&f(x_k+\alpha d_k)\\
&= f(x_k)+\alpha\nabla f(\xi_k)^T d_k\\
&= f(x_k)+\alpha\nabla f(x_k)^T d_k+\alpha[\nabla f(\xi_k)-\nabla f(x_k)]^T d_k\\
&\leq f(x_k)+\alpha\|d_k\|\left(\frac{\nabla f(x_k)^T d_k}{\|d_k\|}+\|\nabla f(\xi_k)-\nabla f(x_k)\|\right), \qquad (2.2.16)
\end{aligned}
$$

where ξ_k lies between x_k and $x_k + \alpha d_k$. Since $\nabla f(x)$ is uniformly continuous on the level set L, there exists $\bar{\alpha}$ such that when $0 \leq \alpha \|d_k\| \leq \bar{\alpha}$, we have

$$\|\nabla f(\xi_k) - \nabla f(x_k)\| \leq \frac{1}{2}\epsilon_1. \tag{2.2.17}$$

By (2.2.15)–(2.2.17), we get

$$\begin{aligned} f\left(x_k + \bar{\alpha}\frac{d_k}{\|d_k\|}\right) &\leq f(x_k) + \bar{\alpha}\left(\frac{\nabla f(x_k)^T d_k}{\|d_k\|} + \frac{1}{2}\epsilon_1\right) \\ &\leq f(x_k) - \frac{1}{2}\bar{\alpha}\epsilon_1. \end{aligned}$$

Therefore

$$f(x_{k+1}) \leq f\left(x_k + \bar{\alpha}\frac{d_k}{\|d_k\|}\right) \leq f(x_k) - \frac{1}{2}\bar{\alpha}\epsilon_1,$$

which contradicts (2.2.14). The contradiction shows that $\nabla f(x_k) \to 0$. We complete this proof. □

In the remainder of this section, we discuss the convergence rate of minimization algorithms with exact line search. For convenience of the proof of the theorem, we first give some lemmas.

Lemma 2.2.5 *Let $\phi(\alpha)$ be twice continuously differentiable on the closed interval $[0, b]$ and $\phi'(0) < 0$. If the minimizer $\alpha^* \in (0, b)$ of $\phi(\alpha)$ on $[0, b]$, then*

$$\alpha^* \geq \tilde{\alpha} = -\phi'(0)/M, \tag{2.2.18}$$

where M is a positive number such that $\phi''(\alpha) \leq M, \forall \alpha \in [0, b]$.

Proof. Construct the auxiliary function

$$\psi(\alpha) = \phi'(0) + M\alpha,$$

which has the unique zero

$$\tilde{\alpha} = -\phi'(0)/M.$$

Noting that $\phi''(\alpha) \leq M$, it follows that

$$\phi'(\alpha) = \phi'(0) + \int_0^\alpha \phi''(\alpha)\mathrm{d}\ \alpha \leq \phi'(0) + \int_0^\alpha M\mathrm{d}\ \alpha = \psi(\alpha).$$

Setting $\alpha = \alpha^*$ in the above inequality and noting that $\phi'(\alpha^*) = 0$, we obtain

$$0 \leq \psi(\alpha^*) = \phi'(0) + M\alpha^*$$

which is (2.2.18). □

Lemma 2.2.6 *Let $f(x)$ be twice continuously differentiable on R^n. Then for any vector $x, d \in R^n$ and any number α, the equality*

$$f(x + \alpha d) = f(x) + \alpha \nabla f(x)^T d + \alpha^2 \int_0^1 (1-t)[d^T \nabla^2 f(x + t\alpha d)d]dt \quad (2.2.19)$$

holds.

Proof. From calculus, we have

$$\begin{aligned} & f(x + \alpha d) - f(x) \\ = & \int_0^1 \mathrm{d} f(x + t\alpha d) \\ = & -\int_0^1 [\alpha \nabla f(x + t\alpha d)^T d]\mathrm{d}(1-t) \\ = & -[(1-t)\alpha \nabla f(x + t\alpha d)^T d]_0^1 + \int_0^1 (1-t)\mathrm{d}[\alpha \nabla f(x + t\alpha d)^T d] \\ = & \alpha \nabla f(x)^T d + \alpha^2 \int_0^1 [(1-t)d^T \nabla^2 f(x + t\alpha d)d]\mathrm{d}t. \quad \square \end{aligned}$$

Lemma 2.2.7 *Let $f(x)$ be twice continuously differentiable in the neighborhood of the minimizer x^*. Assume that there exist $\epsilon > 0$ and $M > m > 0$, such that*

$$m\|y\|^2 \leq y^T \nabla^2 f(x) y \leq M\|y\|^2, \ \forall y \in R^n \quad (2.2.20)$$

holds when $\|x - x^*\| < \epsilon$. Then we have

$$\frac{1}{2} m\|x - x^*\|^2 \leq f(x) - f(x^*) \leq \frac{1}{2} M\|x - x^*\|^2 \quad (2.2.21)$$

and

$$\|\nabla f(x)\| \geq m\|x - x^*\|. \quad (2.2.22)$$

Proof. From Lemma 2.2.6 we have

$$\begin{aligned} & f(x) - f(x^*) \\ = \; & \nabla f(x^*)^T(x - x^*) + \int_0^1 (1-t)(x-x^*)^T \nabla^2 f(tx + (1-t)x^*)(x - x^*)\mathrm{d}t \\ = \; & \int_0^1 (1-t)(x-x^*)^T \nabla^2 f(tx + (1-t)x^*)(x - x^*)\mathrm{d}t. \end{aligned} \tag{2.2.23}$$

Note that (2.2.20) and the integral mean-value theorem give

$$\begin{aligned} & m\|x - x^*\|^2 \int_0^1 (1-t)\mathrm{d}t \\ \leq \; & \int_0^1 (1-t)(x-x^*)^T \nabla^2 f(tx + (1-t)x^*)(x - x^*)\mathrm{d}t \\ \leq \; & M\|x - x^*\|^2 \int_0^1 (1-t)\mathrm{d}t. \end{aligned} \tag{2.2.24}$$

Then combining (2.2.23) and (2.2.24) yields (2.2.21).

Also, using Taylor expansion gives

$$\nabla f(x) = \nabla f(x) - \nabla f(x^*) = \int_0^1 \nabla^2 f(tx + (1-t)x^*)(x - x^*)\mathrm{d}t.$$

Then

$$\begin{aligned} \|\nabla f(x)\|\|x - x^*\| &\geq (x - x^*)^T \nabla f(x) \\ &= \int_0^1 (x-x^*)^T \nabla^2 f(tx + (1-t)x^*)(x - x^*)\mathrm{d}t \\ &\geq m\|x - x^*\|^2 \end{aligned}$$

which proves (2.2.22). □

Now we are in the position to give the theorem about convergence rate which shows that the local convergence rate of Algorithm 2.2.1 with exact line search is at least linear.

Theorem 2.2.8 *Let the sequence $\{x_k\}$ generated by Algorithm 2.2.1 converge to the minimizer x^* of $f(x)$. Let $f(x)$ be twice continuously differentiable in a neighborhood of x^*. If there exist $\epsilon > 0$ and $M > m > 0$ such that when $\|x - x^*\| < \epsilon$,*

$$m\|y\|^2 \leq y^T \nabla^2 f(x) y \leq M\|y\|^2, \; \forall y \in R^n \tag{2.2.25}$$

holds, then the sequence $\{x_k\}$, at least, converges linearly to x^*.

Proof. Let $\lim_{k\to\infty} x_k = x^*$. We may assume that $\|x_k - x^*\| \le \epsilon$ for k sufficiently large. Since $\|x_{k+1} - x^*\| < \epsilon$, there exists $\delta > 0$ such that

$$\|x_k + (\alpha_k + \delta)d_k - x^*\| = \|x_{k+1} - x^* + \delta d_k\| < \epsilon. \tag{2.2.26}$$

Note that $\phi(\alpha) = f(x_k + \alpha d_k), \phi'(\alpha) = \nabla f(x_k + \alpha d_k)^T d_k, \phi'(0) = \nabla f(x_k)^T d_k$ and $|\phi'(0)| \le \|\nabla f(x_k)\|\|d_k\|$. We have $\phi'(0) < 0$,

$$\rho\|\nabla f(x_k)\|\|d_k\| \le -\phi'(0) \le \|\nabla f(x_k)\|\|d_k\|, \text{ for some } \rho \in (0,1) \tag{2.2.27}$$

and

$$\phi''(\alpha) = d_k^T \nabla^2 f(x_k + \alpha d_k) d_k \le M\|d_k\|^2.$$

Then, by Lemma 2.2.5, we know that the minimizer α_k of $\phi(\alpha)$ on $[0, \alpha_k + \delta]$ satisfies

$$\alpha_k \ge \tilde{\alpha}_k = \frac{-\phi'(0)}{M\|d_k\|^2} \ge \frac{\rho\|\nabla f(x_k)\|}{M\|d_k\|} \triangleq \bar{\alpha}_k. \tag{2.2.28}$$

Set $\bar{x}_k = x_k + \bar{\alpha}_k d_k$. Obviously, it follows from (2.2.26) that $\|\bar{x}_k - x^*\| < \epsilon$. Therefore,

$$\begin{aligned}
& f(x_k + \alpha_k d_k) - f(x_k) \\
\le\ & f(x_k + \bar{\alpha}_k d_k) - f(x_k) \\
=\ & \bar{\alpha}_k \nabla f(x_k)^T d_k + \bar{\alpha}_k^2 \int_0^1 (1-t) d_k^T \nabla^2 f(x_k + t\bar{\alpha}_k d_k) d_k \mathrm{d}t \text{ (from Lemma 2.2.6)} \\
\le\ & \bar{\alpha}_k(-\rho)\|\nabla f(x_k)\|\|d_k\| + \frac{1}{2} M \bar{\alpha}_k^2 \|d_k\|^2 \text{ (from (2.2.25) and (2.2.27))} \\
\le\ & -\frac{\rho^2}{2M}\|\nabla f(x_k)\|^2 \text{ (from (2.2.28))} \\
\le\ & -\frac{\rho^2}{2M} m^2 \|x_k - x^*\|^2 \text{ (from (2.2.22))} \\
\le\ & -\left(\frac{\rho m}{M}\right)^2 [f(x_k) - f(x^*)] \text{ (from (2.2.21))}.
\end{aligned}$$

The above inequalities give

$$\begin{aligned}
f(x_{k+1}) - f(x^*) &= [f(x_{k+1}) - f(x_k)] + [f(x_k) - f(x^*)] \\
&\le \left[1 - \left(\frac{\rho m}{M}\right)^2\right] [f(x_k) - f(x^*)].
\end{aligned} \tag{2.2.29}$$

Set

$$\theta = \left[1 - \left(\frac{\rho m}{M}\right)^2\right]^{\frac{1}{2}}. \tag{2.2.30}$$

Obviously $\theta \in (0, 1)$. Therefore (2.2.29) can be written as

$$\begin{aligned} f(x_k) - f(x^*) &\leq \theta^2[f(x_{k-1}) - f(x^*)] \\ &\leq \cdots \\ &\leq \theta^{2k}[f(x_0) - f(x^*)]. \end{aligned} \tag{2.2.31}$$

Furthermore, by (2.2.21), we have

$$\begin{aligned} \|x_k - x^*\|^2 &\leq \frac{2}{m}[f(x_k) - f(x^*)] \\ &\leq \frac{2}{m}\theta^2[f(x_{k-1}) - f(x^*)] \\ &\leq \frac{2}{m}\theta^2\frac{M}{2}\|x_{k-1} - x^*\|^2 \end{aligned}$$

which implies that

$$\|x_k - x^*\| \leq \sqrt{\frac{M}{m}}\theta\|x_{k-1} - x^*\| \tag{2.2.32}$$

and that the sequence $\{x_k\}$, at least, converges linearly to x^*. □

In the end of this section, we give a theorem which describes a descent bound of the function value after each exact line search.

Theorem 2.2.9 *Let α_k be an exact step size. Assume that $f(x)$ satisfies*

$$(x - z)^T[\nabla f(x) - \nabla f(z)] \geq \eta\|x - z\|^2. \tag{2.2.33}$$

Then

$$f(x_k) - f(x_k + \alpha_k d_k) \geq \frac{1}{2}\eta\|\alpha_k d_k\|^2. \tag{2.2.34}$$

Proof. Since α_k is an exact step size, then

$$\nabla f(x_k + \alpha_k d_k)^T d_k = 0. \tag{2.2.35}$$

Therefore, it follows from the mean-value theorem, (2.2.35) and (2.2.33) that

$$\begin{aligned} f(x_k) - f(x_k + \alpha_k d_k) &= \int_0^{\alpha_k} -d_k^T \nabla f(x_k + t d_k) \mathrm{d}t \\ &= \int_0^{\alpha_k} d_k^T [\nabla f(x_k + \alpha_k d_k) - \nabla f(x_k + t d_k)] \mathrm{d}t \\ &\geq \int_0^{\alpha_k} \eta(\alpha_k - t) \mathrm{d}t \|d_k\|^2 \\ &= \frac{1}{2} \eta \|\alpha_k d_k\|^2. \end{aligned} \tag{2.2.36}$$

This completes the proof. □

2.3 The Golden Section Method and the Fibonacci Method

The golden section method and the Fibonacci method are section methods. Their basic idea for minimizing a unimodal function over $[a, b]$ is iteratively reducing the interval of uncertainty by comparing the function values of the observations. When the length of the interval of uncertainty is reduced to some desired degree, the points on the interval can be regarded as approximations of the minimizer. Such a class of methods only needs to evaluate the functions and has wide applications, especially it is suitable to nonsmooth problems and those problems with complicated derivative expressions.

2.3.1 The Golden Section Method

Let

$$\phi(\alpha) = f(x + \alpha d)$$

be a unimodal function on the interval $[a, b]$. At the iteration k of the golden section method, let the interval of uncertainty be $[a_k, b_k]$. Take two observations $\lambda_k, \mu_k \in [a_k, b_k]$ and $\lambda_k < \mu_k$. Evaluate $\phi(\lambda_k)$ and $\phi(\mu_k)$. By Theorem 2.1.5, we have

Case 1 if $\phi(\lambda_k) \leq \phi(\mu_k)$, then set $a_{k+1} = a_k, b_{k+1} = \mu_k$;

Case 2 if $\phi(\lambda_k) > \phi(\mu_k)$, then set $a_{k+1} = \lambda_k, b_{k+1} = b_k$.

How to choose the observations λ_k and μ_k? We require that λ_k and μ_k satisfy the following conditions:

1. The distances from λ_k and μ_k to the end points of the interval $[a_k, b_k]$ are equivalent, that is,

$$b_k - \lambda_k = \mu_k - a_k. \tag{2.3.1}$$

2. The reduction rate of the intervals of uncertainty for each iteration is the same, that is

$$b_{k+1} - a_{k+1} = \tau(b_k - a_k),\ \tau \in (0, 1). \tag{2.3.2}$$

3. Only one extra observation is needed for each new iteration.

Now we consider Case 1. Substituting the values of Case 1 into (2.3.2) and combining (2.3.1) yield

$$\begin{aligned} \mu_k - a_k &= \tau(b_k - a_k), \\ b_k - \lambda_k &= \mu_k - a_k. \end{aligned}$$

Arranging the above equations gives

$$\lambda_k = a_k + (1 - \tau)(b_k - a_k), \tag{2.3.3}$$
$$\mu_k = a_k + \tau(b_k - a_k). \tag{2.3.4}$$

Note that, in this case, the new interval is $[a_{k+1}, b_{k+1}] = [a_k, \mu_k]$. For further reducing the interval of uncertainty, the observations λ_{k+1} and μ_{k+1} are selected. By (2.3.4),

$$\begin{aligned} \mu_{k+1} &= a_{k+1} + \tau(b_{k+1} - a_{k+1}) \\ &= a_k + \tau(\mu_k - a_k) \\ &= a_k + \tau(a_k + \tau(b_k - a_k) - a_k) \\ &= a_k + \tau^2(b_k - a_k). \end{aligned} \tag{2.3.5}$$

If we set

$$\tau^2 = 1 - \tau, \tag{2.3.6}$$

then

$$\mu_{k+1} = a_k + (1 - \tau)(b_k - a_k) = \lambda_k. \tag{2.3.7}$$

It means that the new observation μ_{k+1} does not need to compute, because μ_{k+1} coincides with λ_k.

Similarly, if we consider Case 2, the new observation λ_{k+1} coincides with μ_k. Therefore, for each new iteration, only one extra observation is needed, which is just required by the third condition.

What is the reduction rate of the interval of uncertainty for each iteration? By solving the equation (2.3.6), we immediately obtain

$$\tau = \frac{-1 \pm \sqrt{5}}{2}.$$

Since $\tau > 0$, then take

$$\tau = \frac{b_{k+1} - a_{k+1}}{b_k - a_k} = \frac{\sqrt{5} - 1}{2} \approx 0.618. \tag{2.3.8}$$

Then the formula (2.3.3)–(2.3.4) can be written as

$$\lambda_k = a_k + 0.382(b_k - a_k), \tag{2.3.9}$$
$$\mu_k = a_k + 0.618(b_k - a_k). \tag{2.3.10}$$

Therefore, the golden section method is also called the 0.618 method.

Obviously, comparing with the Fibonacci method below, the golden section method is more simple in performance and we need not know the number of observations in advance.

Since, for each iteration, the reduction rate of the interval of uncertainty is $\tau = 0.618$, then if the initial interval is $[a_1, b_1]$, the length of the interval after n-th iteration is $\tau^{n-1}(b_1 - a_1)$. Therefore the convergence rate of the golden section method is linear.

Algorithm 2.3.1 *(The Golden Section Method)*

Step 1. Initial step. Determine the initial interval $[a_1, b_1]$ *and give the precision* $\delta > 0$. *Compute initial observations* λ_1 *and* μ_1*:*

$$\lambda_1 = a_1 + 0.382(b_1 - a_1),$$
$$\mu_1 = a_1 + 0.618(b_1 - a_1),$$

evaluate $\phi(\lambda_1)$ *and* $\phi(\mu_1)$*, set* $k = 1$.

Step 2. Compare the function values. If $\phi(\lambda_k) > \phi(\mu_k)$*, go to Step 3; if* $\phi(\lambda_k) \leq \phi(\mu_k)$*, go to Step 4.*

Step 3. (Case 2) If $b_k - \lambda_k \le \delta$, stop and output μ_k; otherwise, set

$$a_{k+1} := \lambda_k, b_{k+1} := b_k, \lambda_{k+1} := \mu_k,$$
$$\phi(\lambda_{k+1}) := \phi(\mu_k), \mu_{k+1} := a_{k+1} + 0.618(b_{k+1} - a_{k+1}).$$

Evaluate $\phi(\mu_{k+1})$ and go to Step5.

Step 4. (Case 1) If $\mu_k - a_k \le \delta$, stop and output λ_k; otherwise set

$$a_{k+1} := a_k, b_{k+1} := \mu_k, \mu_{k+1} := \lambda_k,$$
$$\phi(\mu_{k+1}) := \phi(\lambda_k), \lambda_{k+1} := a_{k+1} + 0.382(b_{k+1} - a_{k+1}).$$

Evaluate $\phi(\lambda_{k+1})$ and go to Step 5.

Step 5. $k := k + 1$, go to Step 2. □

2.3.2 The Fibonacci Method

Another section method which is similar to the golden section method is the Fibonacci method. Their main difference is in that the reduction rate of the interval of uncertainty for the Fibonacci method does not use the golden section number $\tau \approx 0.618$, but uses the Fibonacci number. Therefore the reduction of the interval of uncertainty varies from one iteration to another.

The Fibonacci sequence $\{F_k\}$ is defined as follows:

$$F_0 = F_1 = 1, \tag{2.3.11}$$
$$F_{k+1} = F_k + F_{k-1},\ k = 1, 2, \cdots. \tag{2.3.12}$$

If we use F_{n-k}/F_{n-k+1} instead of τ in (2.3.3)–(2.3.4), we immediately obtain the formula

$$\begin{aligned} \lambda_k &= a_k + \left(1 - \frac{F_{n-k}}{F_{n-k+1}}\right)(b_k - a_k) \qquad (2.3.13)\\ &= a_k + \frac{F_{n-k-1}}{F_{n-k+1}}(b_k - a_k),\ k = 1, \cdots, n-1, \\ \mu_k &= a_k + \frac{F_{n-k}}{F_{n-k+1}}(b_k - a_k),\ k = 1, \cdots, n-1, \qquad (2.3.14) \end{aligned}$$

which is called the Fibonacci formula.

As stated in the last section, in Case 1, if $\phi(\lambda_k) \leq \phi(\mu_k)$, the new interval of uncertainty is $[a_{k+1}, b_{k+1}] = [a_k, \mu_k]$. So, by using (2.3.14), we get

$$b_{k+1} - a_{k+1} = \frac{F_{n-k}}{F_{n-k+1}}(b_k - a_k) \tag{2.3.15}$$

which gives a reduction in each iteration. This equality is also true for Case 2.

Assume that we ask for the length of the final interval no more than δ, i.e.,

$$b_n - a_n \leq \delta.$$

Since

$$\begin{aligned} b_n - a_n &= \frac{F_1}{F_2}(b_{n-1} - a_{n-1}) \\ &= \frac{F_1}{F_2}\frac{F_2}{F_3} \cdots \frac{F_{n-1}}{F_n}(b_1 - a_1) \\ &= \frac{1}{F_n}(b_1 - a_1), \end{aligned} \tag{2.3.16}$$

then

$$F_n \geq \frac{b_1 - a_1}{\delta}. \tag{2.3.17}$$

Therefore, given initial interval $[a_1, b_1]$ and the upper bound δ of the length of the final interval, we can find the Fibonacci number F_n and further n from (2.3.17). Our search proceeds until the n-th observation. The procedure of the Fibonacci method is similar to Algorithm 2.3.1. We leave it as an exercise.

Letting $F_k = r^k$ and substituting in (2.3.11)-(2.3.12), we get

$$r^2 - r - 1 = 0. \tag{2.3.18}$$

Solving (2.3.18) gives

$$r_1 = \frac{1 + \sqrt{5}}{2}, \; r_2 = \frac{1 - \sqrt{5}}{2}. \tag{2.3.19}$$

Then, the general solution of the difference equation $F_{k+1} = F_k + F_{k-1}$ is

$$F_k = Ar_1^k + Br_2^k. \tag{2.3.20}$$

Using the initial condition $F_0 = F_1 = 1$, we get

$$A = \frac{r_1}{\sqrt{5}},\ B = -\frac{r_2}{\sqrt{5}}.$$

Substituting in (2.3.20) gives

$$F_k = \frac{1}{\sqrt{5}} \left\{ \left(\frac{1+\sqrt{5}}{2} \right)^{k+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{k+1} \right\}. \tag{2.3.21}$$

Hence

$$\lim_{k\to\infty} \frac{F_{k-1}}{F_k} = \frac{\sqrt{5}-1}{2} = \tau. \tag{2.3.22}$$

This shows that, when $k \to \infty$, the Fibonacci method and the golden section method have the same reduction rate of the interval of uncertainty. Therefore the Fibonacci method converges with convergence ratio τ. It is worth mentioning that the Fibonacci method is the optimal sectioning method for one-dimensional optimization and it requires the smallest observations for a given final length δ, and that the golden section method is approximately optimal. However, since the procedure of the golden section method is very simple, it is more popular.

2.4 Interpolation Method

Interpolation Methods are the other approach of line search. This class of methods approximates $\phi(\alpha) = f(x + \alpha d)$ by fitting a quadratic or cubic polynomial in α to known data, and choosing a new α-value which minimizes the polynomial. Then we reduce the bracketing interval by comparing the new α-value and the known points. In general, when the function has good analytical properties, for example, it is easy to get the derivatives, the interpolation methods are superior to the golden section method and the Fibonacci method discussed in the last subsection.

2.4.1 Quadratic Interpolation Methods

1. Quadratic Interpolation Method with Two Points (I).
Given two points α_1, α_2, and their function values $\phi(\alpha_1)$ and $\phi(\alpha_2)$, and the derivative $\phi'(\alpha_1)$ (or $\phi'(\alpha_2)$). Construct the quadratic interpolation function

$q(\alpha) = a\alpha^2 + b\alpha + c$ with the interpolation conditions:

$$\begin{aligned} q(\alpha_1) &= a\alpha_1^2 + b\alpha_1 + c = \phi(\alpha_1), \\ q(\alpha_2) &= a\alpha_2^2 + b\alpha_2 + c = \phi(\alpha_2), \\ q'(\alpha_1) &= 2a\alpha_1 + b = \phi'(\alpha_1). \end{aligned} \tag{2.4.1}$$

Write $\phi_1 = \phi(\alpha_1), \phi_2 = \phi(\alpha_2), \phi'_1 = \phi'(\alpha_1)$, and $\phi'_2 = \phi'(\alpha_2)$. Solving (2.4.1) gives

$$\begin{aligned} a &= \frac{\phi_1 - \phi_2 - \phi'_1(\alpha_1 - \alpha_2)}{-(\alpha_1 - \alpha_2)^2}, \\ b &= \phi'_1 + 2\frac{\phi_1 - \phi_2 - \phi'_1(\alpha_1 - \alpha_2)}{(\alpha_1 - \alpha_2)^2}\alpha_1. \end{aligned}$$

Hence

$$\begin{aligned} \bar{\alpha} &= -\frac{b}{2a} \\ &= \alpha_1 + \frac{1}{2}\frac{\phi'_1(\alpha_1 - \alpha_2)^2}{\alpha_1 - \alpha_2 - \phi'_1(\alpha_1 - \alpha_2)} \\ &= \alpha_1 - \frac{1}{2}\frac{(\alpha_1 - \alpha_2)\phi'_1}{\phi'_1 - \frac{\phi_1 - \phi_2}{\alpha_1 - \alpha_2}}. \end{aligned} \tag{2.4.2}$$

Then we get the following iteration formula:

$$\alpha_{k+1} = \alpha_k - \frac{1}{2}\frac{(\alpha_k - \alpha_{k-1})\phi'_k}{\phi'_k - \frac{\phi_k - \phi_{k-1}}{\alpha_k - \alpha_{k-1}}}. \tag{2.4.3}$$

where $\phi_k = \phi(\alpha_k), \phi_{k-1} = \phi(\alpha_{k-1})$, and $\phi'_k = \phi'(\alpha_k)$.

After finding the new α_{k+1}, we compare α_{k+1} with α_k and α_{k-1}, and reduce the bracketing interval. The procedure will continue until the length of the interval is less than a prescribed tolerance.

2. Quadratic Interpolation Method with Two Points (II).
Given two points α_1, α_2, and one function value $\phi(\alpha_1)$ (or $\phi(\alpha_2)$), and two derivative values $\phi'(\alpha_1)$ and $\phi'(\alpha_2)$. Construct the quadratic interpolation function with the following conditions:

$$\begin{aligned} q(\alpha_1) &= a\alpha_1^2 + b\alpha_1 + c = \phi(\alpha_1), \\ q'(\alpha_1) &= 2a\alpha_1 + b = \phi'(\alpha_1), \\ q'(\alpha_2) &= 2a\alpha_2 + b = \phi'(\alpha_2). \end{aligned} \tag{2.4.4}$$

Precisely, with the same discussion as above we obtain

$$\bar{\alpha} = -\frac{b}{2a} = \alpha_1 - \frac{\alpha_1 - \alpha_2}{\phi'_1 - \phi'_2}\phi'_1. \tag{2.4.5}$$

Therefore the iteration scheme is

$$\alpha_{k+1} = \alpha_k - \frac{\alpha_k - \alpha_{k-1}}{\phi'_k - \phi'_{k-1}}\phi'_k \tag{2.4.6}$$

which is also called the secant formula. The formula (2.4.5) can also be got by setting $L(\alpha) = 0$ where $L(\alpha)$ is the Lagrange interpolation formula

$$L(\alpha) = \frac{(\alpha - \alpha_1)\phi'_2 - (\alpha - \alpha_2)\phi'_1}{\alpha_2 - \alpha_1} \tag{2.4.7}$$

which interpolates the derivative values of $\phi'(\alpha)$ at two points α_1 and α_2.

In the following discussion, for convenience, we call the quadratic interpolating method (I) the quadratic interpolation formula, and the quadratic interpolation method (II) the secant formula. Next, we turn to the convergence of the quadratic interpolating method with two points.

Theorem 2.4.1 *Let $\phi : R \to R$ be three times continuously differentiable. Let α^* be such that $\phi'(\alpha^*) = 0$ and $\phi''(\alpha^*) \neq 0$. Then the sequence $\{\alpha_k\}$ generated from (2.4.6) converges to α^* with the order $\frac{1+\sqrt{5}}{2} \approx 1.618$ of convergence rate.*

Proof. By the representation of residual term of the Lagrange interpolation formula

$$R_2(\alpha) = \phi'(\alpha) - L(\alpha) = \frac{1}{2}\phi'''(\xi)(\alpha - \alpha_k)(\alpha - \alpha_{k-1}),\ \xi \in (\alpha, \alpha_{k-1}, \alpha_k). \tag{2.4.8}$$

Setting $\alpha = \alpha_{k+1}$ and noting that $L(\alpha_{k+1}) = 0$, we have

$$\phi'(\alpha_{k+1}) = \frac{1}{2}\phi'''(\xi)(\alpha_{k+1} - \alpha_k)(\alpha_{k+1} - \alpha_{k-1}),\ \xi \in (\alpha_{k-1}, \alpha_k, \alpha_{k+1}), \tag{2.4.9}$$

Substituting (2.4.6) into (2.4.9) yields

$$\phi'(\alpha_{k+1}) = \frac{1}{2}\phi'''(\xi)\phi'_k\phi'_{k-1}\frac{(\alpha_k - \alpha_{k-1})^2}{(\phi'_k - \phi'_{k-1})^2},\ \xi \in (\alpha_{k-1}, \alpha_k, \alpha_{k+1}). \tag{2.4.10}$$

We know from the mean-value theorem of differentiation that

$$\frac{\phi'_k - \phi'_{k-1}}{\alpha_k - \alpha_{k-1}} = \phi''(\xi_0),\ \xi_0 \in (\alpha_{k-1}, \alpha_k), \tag{2.4.11}$$

$$\phi'_i = \phi'_i - \phi'(\alpha^*) = (\alpha_i - \alpha^*)\phi''(\xi_i), \tag{2.4.12}$$

where $\xi_i \in (\alpha_i, \alpha^*), i = k-1, k, k+1$. Therefore it follows from (2.4.10)-(2.4.12) that

$$\alpha_{k+1} - \alpha^* = \frac{1}{2}\frac{\phi'''(\xi)\phi''(\xi_k)\phi''(\xi_{k-1})}{\phi''(\xi_{k+1})[\phi''(\xi_0)]^2}(\alpha_k - \alpha^*)(\alpha_{k-1} - \alpha^*). \tag{2.4.13}$$

Let $e_i = |\alpha_i - \alpha^*|, (i = k-1, k, k+1)$. In the intervals considered, let

$$0 < m_2 \le |\phi'''(\alpha)| \le M_2,\ 0 < m_1 \le |\phi''(\alpha)| \le M_1,$$

$$K_1 = m_2 m_1^2/(2M_1^3),\ K = M_2 M_1^2/(2m_1^3).$$

Then

$$K_1|\alpha_k - \alpha^*||\alpha_{k-1} - \alpha^*| \le |\alpha_{k+1} - \alpha^*| \le K|\alpha_k - \alpha^*||\alpha_{k-1} - \alpha^*|. \tag{2.4.14}$$

Noting that ϕ'' and ϕ''' are continuous at α^*, we get

$$\frac{\alpha_{k+1} - \alpha^*}{(\alpha_k - \alpha^*)(\alpha_{k-1} - \alpha^*)} \to \frac{1}{2}\frac{\phi'''(\alpha^*)}{\phi''(\alpha^*)}. \tag{2.4.15}$$

Therefore

$$e_{k+1} = \left|\frac{\phi'''(\eta_1)}{2\phi''(\eta_2)}\right| e_k e_{k-1} \triangleq M e_k e_{k-1}, \tag{2.4.16}$$

where $\eta_1 \in (\alpha_{k-1}, \alpha_k, \alpha^*), \eta_2 \in (\alpha_{k-1}, \alpha_k), M = |\phi'''(\eta_1)/2\phi''(\eta_2)|$. The above relations indicate that there exists $\delta > 0$ such that, when the initial points $\alpha_0, \alpha_1 \in (\alpha^* - \delta, \alpha^* + \delta)$ and $\alpha_0 \ne \alpha_1$, the sequence $\{\alpha_k\} \to \alpha^*$.

Next, we consider the convergence rate. Set $\epsilon_i = Me_i, y_i = \ln \epsilon_i, i = k-1, k, k+1$, then

$$\epsilon_{k+1} = \epsilon_k \epsilon_{k-1}, \tag{2.4.17}$$

$$y_{k+1} = y_k + y_{k-1}. \tag{2.4.18}$$

Obviously, (2.4.18) is the equation that the Fibonacci sequence satisfies, and its characteristic equation is

$$t^2 - t - 1 = 0 \tag{2.4.19}$$

whose solutions are

$$t_1 = \frac{1+\sqrt{5}}{2},\ t_2 = \frac{1-\sqrt{5}}{2}. \tag{2.4.20}$$

Therefore the Fibonacci sequence $\{y_k\}$ can be written as

$$y_k = At_1^k + Bt_2^k,\ k = 0, 1, \cdots, \tag{2.4.21}$$

where A and B are coefficients to be determined. Obviously, when $k \to \infty$,

$$\ln \epsilon_k = y_k \approx At_1^k. \tag{2.4.22}$$

Since

$$\frac{\epsilon_{k+1}}{\epsilon_k^{t_1}} \approx \frac{\exp(At_1^{k+1})}{[\exp(At_1^k)]^{t_1}} = 1,$$

then

$$\frac{e_{k+1}}{e_k^{t_1}} \approx M^{t_1-1} \tag{2.4.23}$$

which implies that the convergence rate is $t_1 = \frac{1+\sqrt{5}}{2} \approx 1.618$. □

This theorem tells us that the secant method has superlinear convergence.

3. Quadratic Interpolation Method with Three Points.
Given three distinct points α_1, α_2 and α_3, and their function values. The required interpolation conditions are

$$q(\alpha_i) = a\alpha_i^2 + b\alpha_i + c = \phi(\alpha_i),\ i = 1, 2, 3. \tag{2.4.24}$$

By solving the above equations, we obtain

$$a = -\frac{(\alpha_2-\alpha_3)\phi_1 + (\alpha_3-\alpha_1)\phi_2 + (\alpha_1-\alpha_2)\phi_3}{(\alpha_1-\alpha_2)(\alpha_2-\alpha_3)(\alpha_3-\alpha_1)},$$

$$b = \frac{(\alpha_2^2-\alpha_3^2)\phi_1 + (\alpha_3^2-\alpha_1^2)\phi_2 + (\alpha_1^2-\alpha_2^2)\phi_3}{(\alpha_1-\alpha_2)(\alpha_2-\alpha_3)(\alpha_3-\alpha_1)}.$$

Then

$$\begin{aligned} \bar{\alpha} &= -\frac{b}{2a} \\ &= \frac{1}{2}\frac{(\alpha_2^2-\alpha_3^2)\phi_1 + (\alpha_3^2-\alpha_1^2)\phi_2 + (\alpha_1^2-\alpha_2^2)\phi_3}{(\alpha_2-\alpha_3)\phi_1 + (\alpha_3-\alpha_1)\phi_2 + (\alpha_1-\alpha_2)\phi_3} && (2.4.25) \\ &= \frac{1}{2}(\alpha_1+\alpha_2) + \frac{1}{2}\frac{(\phi_1-\phi_2)(\alpha_2-\alpha_3)(\alpha_3-\alpha_1)}{(\alpha_2-\alpha_3)\phi_1 + (\alpha_3-\alpha_1)\phi_2 + (\alpha_1-\alpha_2)\phi_3} && (2.4.26) \end{aligned}$$

Equations (2.4.25) and (2.4.26) are called the quadratic interpolation formula with three points. The above formula can also be obtained from considering the Lagrange interpolation formula

$$L(\alpha) = \frac{(\alpha-\alpha_2)(\alpha-\alpha_3)}{(\alpha_1-\alpha_2)(\alpha_1-\alpha_3)}\phi_1 + \frac{(\alpha-\alpha_1)(\alpha-\alpha_3)}{(\alpha_2-\alpha_1)(\alpha_2-\alpha_3)}\phi_2 + \frac{(\alpha-\alpha_1)(\alpha-\alpha_2)}{(\alpha_3-\alpha_1)(\alpha_3-\alpha_2)}\phi_3, \tag{2.4.27}$$

and setting $L'(\alpha) = 0$.

Algorithm 2.4.2 *(Line Search Employing Quadratic Interpolation with Three Points)*

Step 0. Given tolerance ϵ. Find an initial bracket $\{\alpha_1, \alpha_2, \alpha_3\}$ containing α^*; Compute $\phi(\alpha_i), i = 1, 2, 3$.

Step 1. Use the formula (2.4.25) to produce $\bar{\alpha}$;

Step 2. If $(\bar{\alpha} - \alpha_1)(\bar{\alpha} - \alpha_3) \geq 0$ go to Step 3; otherwise go to Step 4;

Step 3. Construct new bracket $\{\alpha_1, \alpha_2, \alpha_3\}$ from $\alpha_1, \alpha_2, \alpha_3$ and $\bar{\alpha}$. Go to Step 1.

Step 4. If $|\bar{\alpha} - \alpha_2| < \varepsilon$, stop; otherwise go to Step 3. □

Figure 2.4.1 is a diagram for the quadratic interpolation line search with three points.

The following theorem shows that the above algorithm has convergence rate with order 1.32.

Theorem 2.4.3 *Let $\phi(\alpha)$ have continuous fourth-order derivatives. Let α^* satisfy $\phi'(\alpha^*) = 0$ and $\phi''(\alpha^*) \neq 0$. Then the sequence $\{\alpha_k\}$ generated from the formula (2.4.25) has convergence rate with order 1.32.*

Proof. By Lagrange interpolation formula (2.4.27), we have

$$\phi(\alpha) = L(\alpha) + R_3(\alpha), \tag{2.4.28}$$

where

$$R_3(\alpha) = \frac{1}{6}\phi'''(\xi(\alpha))(\alpha-\alpha_1)(\alpha-\alpha_2)(\alpha-\alpha_3). \tag{2.4.29}$$

Since $0 = \phi'(\alpha^*) = L'(\alpha^*) + R_3'(\alpha^*)$, we get

$$\begin{aligned}&\phi_1 \frac{2\alpha^* - (\alpha_2+\alpha_3)}{(\alpha_1-\alpha_2)(\alpha_1-\alpha_3)} + \phi_2 \frac{2\alpha^* - (\alpha_3+\alpha_1)}{(\alpha_2-\alpha_3)(\alpha_2-\alpha_1)} \\ &+\phi_3 \frac{2\alpha^* - (\alpha_1+\alpha_2)}{(\alpha_3-\alpha_1)(\alpha_3-\alpha_2)} + R_3'(\alpha^*) = 0. \end{aligned} \tag{2.4.30}$$

Noting that (2.4.25) can be rewritten as

$$\alpha_4 = \frac{1}{2} \frac{\frac{\phi_1(\alpha_2+\alpha_3)}{(\alpha_1-\alpha_2)(\alpha_1-\alpha_3)} + \frac{\phi_2(\alpha_3+\alpha_1)}{(\alpha_2-\alpha_3)(\alpha_2-\alpha_1)} + \frac{\phi_3(\alpha_1+\alpha_2)}{(\alpha_3-\alpha_1)(\alpha_3-\alpha_2)}}{\frac{\phi_1}{(\alpha_1-\alpha_2)(\alpha_1-\alpha_3)} + \frac{\phi_2}{(\alpha_2-\alpha_3)(\alpha_2-\alpha_1)} + \frac{\phi_3}{(\alpha_3-\alpha_1)(\alpha_3-\alpha_2)}}, \tag{2.4.31}$$

it follows from (2.4.30) and (2.4.31) that

$$\alpha^* - \alpha_4 = \frac{1}{2} \frac{R_3'(\alpha^*)}{\frac{\phi_1}{(\alpha_1-\alpha_2)(\alpha_1-\alpha_3)} + \frac{\phi_2}{(\alpha_2-\alpha_3)(\alpha_2-\alpha_1)} + \frac{\phi_3}{(\alpha_3-\alpha_1)(\alpha_3-\alpha_2)}}. \tag{2.4.32}$$

Let $e_i = \alpha^* - \alpha_i, i = 1, 2, 3, 4$. It follows from (2.4.32) that

$$\begin{aligned} & e_4[-\phi_1(e_2-e_3) - \phi_2(e_3-e_1) - \phi_3(e_1-e_2)] \\ = \quad & -\frac{1}{2} R_3'(\alpha^*)(e_1-e_2)(e_2-e_3)(e_3-e_1). \end{aligned} \tag{2.4.33}$$

Noting that $\phi'(\alpha^*) = 0$, it follows from Taylor expansion that

$$\phi_i = \phi(\alpha^*) + \frac{1}{2} e_i^2 \phi''(\alpha^*) + O(e_i^3). \tag{2.4.34}$$

Neglecting the third-order term and substituting (2.4.34) into (2.4.33) give

$$e_4 = \frac{1}{\phi''(\alpha^*)} R_3'(\alpha^*). \tag{2.4.35}$$

Also, by the Lagrange interpolation formula, we have

$$\begin{aligned} R_3'(\alpha) = \quad & \frac{1}{6}\phi'''(\xi(\alpha))[(\alpha-\alpha_2)(\alpha-\alpha_3) + (\alpha-\alpha_1)(\alpha-\alpha_3) \\ & +(\alpha-\alpha_1)(\alpha-\alpha_2)] + \frac{1}{24}\phi^{(4)}(\eta)(\alpha-\alpha_1)(\alpha-\alpha_2)(\alpha-\alpha_3), \end{aligned}$$

which implies

$$R_3'(\alpha^*) = \frac{1}{6}\phi'''(\xi(\alpha^*))(e_1e_2 + e_2e_3 + e_3e_1) + \frac{1}{24}\phi^{(4)}(\eta)e_1e_2e_3. \tag{2.4.36}$$

Neglecting the fourth-order derivative term, it follows from (2.4.35) and (2.4.36) that

$$e_4 = \frac{\phi'''(\xi(\alpha^*))}{6\phi''(\alpha^*)}(e_1e_2 + e_2e_3 + e_3e_1) = M(e_1e_2 + s_2e_3 + e_3e_1),$$

where M is some constant. In general, we have

$$e_{k+2} = M(e_{k-1}e_k + e_ke_{k+1} + e_{k+1}e_{k-1}). \tag{2.4.37}$$

Since $e_{k+1} = O(e_k) = O(e_{k-1})$ when $e_k \to 0$, there exists $\bar{M} > 0$ such that

$$|e_{k+2}| \le \bar{M}|e_{k-1}||e_k|,$$

i.e.,

$$\bar{M}|e_{k+2}| \le \bar{M}|e_{k-1}|\bar{M}|e_k|.$$

When $|e_i|, (i = 1, 2, 3)$ are sufficiently small such that

$$\delta = \max\{\bar{M}|e_1|, \bar{M}|e_2|, \bar{M}|e_3|\} < 1,$$

one has

$$\bar{M}|e_4| \le \bar{M}|e_1|\bar{M}|e_2| \le \delta^2.$$

Set

$$\bar{M}|e_k| \le \delta^{q_k}, \tag{2.4.38}$$

then

$$\bar{M}|e_{k+2}| \le \bar{M}|e_k|\bar{M}|e_{k-1}| \le \delta^{q_k}\delta^{q_{k-1}} \triangleq \delta^{q_{k+2}},$$

hence

$$q_{k+2} = q_k + q_{k-1}, \ (k \ge 2) \tag{2.4.39}$$

where $q_1 = q_2 = q_3 = 1$. Obviously, the characteristic equation of (2.4.39) is

$$t^3 - t - 1 = 0 \tag{2.4.40}$$

with one root $t_1 \approx 1.32$ and other two conjugate complex roots, $|t_2| = |t_3| < 1$. The general solution of (2.4.39) has form

$$q_k = At_1^k + Bt_2^k + Ct_3^k, \tag{2.4.41}$$

where A, B and C are coefficients to be determined. Clearly, when $k \to \infty$,

$$q_{k+1} - t_1q_k = Bt_2^k(t_2 - t_1) + Ct_3^k(t_3 - t_1) \to 0.$$

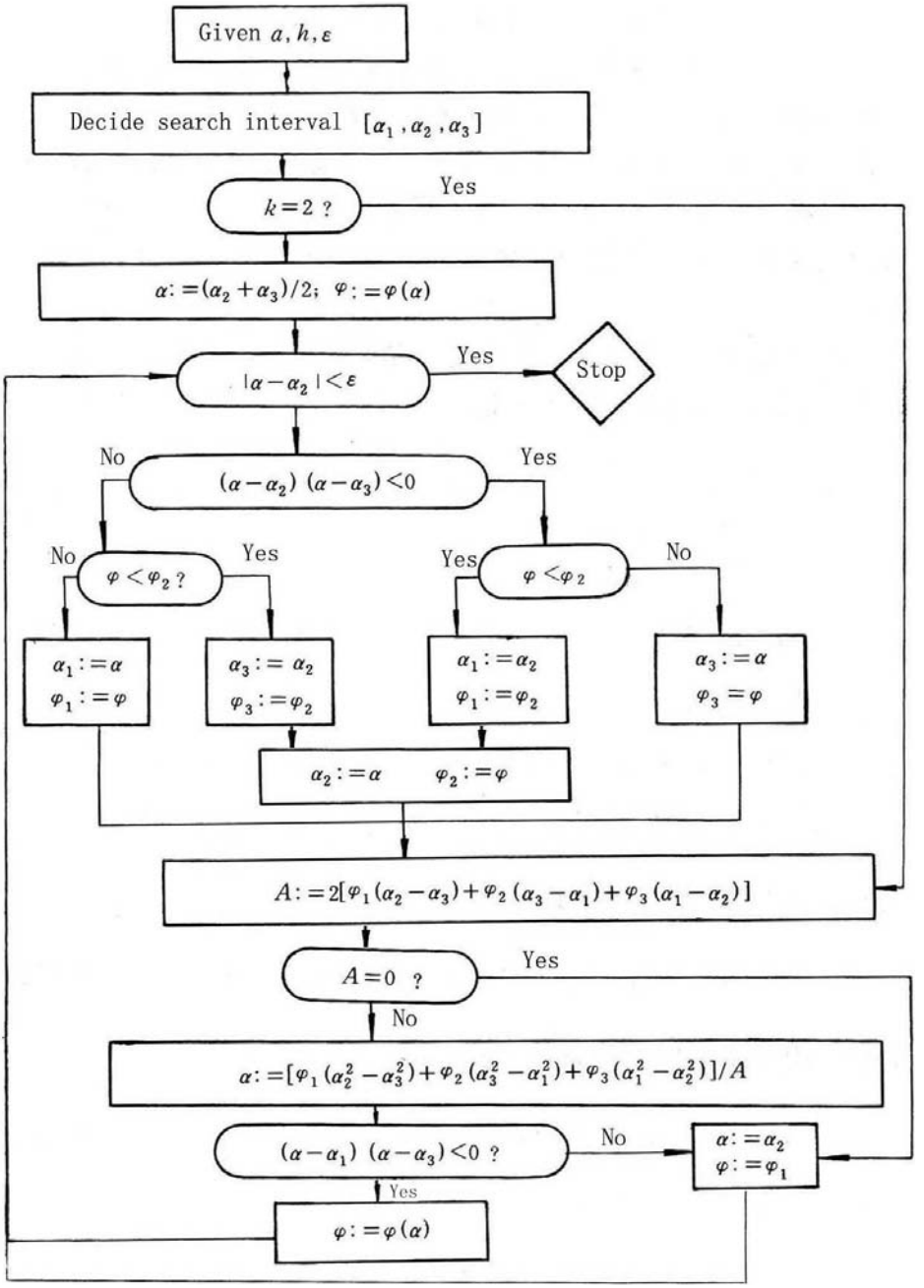


Figure 2.4.1 Flow chart for quadratic interpolation method with three points

So, when k is sufficiently large, we have

$$q_{k+1} - t_1 q_k \geq -0.1. \tag{2.4.42}$$

Note from (2.4.38) that $|e_k| \leq (1/\bar{M})\delta^{q_k} \triangleq B_k$, $(k \geq 1)$. Then, by (2.4.42), when k is sufficiently large,

$$\frac{B_{k+1}}{B_k} = \frac{\delta^{q_{k+1}}/\bar{M}}{\delta^{t_1 q_k}/(\bar{M})^{t_1}} = \bar{M}^{t_1-1}\delta^{q_{k+1}-t_1 q_k} \leq \delta^{-0.1}\bar{M}^{t_1-1},$$

which indicates that the convergence order $t_1 \approx 1.32$. □

2.4.2 Cubic Interpolation Method

The cubic interpolation method approximates the objective function $\phi(\alpha)$ by a cubic polynomial. To construct the cubic polynomial $p(\alpha)$, four interpolation conditions are required. For example, we may use function values at four points, or function values at three points and a derivative value at one point, or function values and derivative values at two points. Note that, in general, the cubic interpolation has better convergence than the quadratic interpolation, but that it needs computing of derivatives and more expensive computation. Hence it is often used for smooth functions. In the following, we discuss the cubic interpolation method with two points.

We are given two points a and b, the function values $\phi(a)$ and $\phi(b)$, and the derivative values $\phi'(a)$ and $\phi'(b)$ to construct a cubic polynomial of the form

$$p(\alpha) = c_1(\alpha - a)^3 + c_2(\alpha - a)^2 + c_3(\alpha - a) + c_4 \tag{2.4.43}$$

where c_i are the coefficients of the polynomial which are chosen such that

$$\begin{aligned}
&p(a) = c_4 = \phi(a),\\
&p'(a) = c_3 = \phi'(a),\\
&p(b) = c_1(b-a)^3 + c_2(b-a)^2 + c_3(b-a) + c_4 = \phi(b),\\
&p'(b) = 3c_1(b-a)^2 + 2c_2(b-a) + c_3 = \phi'(b).
\end{aligned} \tag{2.4.44}$$

From the sufficient condition of the minimizer, we have

$$p'(\alpha) = 3c_1(\alpha - a)^2 + 2c_2(\alpha - a) + c_3 = 0 \tag{2.4.45}$$

and

$$p''(\alpha) = 6c_1(\alpha - a) + 2c_2 > 0. \tag{2.4.46}$$

Solving (2.4.45) yields

$$\alpha = a + \frac{-c_2 \pm \sqrt{c_2^2 - 3c_1c_3}}{3c_1}, \text{ if } c_1 \neq 0, \tag{2.4.47}$$

$$\alpha = a - \frac{c_3}{2c_2}, \text{ if } c_1 = 0. \tag{2.4.48}$$

In order to guarantee the condition (2.4.46) holding, we only take the positive in (2.4.47). So we combine (2.4.47) with (2.4.48), and get

$$\alpha - a = \frac{-c_2 + \sqrt{c_2^2 - 3c_1c_3}}{3c_1} = \frac{-c_3}{c_2 + \sqrt{c_2^2 - 3c_1c_3}}. \tag{2.4.49}$$

When $c_1 = 0$, (2.4.49) is just (2.4.48). Then the minimizer of $p(\alpha)$ is

$$\bar{\alpha} = a - \frac{c_3}{c_2 + \sqrt{c_2^2 - 3c_1c_3}}. \tag{2.4.50}$$

The minimizer in (2.4.50) is represented by c_1, c_2 and c_3. We hope to represent $\bar{\alpha}$ by $\phi(a), \phi(b), \phi'(a)$ and $\phi'(b)$ directly.

Let

$$\begin{aligned} &s = 3\frac{\phi(b) - \phi(a)}{b - a},\ z = s - \phi'(a) - \phi'(b), \\ &w^2 = z^2 - \phi'(a)\phi'(b). \end{aligned} \tag{2.4.51}$$

By use of (2.4.44), we have

$$\begin{aligned} s &= 3\frac{\phi(b) - \phi(a)}{b - a} = 3[c_1(b-a)^2 + c_2(b-a) + c_3], \\ z &= s - \phi'(a) - \phi'(b) = c_2(b-a) + c_3, \\ w^2 &= z^2 - \phi'(a)\phi'(b) = (b-a)^2(c_2^2 - 3c_1c_3). \end{aligned}$$

Then

$$(b-a)c_2 = z - c_3,\ \sqrt{c_2^2 - 3c_1c_3} = \frac{w}{b-a},$$

and so

$$c_2 + \sqrt{c_2^2 - 3c_1c_3} = \frac{z + w - c_3}{b - a}. \tag{2.4.52}$$

Using $c_3 = \phi'(a)$ and substituting (2.4.52) into (2.4.50), we get

$$\bar{\alpha} - a = \frac{-(b-a)\phi'(a)}{z + w - \phi'(a)}, \tag{2.4.53}$$

which is

$$\begin{aligned} \bar{\alpha} - a &= \frac{-(b-a)\phi'(a)\phi'(b)}{(z + w - \phi'(a))\phi'(b)} = \frac{-(b-a)(z^2 - w^2)}{\phi'(b)(z+w) - (z^2 - w^2)} \\ &= \frac{(b-a)(w-z)}{\phi'(b) - z + w}. \end{aligned} \tag{2.4.54}$$

Unfortunately, the formula (2.4.54) is not adequate for calculating $\bar{\alpha}$, because its denominator is possibly zero or merely very small. Fortunately, it can be overcome by use of (2.4.53) and (2.4.54), and we have

$$\begin{aligned} \bar{\alpha} - a &= \frac{-(b-a)\phi'(a)}{z + w - \phi'(a)} = \frac{(b-a)(w-z)}{\phi'(b) - z + w} \\ &= \frac{(b-a)(-\phi'(a) + w - z)}{\phi'(b) - \phi'(a) + 2w} \\ &= (b-a)\left(1 - \frac{\phi'(b) + z + w}{\phi'(b) - \phi'(a) + 2w}\right), \end{aligned} \tag{2.4.55}$$

or

$$\bar{\alpha} = a + (b-a)\frac{w - \phi'(a) - z}{\phi'(b) - \phi'(a) + 2w}. \tag{2.4.56}$$

In (2.4.55) and (2.4.56), the denominator $\phi'(b) - \phi'(a) + 2w \neq 0$. In fact, since $\phi'(a) < 0$ and $\phi'(b) > 0$, then $w^2 = z^2 - \phi'(a)\phi'(b) > 0$. Taking $w > 0$, it follows that the denominator $\phi'(b) - \phi'(a) + 2w > 0$.

In the same way as we did in the last subsection, we can discuss the convergence rate of the cubic interpolation method. Similar to (2.4.16), we can obtain

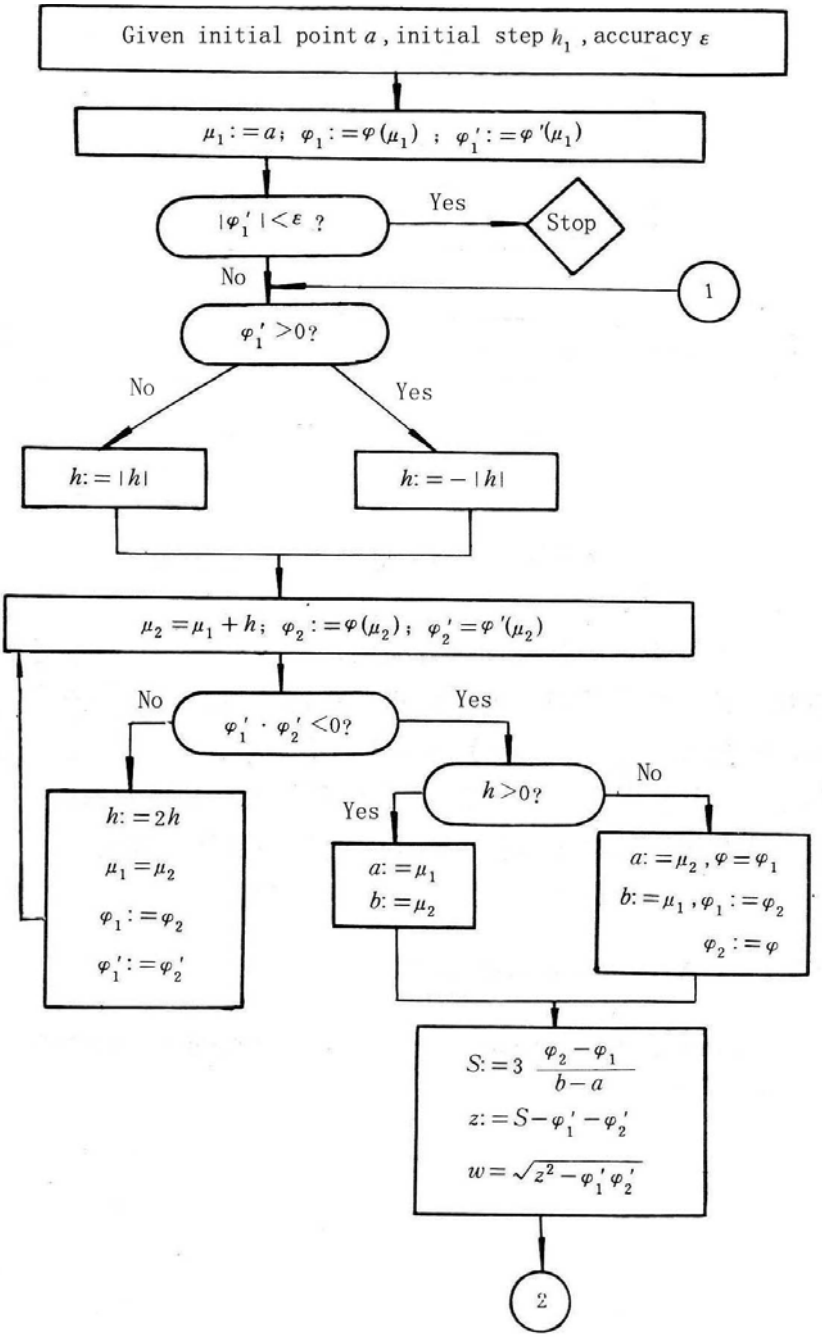
$$e_{k+1} = M(e_k e_{k-1}^2 + e_k^2 e_{k-1}),$$

where M is some constant. We can show that the characteristic equation is

$$t^2 - t - 2 = 0,$$

which solution is $t = 2$. Therefore the cubic interpolation method with two points has convergence rate with order 2.

Finally, we give a flow diagram of the method in Figure 2.4.2.



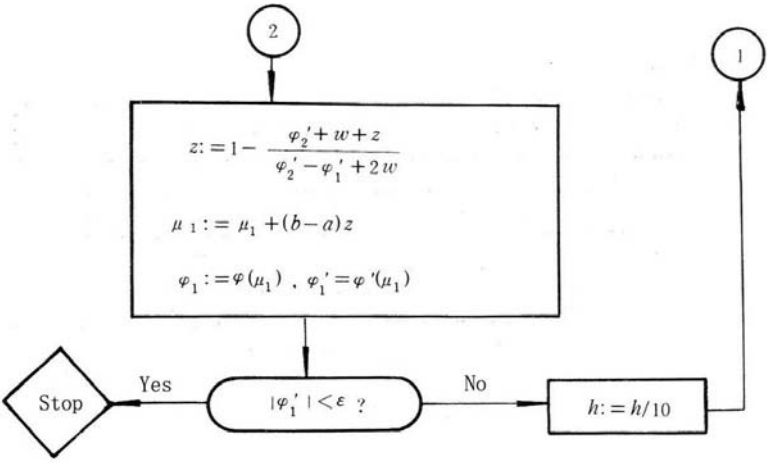


Figure 2.4.2 Flow chart for cubic interpolation method with two points

2.5 Inexact Line Search Techniques

Line search is a basic part of optimization methods. In the last sections we have discussed some exact line search techniques which find α_k such that

$$f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha d_k),$$

or

$$\alpha_k = \min\{\alpha |\ \nabla f(x_k + \alpha d_k)^T d_k = 0,\ \alpha \geq 0\}.$$

However, commonly, the exact line search is expensive. Especially, when an iterate is far from the solution of the problem, it is not effective to solve exactly a one-dimension subproblem. Also, in practice, for many optimization methods, for example, Newton method and quasi-Newton method, their convergence rate does not depend on the exact line search. Therefore, as long as there is an acceptable steplength rule which ensures that the objective function has sufficient descent, the exact line search can be avoided and the computing efforts will be decreased greatly. In the following, we define $g_k = \nabla f(x_k)$ without special indication.

2.5.1 Armijo and Goldstein Rule

Armijo rule [4] is as follows: Given $\beta \in (0,1), \rho \in (0, \frac{1}{2}), \tau > 0$, there exists the least nonnegative integer m_k such that

$$f(x_k) - f(x_k + \beta^m \tau d_k) \geq -\rho \beta^m \tau g_k^T d_k. \tag{2.5.1}$$

Goldstein (1965) [157] presented the following rule. Let

$$J = \{\alpha > 0 \mid f(x_k + \alpha d_k) < f(x_k)\} \tag{2.5.2}$$

be an interval. In Figure 2.5.1 $J = (0, a)$. In order to guarantee the objective function decreases sufficiently, we want to choose α such that it is away from the two end points of the interval J. The two reasonable conditions are

$$f(x_k + \alpha d_k) \leq f(x_k) + \rho \alpha g_k^T d_k \tag{2.5.3}$$

and

$$f(x_k + \alpha d_k) \geq f(x_k) + (1-\rho) \alpha g_k^T d_k, \tag{2.5.4}$$

which exclude those points near the right end-point and the left end-point, where $0 < \rho < \frac{1}{2}$, All α satisfying (2.5.3)-(2.5.4) constitute the interval $J_2 = [b, c]$. We call (2.5.3)-(2.5.4) Goldstein inexact line search rule, in brief, Goldstein rule. When a step-length factor α satisfies (2.5.3)-(2.5.4), it is called an acceptable step-length factor, and the obtained interval $J_2 = [b, c]$ is called an acceptable interval.

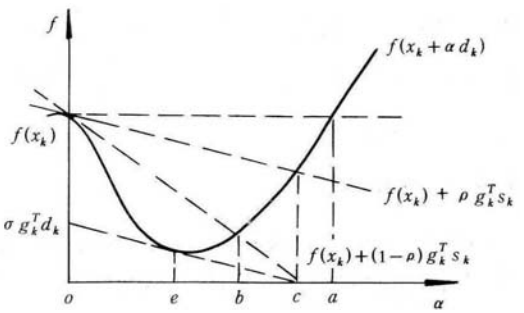


Figure 2.5.1 Inexact line search

As before, let $\phi(\alpha) = f(x_k + \alpha d_k)$. Then (2.5.3) and (2.5.4) can be rewritten respectively

$$\phi(\alpha_k) \leq \phi(0) + \rho \alpha_k \phi'(0), \tag{2.5.5}$$

$$\phi(\alpha_k) \geq \phi(0) + (1-\rho) \alpha_k \phi'(0). \tag{2.5.6}$$

Note that the restriction $\rho < \frac{1}{2}$ is necessary. In fact, if $\phi(\alpha)$ is a quadratic function satisfying $\phi'(0) < 0$ and $\phi''(0) > 0$, then the global minimizer α^* of ϕ satisfies

$$\phi(\alpha^*) = \phi(0) + \frac{1}{2}\alpha^*\phi'(0).$$

Hence α^* satisfies (2.5.5) if and only if $\rho < \frac{1}{2}$. The restriction $\rho < \frac{1}{2}$ will also finally permit $\alpha = 1$ for Newton method and quasi-Newton method. Therefore, without the restriction $\rho < \frac{1}{2}$, the superlinear convergence of the methods will not be guaranteed.

2.5.2 Wolfe-Powell Rule

As shown in Figure 2.5.1, it is possible that the rule (2.5.4) excludes the minimizing value of α outside the acceptable interval. Instead, the Wolfe-Powell rule gives another rule to replace (2.5.4):

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k, \ \sigma \in (\rho, 1), \tag{2.5.7}$$

which implies that

$$\begin{aligned} \phi'(\alpha_k) &= [\nabla f(x_k + \alpha_k d_k)]^T d_k \geq \sigma \nabla f(x_k)^T d_k \\ &= \sigma\phi'(0) > \phi'(0). \end{aligned} \tag{2.5.8}$$

It shows that the geometric interpretation of (2.5.7) is that the slope $\phi'(\alpha_k)$ at the acceptable point must be greater than or equal to some multiple $\sigma \in (0, 1)$ of the initial slope. The rule (2.5.3) and (2.5.7) is called the Wolfe-Powell inexact line search rule, in brief, the Wolfe-Powell rule, which gives the acceptable interval $J_3 = [e, c]$ that includes the minimizing values of α.

In fact, the rule (2.5.7) can be obtained from the mean-value theorem and (2.5.4). Let α_k satisfy (2.5.4). Then

$$\begin{aligned} \alpha_k[\nabla f(x_k + \theta_k\alpha_k d_k)]^T d_k &= f(x_k + \alpha_k d_k) - f(x_k) \\ &\geq (1 - \rho)\alpha_k \nabla f(x_k)^T d_k \end{aligned}$$

which shows (2.5.7). Now we show the existence of α_k satisfying (2.5.3) and (2.5.7). Let $\hat{\alpha}_k$ satisfy the equality in (2.5.3). By the mean-value theorem and (2.5.3), we have

$$\begin{aligned} \hat{\alpha}_k[\nabla f(x_k + \theta_k\hat{\alpha}_k d_k)]^T d_k &= f(x_k + \hat{\alpha}_k d_k) - f(x_k) \\ &= \rho\hat{\alpha}_k \nabla f(x_k)^T d_k, \end{aligned}$$

where $\theta_k \in (0,1)$. Let $\rho < \sigma < 1$, and note that $\nabla f(x_k)^T d_k < 0$, we have

$$[\nabla f(x_k + \theta_k \hat{\alpha}_k d_k)]^T d_k = \rho \nabla f(x_k)^T d_k > \sigma \nabla f(x_k)^T d_k$$

which is just (2.5.7) if we set $\alpha_k = \theta_k \hat{\alpha}_k$. The discussion above also shows that the requirement $\rho < \sigma < 1$ is necessary, such that there exists steplength factor α_k satisfying the Wolfe-Powell rule.

It should point out that the inequality requirement (2.5.7) is an approximation of the orthogonal condition

$$g_{k+1}^T d_k = 0$$

which is satisfied by exact line search. However, unfortunately, one possible disadvantage of (2.5.7) is that it does not reduce to an exact line search in the limit $\sigma \to 0$. In addition, a steplength may satisfy the Wolfe-Powell rule (2.5.3) and (2.5.7) without being close to a minimizer of ϕ. Luckily, if we replace (2.5.7) by using the rule

$$|g_{k+1}^T d_k| \leq -\sigma g_k^T d_k, \tag{2.5.9}$$

the exact line search is obtained in the limit $\sigma \to 0$, and the points that are far from a stationary point of ϕ will be excluded. Therefore the rule (2.5.3) and (2.5.9) is also a successful pair of inexact line search rules which is called the strong Wolfe-Powell rule. Furthermore, we often employ the following form of the strong Wolfe-Powell rule:

$$|g_{k+1}^T d_k| \leq \sigma |g_k^T d_k| \tag{2.5.10}$$

or

$$|\phi'(\alpha_k)| \leq \sigma |\phi'(0)|. \tag{2.5.11}$$

In general, the smaller the value σ, the more exact the line search. Normally, taking $\sigma = 0.1$ gives a fairly accurate line search, whereas the value $\sigma = 0.9$ gives a weak line search. However, taking too small σ may be unwise, because the smaller the value σ, the more expensive the computing effort. Usually, $\rho = 0.1$ and $\sigma = 0.4$ are suitable, and it depends on the specific problem.

2.5.3 Goldstein Algorithm and Wolfe-Powell Algorithm

Although it is possible that the minimizing value of α may be excluded by the rule (2.5.4), it seldom occurs in practice. Therefore, Goldstein rule (2.5.3)-(2.5.4) is a frequently used rule in practice. The overall structure is illustrated in Figure 2.5.2 and the details of the algorithm are described in Algorithm 2.5.1.

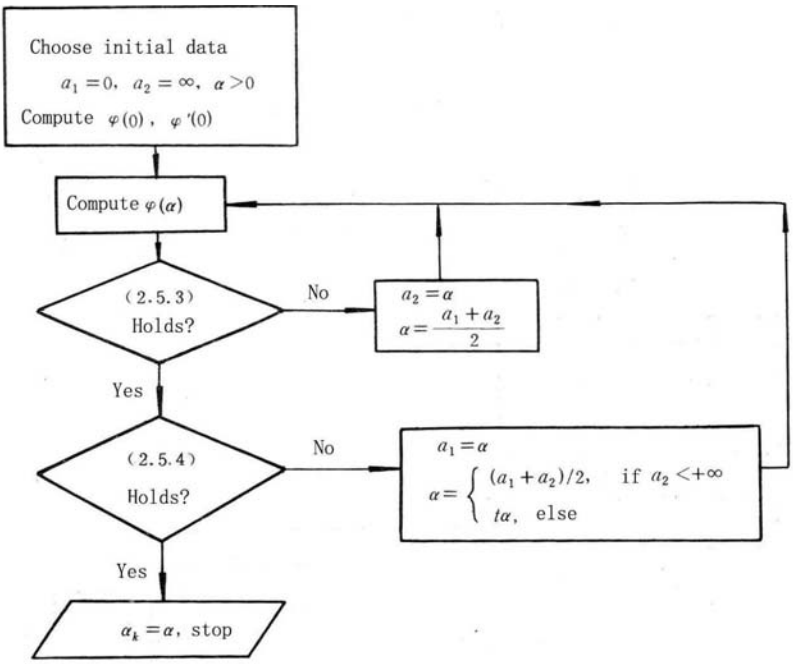


Figure 2.5.2 Flow chart for Goldstein inexact line search

Algorithm 2.5.1 *(Inexact Line Search with Goldstein Rule)*

Step 1. Choose initial data. Take initial point α_0 *in* $[0,+\infty)$ *(or* $[0,\alpha_{max}]$*). Compute* $\phi(0),\phi'(0)$*. Given* $\rho\in(0,\frac{1}{2}), t>1$*. Set* $a_0:=0, b_0:=+\infty$ *(or* α_{max}*),* $k:=0$*.*

Step 2. Check the rule (2.5.3). Compute $\phi(\alpha_k)$*. If*

$$\phi(\alpha_k)\le\phi(0)+\rho\alpha_k\phi'(0),$$

go to Step 3; otherwise, set $a_{k+1} := a_k, b_{k+1} := \alpha_k$, *go to Step 4.*

Step 3. Check the rule (2.5.4). If

$$\phi(\alpha_k) \geq \phi(0) + (1-\rho)\alpha_k\phi'(0),$$

stop, and output α_k*; otherwise, set* $a_{k+1} := \alpha_k, b_{k+1} := b_k$. *If* $b_{k+1} < +\infty$, *go to Step 4; otherwise set* $\alpha_{k+1} := t\alpha_k, k := k+1$, *go to Step 2.*

Step 4. Choose a new point. Set

$$\alpha_{k+1} := \frac{a_{k+1} + b_{k+1}}{2},$$

and $k := k+1$, *go to Step 2.* □

Similarly, we give in Figure 2.5.3 the diagram of the Wolfe-Powell algorithm.

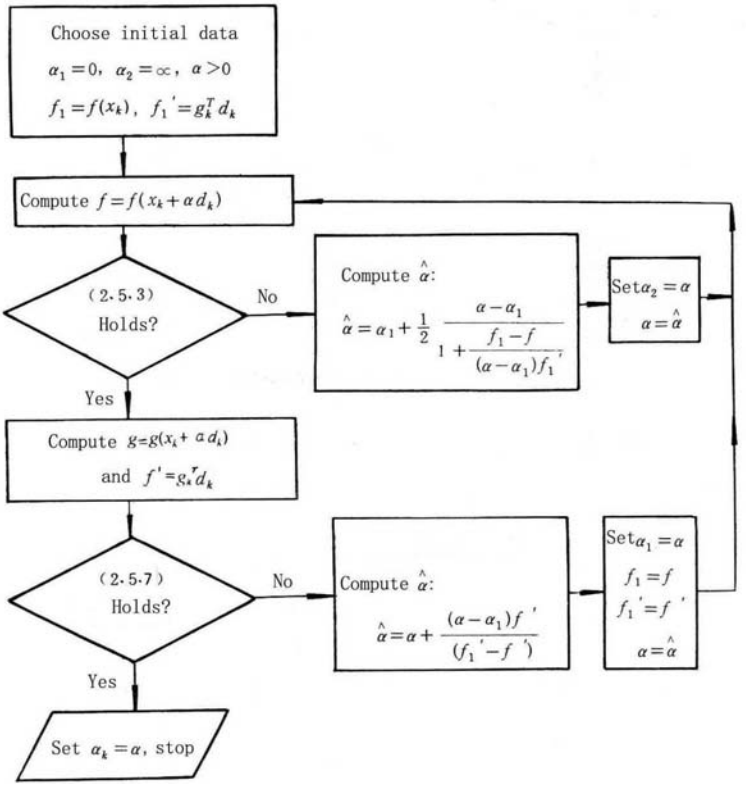


Figure 2.5.3 Flow chart for Wolfe-Powell inexact line search

2.5.4 Backtracking Line Search

In practice, frequently, we also use only the condition (2.5.3) if we choose an appropriate α which is not too small. This method is called backtracking line search. The idea of backtracking is, at the beginning, to set $\alpha = 1$. If $x_k + \alpha d_k$ is not acceptable, we reduce α until $x_k + \alpha d_k$ satisfies (2.5.3).

Algorithm 2.5.2

Step 1. *Given* $\rho \in (0, \frac{1}{2}), 0 < l < u < 1$, *set* $\alpha = 1$.

Step 2. *Test*

$$f(x_k + \alpha d_k) \leq f(x_k) + \rho \alpha g_k^T d_k;$$

Step 3. *If (2.5.3) is not satisfied, set* $\alpha := \omega\alpha, \omega \in [l, u]$, *and go to Step 2; otherwise, set* $\alpha_k := \alpha$ *and* $x_{k+1} := x_k + \alpha_k d_k$. □

In Step 3 of the above algorithm, the quadratic interpolation can be used to reduce α. Let

$$\phi(\alpha) = f(x_k + \alpha d_k). \tag{2.5.12}$$

At the beginning, we have

$$\phi(0) = f(x_k),\ \phi'(0) = \nabla f(x_k)^T d_k. \tag{2.5.13}$$

After computing $f(x_k + d_k)$, we have

$$\phi(1) = f(x_k + d_k). \tag{2.5.14}$$

If $f(x_k + d_k)$ does not satisfy (2.5.3), the following quadratic model can be used to approximate $\phi(\alpha)$:

$$m(\alpha) = [\phi(1) - \phi(0) - \phi'(0)]\alpha^2 + \phi'(0)\alpha + \phi(0), \tag{2.5.15}$$

which obeys the three conditions in (2.5.13)-(2.5.14). Setting $m'(\alpha) = 0$ gives

$$\hat{\alpha} = -\frac{\phi'(0)}{2[\phi(1) - \phi(0) - \phi'(0)]}, \tag{2.5.16}$$

which can be taken as the next value of α.

In order to prevent α from being too small and not terminating, some safeguards are needed. For example, given the least step minstep, if (2.5.3) is not satisfied but $\|\alpha d_k\| <$ minstep, the line search stops.

In summary, in this section we introduced three kind of inexact line search rules:

1. Goldstein rule: (2.5.3)-(2.5.4).
2. Wolfe-Powell rule: (2.5.3) and (2.5.7); Strong Wolfe-Powell rule: (2.5.3) and (2.5.9).
3. Backtracking rule (also called Armijo rule): (2.5.3) or (2.5.1).

The above three inexact line search rules are frequently used in optimization methods below.

2.5.5 Convergence Theorems of Inexact Line Search

In the final subsection we establish convergence theorems of inexact line search methods. To prove the descent property of the methods, we try to avoid the case in which the search directions $s_k = \alpha_k d_k$ are nearly orthogonal to the negative gradient $-g_k$, that is, the angle θ_k between s_k and $-g_k$ is uniformly bounded away from 90^o,

$$\theta_k \leq \frac{\pi}{2} - \mu, \ \forall k \tag{2.5.17}$$

where $\mu > 0, \theta_k \in [0, \frac{\pi}{2}]$ is defined by

$$\cos\theta_k = -g_k^T s_k / (\|g_k\| \|s_k\|), \tag{2.5.18}$$

because, otherwise, $g_k^T s_k$ will approach zero and so s_k is almost not a descent direction.

A general descent algorithm with inexact line search is as follows:

Algorithm 2.5.3

Step 1. Given $x_0 \in R^n, 0 \leq \varepsilon < 1, k := 0$.

Step 2 If $\|\nabla f(x_k)\| \leq \varepsilon$, stop; otherwise, find a descent direction d_k such that $d_k^T \nabla f(x_k) < 0$.

Step 3 Find the steplength factor α_k by use of Goldstein rule (2.5.3)-(2.5.4) or Wolfe-Powell rule (2.5.3) and (2.5.7).

Step 4 Set $x_{k+1} = x_k + \alpha_k d_k; k := k+1$, go to Step 2. □

In Algorithm 2.5.3, d_k is a general descent direction provided it satisfies $d_k^T \nabla f(x_k) < 0$, and α_k is a general inexact line-search factor provided some inexact line search rule is satisfied. So, this algorithm is a very general algorithm, that is, it contains a great class of methods.

Now, we establish the global convergence of the general descent algorithm with inexact line search.

Theorem 2.5.4 *Let α_k in Algorithm 2.5.3 be defined by Goldstein rule (2.5.3)-(2.5.4) or Wolfe-Powell rule (2.5.3) and (2.5.7). Let also s_k satisfy (2.5.17). If ∇f exists and is uniformly continuous on the level set $\{x|\ f(x) \le f(x_0)\}$, then either $\nabla f(x_k) = 0$ for some k, or $f(x_k) \to -\infty$, or $\nabla f(x_k) \to 0$.*

Proof. Let α_k be defined by (2.5.3)-(2.5.4). Assume that, for all k, $g_k = \nabla f(x_k) \neq 0$ (whence $s_k = \alpha_k d_k \neq 0$) and $f(x_k)$ is bounded below, it follows that $f(x_k) - f(x_{k+1}) \to 0$, hence $-g_k^T s_k \to 0$ from (2.5.3).

Now assume that $g_k \to 0$ does not hold. Then there exist $\varepsilon > 0$ and a subsequence such that $\|g_k\| \ge \varepsilon$ and $\|s_k\| \to 0$. Since $\theta_k \le \frac{\pi}{2} - \mu$, we get

$$\cos\theta_k \ge \cos(\frac{\pi}{2} - \mu) = \sin\mu,$$

hence

$$-g_k^T s_k \ge \sin\mu \|g_k\| \|s_k\| \ge \varepsilon \sin\mu \|s_k\|.$$

But the Taylor series gives

$$f(x_{k+1}) = f(x_k) + g(\xi_k)^T s_k,$$

where ξ_k is on the line segment (x_k, x_{k+1}). By uniform continuity, we have $g(\xi_k) \to g_k$ when $s_k \to 0$. So

$$f(x_{k+1}) = f(x_k) + g_k^T s_k + o(\|s_k\|).$$

Therefore we obtain

$$\frac{f(x_k) - f(x_{k+1})}{-g_k^T s_k} \to 1,$$

which contradicts (2.5.4). Hence, $g_k \to 0$, and the proof is complete.

Similarly, instead of (2.5.4), if we use (2.5.7), we can get global convergence of the Wolfe-Powell algorithm. The proof is essentially the same as above. We need only to note that, by uniform continuity of $g(x)$, it follows that

$$g_{k+1}^T s_k = g_k^T s_k + o(\|s_k\|),$$

such that

$$\frac{g_{k+1}^T s_k}{g_k^T s_k} \to 1.$$

This contradicts $g_{k+1}^T s_k / g_k^T s_k \leq \sigma < 1$ given by (2.5.7). Hence $g_k \to 0$. Therefore, the global convergence theorem also holds when α_k is defined by Wolfe-Powell rule (2.5.3) and (2.5.7). □

Next, we give the convergence theorems with the Wolfe-Powell rule.

Theorem 2.5.5 *Let $f : R^n \to R$ be continuously differentiable and bounded below, and let ∇f be uniformly continuous on the level set $\Omega = \{x \mid f(x) \leq f(x_0)\}$. Assume that α_k is defined by Wolfe-Powell rule (2.5.3) and (2.5.7). Then the sequence generated by Algorithm 2.5.3 satisfies*

$$\lim_{k \to +\infty} \frac{\nabla f(x_k)^T s_k}{\|s_k\|} = 0, \tag{2.5.19}$$

which means

$$\|\nabla f(x_k)\| \cos \theta_k \to 0. \tag{2.5.20}$$

Proof. Since $\nabla f(x_k)^T s_k < 0$ and f is bounded below, then the sequence $\{x_k\}$ is well-defined and $\{x_k\} \subset \Omega$. Also, since $\{f(x_k)\}$ is a descent sequence, hence it is convergent.

We now prove (2.5.19) by contradiction. Assume that (2.5.19) does not hold. Then there exist $\varepsilon > 0$ and a subsequence with index set K, such that

$$-\frac{\nabla f(x_k)^T s_k}{\|s_k\|} \geq \varepsilon, \ k \in K.$$

From (2.5.3), one has

$$f(x_k) - f(x_{k+1}) \geq \rho \|s_k\| \left(-\frac{\nabla f(x_k)^T s_k}{\|s_k\|} \right) \geq \rho \|s_k\| \varepsilon.$$

Since also $\{f(x_k)\}$ is a convergent sequence, then $\{s_k : k \in K\}$ converges to zero. Also by (2.5.7), we have

$$(1-\sigma)(-\nabla f(x_k)^T s_k) \le (\nabla f(x_k + s_k) - \nabla f(x_k))^T s_k,\ k \ge 0.$$

Therefore

$$\varepsilon \le -\frac{\nabla f(x_k)^T s_k}{\|s_k\|} \le \frac{1}{1-\sigma}\|\nabla f(x_k + s_k) - \nabla f(x_k)\|,\ k \in K. \tag{2.5.21}$$

However, since we have proved $\{s_k \,|\, k \in K\} \to 0$, then the right-hand side of (2.5.21) goes to zero by the uniform continuity of ∇f on the level set Ω. Hence there is a contradiction which completes the proof. □

Note that (2.5.19) implies

$$\|\nabla f(x_k)\| \cos\theta_k \to 0,$$

which is called the Zoutendijk condition, where θ_k is the angle between $-\nabla f(x_k)$ and s_k. If $\cos\theta_k \ge \delta > 0$, we have $\lim_{k\to\infty} \|\nabla f(x_k)\| = 0$. Also, if the assumption of uniform continuity is replaced by Lipschitz continuity, the theorem is also true. In the theorem below, we prove this case. We first prove a lemma which gives a bound of descent for a single step.

Lemma 2.5.6 *Let $f : D \subset R^n \to R$ be continuously differentiable, also let $\nabla f(x)$ satisfy Lipschitz condition*

$$\|\nabla f(y) - \nabla f(z)\| \le M\|y - z\|,$$

where $M > 0$ is a constant. If $f(x_k + \alpha d_k)$ is bounded below and $\alpha > 0$, then for all $\alpha_k > 0$ satisfying (2.5.3) and (2.5.7), we have

$$f(x_k) - f(x_k + \alpha_k d_k) \ge \beta\|\nabla f(x_k)\|^2 \cos^2\langle d_k, -\nabla f(x_k)\rangle, \tag{2.5.22}$$

where $\beta > 0$ is a constant.

Proof. From Lipschitz condition of ∇f and (2.5.7) we have

$$\alpha_k M\|d_k\|^2 \ge d_k^T[\nabla f(x_k + \alpha_k d_k) - \nabla f(x_k)] \ge -(1-\sigma)d_k^T \nabla f(x_k),$$

that is

$$\begin{aligned} \alpha_k\|d_k\| &\ge \frac{1-\sigma}{M\|d_k\|}\|d_k\|\|\nabla f(x_k)\|\cos\langle d_k, -\nabla f(x_k)\rangle \\ &= \frac{1-\sigma}{M}\|\nabla f(x_k)\|\cos\langle d_k, -\nabla f(x_k)\rangle. \end{aligned}$$

Using (2.5.3) yields

$$\begin{aligned} & f(x_k) - f(x_k + \alpha_k d_k) \geq -\alpha_k \rho d_k^T \nabla f(x_k) \\ = \; & \alpha_k \rho \|d_k\| \|\nabla f(x_k)\| \cos\langle d_k, -\nabla f(x_k)\rangle \\ \geq \; & \rho \|\nabla f(x_k)\| \cos\langle d_k, -\nabla f(x_k)\rangle \frac{1-\sigma}{M} \|\nabla f(x_k)\| \cos\langle d_k, -\nabla f(x_k)\rangle \\ = \; & \frac{\rho(1-\sigma)}{M} \|\nabla f(x_k)\|^2 \cos^2\langle d_k, -\nabla f(x_k)\rangle, \end{aligned}$$

which is (2.5.22) in which $\beta = \rho(1-\sigma)/M$. □

Theorem 2.5.7 *Let $f(x)$ be continuously differentiable on R^n, and let $\nabla f(x)$ satisfy Lipschitz condition*

$$\|\nabla f(x) - \nabla f(y)\| \leq M\|x - y\|. \tag{2.5.23}$$

Also let α_k in Algorithm 2.5.3 be defined by Wolfe-Powell rule (2.5.3) and (2.5.7). If the condition (2.5.17) is satisfied, then, for the sequence $\{x_k\}$ generated by Algorithm 2.5.3, either $\nabla f(x_k) = 0$ for some k, or $f(x_k) \to -\infty$, or $\nabla f(x_k) \to 0$.

Proof. Assume that $\nabla f(x_k) \neq 0, \forall k$. By Lemma 2.5.6, we have

$$f(x_k) - f(x_{k+1}) \geq \beta \cos^2\theta_k \|\nabla f(x_k)\|^2, \tag{2.5.24}$$

where $\beta = \rho(1-\sigma)/M$ is a positive constant being independent of k. Then, for all $k > 0$, we have

$$\begin{aligned} f(x_0) - f(x_k) &= \sum_{i=0}^{k-1} [f(x_i) - f(x_{i+1})] \\ &\geq \beta \min_{0\leq i\leq k} \|\nabla f(x_i)\|^2 \sum_{i=0}^{k-1} \cos^2\theta_i. \end{aligned} \tag{2.5.25}$$

Since θ_k satisfies (2.5.17), this means that

$$\sum_{k=0}^{\infty} \cos^2\theta_k = +\infty. \tag{2.5.26}$$

Then it follows from (2.5.25) that either $\nabla f(x_k) \to 0$ or $f(x_k) \to -\infty$. This completes the proof. □

In fact, Theorem 2.5.7 is a direct result coming from (2.5.20) and the angle condition (2.5.17).

Finally, we derive an estimate of descent amount of $f(x)$ under inexact line search.

Theorem 2.5.8 *Let α_k satisfy (2.5.3). If $f(x)$ is a uniformly convex function, i.e., there exists a constant $\eta > 0$ such that*

$$(y-z)^T[\nabla f(y) - \nabla f(z)] \geq \eta \|y-z\|^2, \tag{2.5.27}$$

or there exist positive constants m and M $(m < M)$, such that

$$m\|y\|^2 \leq y^T \nabla^2 f(x) y \leq M\|y\|^2. \tag{2.5.28}$$

Then

$$f(x_k) - f(x_k + \alpha_k d_k) \geq \frac{\rho\eta}{1+\sqrt{M/m}} \|\alpha_k d_k\|^2, \tag{2.5.29}$$

where ρ is defined in (2.5.3).

Proof. We divide into two cases.

First, assume that $d_k^T \nabla f(x_k + \alpha_k d_k) \leq 0$. In this case we have

$$\begin{aligned} f(x_k) - f(x_k + \alpha_k d_k) &= \int_0^{\alpha_k} -d_k^T \nabla f(x_k + t d_k) \mathrm{d}t \\ &= \int_0^{\alpha_k} d_k^T [\nabla f(x_k + \alpha_k d_k) - \nabla f(x_k + t d_k)] \mathrm{d}t \\ &\geq \int_0^{\alpha_k} \eta(\alpha_k - t) \mathrm{d}t \|d_k\|^2 \\ &= \frac{1}{2}\eta \|\alpha_k d_k\|^2 \\ &\geq \frac{\rho\eta}{1+\sqrt{M/m}} \|\alpha_k d_k\|^2. \end{aligned} \tag{2.5.30}$$

Second, assume that $d_k^T \nabla f(x_k + \alpha_k d_k) > 0$. Then there exists $0 < \alpha^* < \alpha_k$, such that $d_k^T \nabla f(x_k + \alpha^* d_k) = 0$. So, it follows from (2.5.28) that

$$f(x_k) - f(x_k + \alpha^* d_k) \leq \frac{1}{2} M \|\alpha^* d_k\|^2, \tag{2.5.31}$$

and

$$f(x_k + \alpha_k d_k) - f(x_k + \alpha^* d_k) \geq \frac{1}{2} m \|(\alpha_k - \alpha^*) d_k\|^2. \tag{2.5.32}$$

Since $f(x_k + \alpha_k d_k) < f(x_k)$, it follows from (2.5.31) and (2.5.32) that

$$\alpha_k \leq \left(1 + \sqrt{\frac{M}{m}}\right) \alpha^*. \tag{2.5.33}$$

Hence

$$\begin{aligned} f(x_k) - f(x_k + \alpha_k d_k) &\geq -\alpha_k \rho d_k^T \nabla f(x_k) \\ &\geq \alpha_k \rho d_k^T [\nabla f(x_k + \alpha^* d_k) - \nabla f(x_k)] \\ &\geq \eta \rho \alpha_k \alpha^* \|d_k\|^2 \\ &\geq \frac{\rho \eta}{1 + \sqrt{M/m}} \|\alpha_k d_k\|^2. \end{aligned} \tag{2.5.34}$$

Hence (2.5.29) holds in both cases. This completes the proof. □

In this chapter we have discussed exact and inexact line search techniques which guarantee monotonic decrease of the objective function. On the other hand it is found that enforcing monotonicity of the function values may considerably slow the rate of convergence, especially in the presence of narrow curved valleys. Therefore, it is reasonable to present a nonmonotonic line search technique for optimization which allows an increase in function value at each step, while retaining global convergence. Grippo etc. [164] generalized the Armijo rule to the nonmonotone case and relaxed the condition of monotonic decrease. Several papers also deal with these techniques. Here we only state the basic result of nonmonotonic line search as follows.

Theorem 2.5.9 *Let $\{x_k\}$ be a sequence defined by*

$$x_{k+1} = x_k + \alpha_k d_k, \ d_k \neq 0.$$

Let $\tau > 0, \sigma \in (0,1), \gamma \in (0,1)$ and let M be a nonnegative integer. Assume that

(i) the level set $\Omega = \{x \mid f(x) \leq f(x_0)\}$ is compact;

(ii) there exist positive numbers c_1, c_2 such that

$$\nabla f(x_k)^T d_k \leq -c_1 \|\nabla f(x_k)\|^2, \tag{2.5.35}$$

$$\|d_k\| \leq c_2 \|\nabla f(x_k)\|; \tag{2.5.36}$$

(iii) $\alpha_k = \sigma^{h_k} \tau$, where h_k is the first nonnegative integer h, such that

$$f(x_k + \sigma^h \tau d_k) \leq \max_{0 \leq j \leq m(k)} [f(x_{k-j})] + \gamma \sigma^h \tau \nabla f(x_k)^T d_k, \tag{2.5.37}$$

where $m(0) = 0$ *and* $0 \leq m(k) \leq \min[m(k-1)+1, M], k \geq 1$.

Then the sequence $\{x_k\}$ *remains in* Ω *and every accumulation point* $\bar{x}$ *satisfies* $\nabla f(\bar{x}) = 0$.

Proof. See Grippo etc. [164]. □

Exercises

1. Let $f(x) = (\sin x)^6 \tan(1-x)e^{30x}$. Find the maximum of $f(x)$ in $[0, 1]$ by use of the 0.618 method, quadratic interpolation method, and Goldstein line search, respectively.

2. Write the Fibonacci algorithm and its program in MATLAB (or FORTRAN, C).

3. Let $\phi(t) = e^{-t} + e^t$. Let the initial interval be $[-1, 1]$.
(1) Minimize $\phi(t)$ by 0.618 method.
(2) Minimize $\phi(t)$ by Fibonacci method.
(3) Minimize $\phi(t)$ by Armijo line search.

4. Let $\phi(t) = 1 - te^{-t^2}$. Let the initial interval be $[0, 1]$. Try to minimize $\phi(t)$ by quadratic interpolation method.

5. Let $\phi(t) = -2t^3 + 21t^2 - 60t + 50$.
(1) Minimize $\phi(t)$ by Armijo rule if $t_0 = 0.5$ and $\rho = 0.1$.
(2) Minimize $\phi(t)$ by Goldstein rule if $t_0 = 0.5$ and $\rho = 0.1$.
(3) Minimize $\phi(t)$ by Wolfe rule if $t_0 = 0.5, \rho = 0.1$, and $\sigma = 0.8$.

6. Let $f(x) = x_1^4 + x_1^2 + x_2^2$. Given current point $x_k = (1, 1)^T$ and $d_k = (-3, -1)^T$. Let $\rho = 0.1, \sigma = 0.5$.
(1) Try using the Wolfe rule to find a new point x_{k+1}.
(2) Set $\alpha = 1, \alpha = 0.5, \alpha = 0.1$ respectively, describe that for which α satisfies the Wolfe rule and for which α does not satisfy the Wolfe rule.

7. Show that if $0 < \sigma < \rho < 1$, then there may be no steplengths that satisfy the Wolfe rule.

8. Describe the outline of Theorem 2.5.4.

9. Prove the other form of Theorem 2.5.5: Let $f : R^n \to R$ be continuously differentiable and bounded below, and let ∇f be Lipschitz continuous on the level set $\Omega = \{x \mid f(x) \leq f(x_0)\}$. Assume that α_k is defined by Wolfe-Powell rule (2.5.3) and (2.5.7). Then the sequence generated by Algorithm 2.5.3 satisfies

$$\lim_{k \to +\infty} \frac{\nabla f(x_k)^T s_k}{\|s_k\|} = 0,$$

which means

$$\|\nabla f(x_k)\| \cos \theta_k \to 0.$$

Chapter 3

Newton's Methods

3.1 The Steepest Descent Method

3.1.1 The Steepest Descent Method

The steepest descent method is one of the simplest and the most fundamental minimization methods for unconstrained optimization. Since it uses the negative gradient as its descent direction, it is also called the gradient method.

Suppose that $f(x)$ is continuously differentiable near x_k, and the gradient $g_k \stackrel{\text{def}}{=} \nabla f(x_k) \neq 0$. From the Taylor expansion

$$f(x) = f(x_k) + (x - x_k)^T g_k + o(\|x - x_k\|), \tag{3.1.1}$$

we know that, if we write $x - x_k = \alpha d_k$, then the direction d_k satisfying $d_k^T g_k < 0$ is called a descent direction that is such that $f(x) < f(x_k)$. Fixing α, it follows that the smaller the value $d_k^T g_k$ (i.e., the larger the value $|d_k^T g_k|$) is, the faster the function value decreases. By the Cauchy-Schwartz inequality

$$|d_k^T g_k| \leq \|d_k\| \|g_k\|, \tag{3.1.2}$$

we have that the value $d_k^T g_k$ is the smallest if and only if $d_k = -g_k$. Therefore $-g_k$ is the steepest descent direction.

The iterative scheme of the steepest descent method is

$$x_{k+1} = x_k - \alpha_k g_k. \tag{3.1.3}$$

In the following we give the algorithm.

Algorithm 3.1.1 *(The Steepest Descent Method)*

Step 0. *Let* $0 < \varepsilon \ll 1$ *be the termination tolerance. Given an initial point* $x_0 \in R^n$. *Set* $k = 0$.

Step 1. *If* $\|g_k\| \leq \varepsilon$, *stop ; otherwise let* $d_k = -g_k$.

Step 2. *Find the steplength factor* α_k, *such that*

$$f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha d_k);$$

Step 3. *Compute* $x_{k+1} = x_k + \alpha_k d_k$.

Step 4. $k := k + 1$, *return to Step 1.* □

3.1.2 Convergence of the Steepest Descent Method

The steepest descent method is of importance in the area of optimization from the theoretical point of view. The importance of its convergence theory is not only in itself but also in other optimization methods. In the following, we discuss the global convergence and local convergence rate of the steepest descent method.

Theorem 3.1.2 *(Global convergence theorem of the steepest descent method) Let* $f \in C^1$. *Then each accumulation point of the iterative sequence* $\{x_k\}$ *generated by the steepest descent Algorithm 3.1.1 with exact line search is a stationary point.*

Proof. Let $\bar{x}$ be any accumulation point of $\{x_k\}$ and K an infinite index set such that $\lim_{k \in K} x_k = \bar{x}$. Set $d_k = -\nabla f(x_k)$. Since $f \in C^1$, the sequence $\{d_k \mid k \in K\}$ is uniformly bounded and $\|d_k\| = \|\nabla f(x_k)\|$. Since the assumptions of Theorem 2.2.3 are satisfied, it follows that $\|\nabla f(\bar{x})\|^2 = 0$, i.e., $\nabla f(\bar{x}) = 0$. □

Theorem 3.1.3 *(Global convergence theorem of the steepest descent method) Let* $f(x)$ *be twice continuously differentiable in* R^n *and* $\|\nabla^2 f(x)\| \leq M$ *for a positive constant* M. *Given any initial* x_0 *and* $\varepsilon > 0$. *Then the sequence generated from Algorithm 3.1.1 terminates in finitely many iterations, or* $\lim_{k \to \infty} f(x_k) = -\infty$, *or* $\lim_{k \to \infty} \nabla f(x_k) = 0$.

Proof. Consider the infinite case. From Algorithm 3.1.1 and Theorem 2.2.2, we have

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2M}\|\nabla f(x_k)\|^2.$$

Then

$$f(x_0) - f(x_k) = \sum_{i=0}^{k-1}[f(x_i) - f(x_{i+1})] \geq \frac{1}{2M}\sum_{i=0}^{k-1}\|\nabla f(x_i)\|^2.$$

Taking limits yields either $\lim_{k\to\infty} f(x_k) = -\infty$ or $\lim_{k\to\infty}\|\nabla f(x_k)\| = 0$. The result then follows. □

Instead of the exact line search in Step 2 of Algorithm 3.1.1, the steepest descent method can also use inexact line search technique. For this case, the global convergence is given below.

Theorem 3.1.4 *(Convergence theorem of the steepest descent method with inexact line search)*

Let $f \in C^1$. Consider the steepest descent method with inexact line search. Then each accumulation point of the sequence $\{x_k\}$ is a stationary point.

Proof. It follows directly from Theorem 2.5.4. □

Unfortunately, the global convergence does not guarantee that the steepest descent method is an effective method. We can clearly find this problem from the following analysis and the local convergence rate theorem.

In fact, the steepest descent direction is only a local property of the algorithm. For many problems, the steepest descent method is not the actual "steepest", but is very slow. Although the method usually works well in the early steps, as a stationary point is approached, it descends very slowly with zigzagging phenomena. This zigzagging phenomena is illustrated in Figure 3.1.1 for the problem

$$\min(x_1 - 2)^4 + (x_1 - 2x_2)^2,$$

in which zigzagging occurs along the valley shown by the dotted lines.

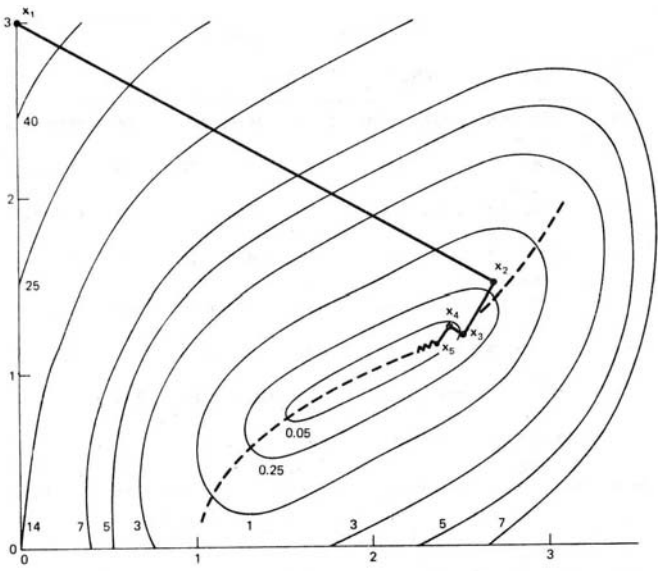


Figure 3.1.1 Zigzagging in the steepest descent method

In fact, the zigzagging of the steepest descent method can be explained by the following facts. Since, from exact line search, one has

$$g_{k+1}^T d_k = 0,$$

then

$$g_{k+1}^T g_k = d_{k+1}^T d_k = 0. \tag{3.1.4}$$

This shows that two gradients are orthogonal to each other on the successive iterates, and thus two successive directions are also orthogonal, which leads to the zigzagging. When the stationary point is approached, $\|g_k\|$ will be very small. By means of the expression

$$f(x_k + \alpha d) = f(x_k) + \alpha g_k^T d + o(\|\alpha d\|), \tag{3.1.5}$$

it is easy to see that the first order term $\alpha g_k^T d = -\alpha \|g_k\|^2$ is of a very small order of magnitude. Hence the descent of f is very small.

Next, we discuss the convergence rate of the steepest descent method, first for the case of a quadratic function and then for the case of a general function.

When the objective function is quadratic, the convergence rate of the steepest descent method depends on the ratio of the longest axis and the

shortest axis of the ellipsoid which corresponds to the contour of the objective function. The bigger the ratio is, the slower the descent is. The following theorem indicates this fact and says that the steepest descent method converges linearly.

Theorem 3.1.5 *(The convergence rate theorem of the steepest descent method for the case of a quadratic function)*

Consider the unconstrained minimization problem

$$\min_{x\in R^n} f(x) = \frac{1}{2}x^T Gx, \tag{3.1.6}$$

where G is an $n \times n$ symmetric and positive definite matrix. Let λ_1 and λ_n be the largest and the smallest eigenvalues of G respectively. Let x^* be the solution of the problem (3.1.6). Then the sequence $\{x_k\}$ generated by the steepest descent method converges to x^*, the convergence rate is at least linear, and the following bounds hold:

$$\frac{f(x_{k+1}) - f(x^*)}{f(x_k) - f(x^*)} \leq \frac{(\kappa-1)^2}{(\kappa+1)^2} = \frac{(\lambda_1-\lambda_n)^2}{(\lambda_1+\lambda_n)^2}, \tag{3.1.7}$$

$$\frac{\|x_{k+1} - x^*\|_G}{\|x_k - x^*\|_G} \leq \frac{\kappa-1}{\kappa+1} = \left(\frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n}\right), \tag{3.1.8}$$

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \leq \sqrt{\kappa}\frac{\kappa-1}{\kappa+1} = \sqrt{\frac{\lambda_1}{\lambda_n}}\left(\frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n}\right), \tag{3.1.9}$$

where $\kappa = \lambda_1/\lambda_n$.

Proof. Consider the minimization of (3.1.6); we have

$$x_{k+1} = x_k - \alpha_k g_k, \tag{3.1.10}$$

with

$$\alpha_k = \frac{g_k^T g_k}{g_k^T G g_k} \tag{3.1.11}$$

and $g_k = Gx_k$.

$$\begin{aligned}\frac{f(x_k) - f(x_{k+1})}{f(x_k)} &= \frac{\frac{1}{2}x_k^T Gx_k - \frac{1}{2}(x_k - \alpha_k g_k)^T G(x_k - \alpha_k g_k)}{\frac{1}{2}x_k^T Gx_k} \\ &= \frac{\alpha_k g_k^T Gx_k - \frac{1}{2}\alpha_k^2 g_k^T Gg_k}{\frac{1}{2}x_k^T Gx_k}\end{aligned}$$

$$= \frac{\frac{(g_k^T g_k)^2}{g_k^T G g_k} - \frac{1}{2}\frac{(g_k^T g_k)^2}{g_k^T G g_k}}{\frac{1}{2} g_k^T G^{-1} g_k}$$

$$= \frac{(g_k^T g_k)^2}{(g_k^T G g_k)(g_k^T G^{-1} g_k)}. \tag{3.1.12}$$

By using Kantorovich inequality (3.1.33), we have immediately that

$$\frac{f(x_{k+1})}{f(x_k)} = \left[1 - \frac{(g_k^T g_k)^2}{(g_k^T G g_k)(g_k^T G^{-1} g_k)}\right] \tag{3.1.13}$$

$$\leq \left[1 - \frac{4\lambda_1 \lambda_n}{(\lambda_1 + \lambda_n)^2}\right] = \left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}\right)^2, \tag{3.1.14}$$

which is just (3.1.7).

By using (3.1.13), it is not difficult to get (3.1.8) and (3.1.9). In fact, let $e_k = x_k - x^*$, $\forall k \geq 0$. Noting that G is symmetric and positive definite, we have

$$\lambda_n e_k^T e_k \leq e_k^T G e_k \leq \lambda_1 e_k^T e_k. \tag{3.1.15}$$

Since $x^* = 0$, we have

$$\|x_k - x^*\|_G^2 = e_k^T G e_k = x_k^T G x_k = 2f(x_k). \tag{3.1.16}$$

So, it follows from (3.1.15) that

$$\lambda_n \|x_k - x^*\|^2 \leq 2f(x_k) \leq \lambda_1 \|x_k - x^*\|^2, \ \forall k \geq 0. \tag{3.1.17}$$

From (3.1.13), (3.1.16) and (3.1.17), we get

$$\frac{\lambda_n \|x_{k+1} - x^*\|^2}{\lambda_1 \|x_k - x^*\|^2} \leq \frac{\|x_{k+1} - x^*\|_G^2}{\|x_k - x^*\|_G^2} \leq \left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}\right)^2, \tag{3.1.18}$$

which gives (3.1.8) and (3.1.9). □

If we consider, more generally, the objective function with the form

$$f(x) = \frac{1}{2} x^T G x - b^T x, \tag{3.1.19}$$

where G is an $n \times n$ symmetric positive definite matrix and $b \in R^n$, the above theorem is also true.

If the objective function is generalized to the non-quadratic case, we also can get the linear convergence rate of the steepest descent method.

Theorem 3.1.6 *Let $f(x)$ satisfy the assumptions of Theorem 2.2.8. If the sequence $\{x_k\}$ generated from the steepest descent method converges to x^*, then the convergence rate is at least linear.*

Proof. It is a direct result from Theorem 2.2.8. □

The above convergence rate theorem of the steepest descent method for a general function can also be described as follows.

Theorem 3.1.7 *Let $f(x)$ be twice continuously differentiable near x^* with $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ positive definite. Let the sequence $\{x_k\}$ generated by the steepest descent method converge to x^*. Let*

$$\frac{f(x_{k+1}) - f(x^*)}{f(x_k) - f(x^*)} = \beta_k. \tag{3.1.20}$$

Then $\beta_k < 1, \forall k$ and

$$\limsup_{k \to +\infty} \beta_k \leq \frac{M - m}{M} < 1, \tag{3.1.21}$$

where M and m satisfy

$$0 < m \leq \lambda_n \leq \lambda_1 \leq M, \tag{3.1.22}$$

and λ_n and λ_1 are respectively the smallest and the largest eigenvalues of $\nabla^2 f(x)$.

Proof. From Theorem 2.2.2, we have

$$\begin{aligned} [f(x_k) - f(x^*)] - [f(x_{k+1}) - f(x^*)] &= f(x_k) - f(x_{k+1}) \\ &\geq \frac{1}{2M} \|\nabla f(x_k)\|^2, \end{aligned} \tag{3.1.23}$$

which is, by the definition of β_k, that

$$(1 - \beta_k)[f(x_k) - f(x^*)] \geq \frac{1}{2M} \|\nabla f(x_k)\|^2.$$

Hence, by the assumption of f, we get

$$\beta_k \leq 1 - \frac{\|\nabla f(x_k)\|^2}{2M[f(x_k) - f(x^*)]} < 1. \tag{3.1.24}$$

Now suppose that $(x_k - x^*)/\|x_k - x^*\| \to \bar{d}$. It is obvious that

$$\|\nabla f(x_k)\|^2 = \|x_k - x^*\|^2(\|\nabla^2 f(x^*)\bar{d}\|^2 + o(1))$$

and

$$f(x_k) - f(x^*) = \frac{1}{2}\|x_k - x^*\|^2(\bar{d}^T\nabla^2 f(x^*)\bar{d} + o(1)).$$

Using the above equalities and (3.1.22) yields

$$\lim_{k\to\infty} \frac{\|\nabla f(x_k)\|^2}{f(x_k) - f(x^*)} = \frac{2\|\nabla^2 f(x^*)\bar{d}\|^2}{\bar{d}^T\nabla^2 f(x^*)\bar{d}} \geq 2m. \tag{3.1.25}$$

Hence, it follows from (3.1.24) and (3.1.25) that

$$\begin{aligned} \limsup_{k\to\infty} \beta_k &\leq 1 - \liminf_{k\to\infty} \frac{\|\nabla f(x_k)\|^2}{2M[f(x_k) - f(x^*)]} \\ &\leq 1 - \frac{m}{M} < 1. \end{aligned}$$

We complete the proof. □

3.1.3 Barzilai and Borwein Gradient Method

From the above discussions we know that the classical steepest descent method performs poorly, converges linearly, and is badly affected by ill-conditioning.

Barzilai and Borwein [8] presented a two-point step size gradient method, which is called usually the Barzilai-Borwein (or BB) gradient method. In the method, the step size is derived from a two-point approximation to the secant equation underlying quasi-Newton methods (see Chapter 5).

Consider the gradient iteration form

$$x_{k+1} = x_k - \alpha_k g_k \tag{3.1.26}$$

which can be written as

$$x_{k+1} = x_k - D_k g_k, \tag{3.1.27}$$

where $D_k = \alpha_k I$. In order to make the matrix D_k have quasi-Newton property, we compute α_k such that

$$\min \quad \|s_{k-1} - D_k y_{k-1}\|. \tag{3.1.28}$$

This yields that

$$\alpha_k = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}, \tag{3.1.29}$$

where $s_{k-1} = x_k - x_{k-1}, y_{k-1} = g_k - g_{k-1}$.

By symmetry, we may minimize $\|D_k^{-1} s_{k-1} - y_{k-1}\|$ with respect to α_k and get

$$\alpha_k = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}. \tag{3.1.30}$$

The above description produces the following algorithm.

Algorithm 3.1.8 *(The Barzilai-Borwein gradient method)*

Step 0. *Given $x_0 \in R^n, 0 < \varepsilon \ll 1$. Set $k = 0$.*

Step 1. *If $\|g_k\| \leq \varepsilon$, stop ; otherwise let $d_k = -g_k$.*

Step 2. *If $k = 0$, find α_0 by line search; otherwise compute α_k by (3.1.29) or (3.1.30).*

Step 3. *Set $x_{k+1} = x_k + \alpha_k d_k$.*

Step 4. *$k := k + 1$, return to Step 1.* □

It is easy to see that in this method no matrix computations and no line searches (except $k = 0$) are required. The Barzilai-Borwein method is, in fact, a gradient method, but requires less computational work, and greatly speeds up the convergence of the gradient method. Barzilai and Borwein [8] proved that the above algorithm is R-superlinearly convergent for the quadratic case.

In the general non-quadratic case, a globalization strategy based on non-monotone line search is suitable to Barzilai-Borwein gradient method. In addition, in general non-quadratic case, α_k computed by (3.1.29) or (3.1.30) can be unacceptably large or small. Therefore, we must assume that α_k satisfies the condition

$$0 < \alpha^{(l)} \leq \alpha_k \leq \alpha^{(u)}, \quad \text{for all } k,$$

where $\alpha^{(l)}$ and $\alpha^{(u)}$ are previously determined numbers.

If we employ the iteration

$$x_{k+1} = x_k - \frac{1}{\alpha_k} g_k = x_k - \lambda_k g_k \tag{3.1.31}$$

with

$$\alpha_k = \frac{s_{k-1}^T y_{k-1}}{s_{k-1}^T s_{k-1}}, \quad \lambda_k = \frac{1}{\alpha_k}, \tag{3.1.32}$$

note that $s_k = -\frac{1}{\alpha_k} g_k = -\lambda_k g_k$, then we have

$$\alpha_{k+1} = \frac{s_k^T y_k}{s_k^T s_k} = \frac{-\lambda_k g_k^T y_k}{\lambda_k^2 g_k^T g_k} = -\frac{g_k^T y_k}{\lambda_k g_k^T g_k}.$$

Now we give the following Barzilai-Borwein gradient algorithm with nonmonotone globalization.

Algorithm 3.1.9 *(The Barzilai-Borwein gradient algorithm with nonmonotone linesearch)*

Step 0. *Given $x_0 \in R^n, 0 < \varepsilon \ll 1$, an integer $M \geq 0$, $\rho \in (0,1), \delta > 0, 0 < \sigma_1 < \sigma_2 < 1$, $\alpha^{(l)}, \alpha^{(u)}$. Set $k = 0$.*

Step 1. *If $\|g_k\| \leq \varepsilon$, stop.*

Step 2. *If $\alpha_k \leq \alpha^{(l)}$ or $\alpha_k \geq \alpha^{(u)}$ then set $\alpha_k = \delta$.*

Step 3. *Set $\lambda = 1/\alpha_k$.*

Step 4. *(nonmonotone line search) If*

$$f(x_k - \lambda g_k) \leq \max_{0 \leq j \leq \min(k,M)} f(x_{k-j}) - \rho \lambda g_k^T g_k,$$

then set

$$\lambda_k = \lambda, \quad x_{k+1} = x_k - \lambda_k g_k,$$

and go to Step 6.

Step 5. *Choose $\sigma \in [\sigma_1, \sigma_2]$, set $\lambda = \sigma\lambda$, and go to Step 4.*

Step 6. *Set $\alpha_{k+1} = -(g_k^T y_k)/(\lambda_k g_k^T g_k)$, $k := k+1$, return to Step 1.* □

Obviously, the above algorithm is globally convergent.

3.1.4 Appendix: Kantorovich Inequality

We conclude this section with a famous Kantorovich Inequality which is used in the proof of Theorem 3.1.5.

Theorem 3.1.10 *(Kantorovich Inequality) Let G be an $n \times n$ symmetric positive definite matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$. Then, for any $x \in R^n$, the following inequality holds:*

$$\frac{(x^T x)^2}{(x^T G x)(x^T G^{-1} x)} \geq \frac{4\lambda_1 \lambda_n}{(\lambda_1 + \lambda_n)^2}. \tag{3.1.33}$$

Proof. Let the spectral decomposition of G be

$$G = U \Lambda U.$$

Set $x = Uy$, then

$$\begin{aligned} \frac{(x^T x)^2}{(x^T G x)(x^T G^{-1} x)} &= \frac{(y^T y)^2}{(y^T \Lambda y)(y^T \Lambda^{-1} y)} \\ &= \frac{(\sum_{i=1}^n y_i^2)^2}{(\sum_{i=1}^n \lambda_i y_i^2)(\sum_{i=1}^n y_i^2/\lambda_i)}. \end{aligned} \tag{3.1.34}$$

Let

$$\xi_i = \frac{y_i^2}{\sum_{i=1}^n y_i^2},\ \phi(\lambda) = \frac{1}{\lambda}, \tag{3.1.35}$$

then (3.1.34) becomes

$$\frac{(x^T x)^2}{(x^T G x)(x^T G^{-1} x)} = \frac{1}{(\sum_{i=1}^n \lambda_i \xi_i)(\sum_{i=1}^n \phi(\lambda_i)\xi_i)}. \tag{3.1.36}$$

Below we use the convexity of ϕ to estimate the lower bound of the right-hand side of (3.1.36). Let

$$\lambda = \sum_{i=1}^n \lambda_i \xi_i,\ \lambda_\phi = \sum_{i=1}^n \phi(\lambda_i)\xi_i. \tag{3.1.37}$$

Since $\xi_i \geq 0$ $(i = 1, \cdots, n)$ and $\sum_{i=1}^n \xi_i = 1$, we have $\lambda_n \leq \lambda \leq \lambda_1$. Then each λ_i can be represented as a convex combination of λ_1 and λ_n:

$$\lambda_i = \frac{\lambda_1 - \lambda_i}{\lambda_1 - \lambda_n}\lambda_n + \frac{\lambda_i - \lambda_n}{\lambda_1 - \lambda_n}\lambda_1.$$

From the convexity of ϕ, we have obviously

$$\phi(\lambda_i) \leq \frac{\lambda_1 - \lambda_i}{\lambda_1 - \lambda_n}\phi(\lambda_n) + \frac{\lambda_i - \lambda_n}{\lambda_1 - \lambda_n}\phi(\lambda_1). \tag{3.1.38}$$

Then, it follows from (3.1.37), (3.1.38) and (3.1.35) that

$$\begin{aligned} \lambda_\phi &\leq \sum_{i=1}^{n} \left[\frac{\lambda_1 - \lambda_i}{\lambda_1 - \lambda_n}\phi(\lambda_n) + \frac{\lambda_i - \lambda_n}{\lambda_1 - \lambda_n}\phi(\lambda_1)\right] \xi_i \\ &= \sum_{i=1}^{n} \frac{\lambda_1 + \lambda_n - \lambda_i}{\lambda_1 \lambda_n}\xi_i \\ &= \frac{\lambda_1 + \lambda_n - \lambda}{\lambda_1 \lambda_n}. \end{aligned} \tag{3.1.39}$$

Therefore, by (3.1.36), (3.1.37) and (3.1.39) we obtain

$$\begin{aligned} \frac{(x^T x)^2}{(x^T G x)(x^T G^{-1} x)} &= \frac{1}{\lambda \lambda_\phi} \geq \frac{\lambda_1 \lambda_n}{\lambda(\lambda_1 + \lambda_n - \lambda)} \\ &\geq \frac{\lambda_1 \lambda_n}{\max_{\lambda \in [\lambda_n, \lambda_1]} \lambda(\lambda_1 + \lambda_n - \lambda)} = \frac{4\lambda_1 \lambda_n}{(\lambda_1 + \lambda_n)^2}, \end{aligned}$$

which is our result. □

3.2 Newton's Method

The basic idea of Newton's method for unconstrained optimization is to iteratively use the quadratic approximation $q^{(k)}$ to the objective function f at the current iterate x_k and to minimize the approximation $q^{(k)}$.

Let $f : R^n \to R$ be twice continuously differentiable, $x_k \in R^n$, and the Hessian $\nabla^2 f(x_k)$ positive definite. We model f at the current point x_k by the quadratic approximation $q^{(k)}$,

$$f(x_k + s) \approx q^{(k)}(s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T \nabla^2 f(x_k) s, \tag{3.2.1}$$

where $s = x - x_k$. Minimizing $q^{(k)}(s)$ yields

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) \tag{3.2.2}$$

which is Newton's formula. Set

$$G_k = \nabla^2 f(x_k),\ g_k = \nabla f(x_k). \tag{3.2.3}$$

Then we write (3.2.2) as

$$x_{k+1} = x_k - G_k^{-1} g_k, \tag{3.2.4}$$

where $s_k = x_{k+1} - x_k = -G_k^{-1} g_k$ is a Newton's direction. Clearly, the Newton's direction is a descent direction because it satisfies $g_k^T s_k = -g_k^T G_k^{-1} g_k < 0$ if G_k is positive definite. Please note, in the remainder of this book, the first and the second derivatives of f will be denoted by

$$g(x) \triangleq \nabla f(x),\ G(x) \triangleq \nabla^2 f(x) \tag{3.2.5}$$

for convenience, if they exist.

The corresponding algorithm is stated as follows.

Algorithm 3.2.1 *(Newton's Method)*

Step 1. Given $x_0 \in R^n, \epsilon > 0, k := 0$;

Step 2. If $\|g_k\| \leq \epsilon$, *stop;*

Step 3. Solve $G_k s = -g_k$ *for* s_k;

Step 4. Set $x_{k+1} = x_k + s_k$;

Step 5. $k := k + 1$, *go to Step 2.* □

Obviously, Newton's method can be regarded as a steepest descent method under the ellipsoid norm $\|\cdot\|_{G_k}$. In fact, for $f(x_k + s) \approx f(x_k) + g_k^T s$, we regard s_k as the solution of the minimization problem

$$\min_{s \in R^n} \frac{g_k^T s}{\|s\|}. \tag{3.2.6}$$

The solution of (3.2.6) depends on the norm. If we employ l_2 norm, then we get $s_k = -g_k$ and the resultant method is the steepest descent method. If we employ the ellipsoid norm $\|\cdot\|_{G_k}$, then we get $s_k = -G_k^{-1} g_k$ which is just the Newton's method. In fact, in this case, (3.2.6) is equivalent to

$$\begin{aligned} \min_{s \in R^n} \quad & g_k^T s \\ \text{s.t.} \quad & \|s\|_{G_k} \leq 1. \end{aligned}$$

Note that, by (1.2.36), we have that

$$(g_k^T s)^2 \leq (g_k^T G_k^{-1} g_k)(s^T G_k s)$$

and that $g_k^T s$ will be the smallest when $s = -G_k^{-1} g_k$. The above discussion gives us a clear explanation.

For the positive definite quadratic function, Newton's method can reach the minimizer with one iteration. However, for a general non-quadratic function, it is not sure that Newton's method can reach the minimizer with finite iterations. Fortunately, since the objective function is approximate to a quadratic function near the minimizer, then if the starting point is close to the minimizer the Newton's method will converge rapidly. The following theorem shows the local convergence and the quadratic convergence rate of Newton's method.

Theorem 3.2.2 *(Convergence Theorem of Newton's Method) Let $f \in C^2$ and x_k be close enough to the solution x^* of the minimization problem with $g(x^*) = 0$. If the Hessian $G(x^*)$ is positive definite and $G(x)$ satisfies Lipschitz condition*

$$|G_{ij}(x) - G_{ij}(y)| \leq \beta \|x - y\|, \text{ for some } \beta, \text{ for all } i, j \tag{3.2.7}$$

where $G_{ij}(x)$ is the (i, j)-element of $G(x)$, then for all k, Newton's iteration (3.2.4) is well-defined; the generated sequence $\{x_k\}$ converges to x^* with a quadratic rate.

Proof. Let $h_k = x_k - x^*$. From Taylor's formula, it follows that

$$0 = g(x^*) = g_k - G_k h_k + O(\|h_k\|^2).$$

Since $f \in C^2$, x_k is close enough to x^*, and $G(x^*)$ is positive definite, it is reasonable to assume that x_k is in the neighborhood of x^*, G_k positive definite, G_k^{-1} upper bounded. Hence the k-th Newton's iteration exists. Multiplying through by G_k^{-1} yields

$$\begin{aligned} 0 &= G_k^{-1} g_k - h_k + O(\|h_k\|^2) \\ &= -s_k - h_k + O(\|h_k\|^2) \\ &= -h_{k+1} + O(\|h_k\|^2). \end{aligned}$$

By definition of $O(\cdot)$, there is a constant C such that

$$\|h_{k+1}\| \leq C \|h_k\|^2. \tag{3.2.8}$$

If $x_k \in \Omega = \{x \mid \|h\| \leq \gamma/C, h = x - x^*, \gamma \in (0,1)\}$, then

$$\|h_{k+1}\| \leq \gamma\|h_k\| \leq \gamma^2/C < \gamma/C. \tag{3.2.9}$$

Hence $x_{k+1} \in \Omega$. By induction on k, Newton's iteration is well-defined for all k, and $\|h_k\| \to 0$ as $k \to \infty$. Therefore the iteration converges. Also, (3.2.8) shows that the convergence rate of the iteration sequence is quadratic. □

Note that Newton's method is a local method. When the starting point is far away from the solution, it is not sure that G_k is positive definite and Newton's direction d_k is a descent direction. Hence the convergence is not guaranteed. Since, as we know, the line search is a global strategy, we can employ Newton's method with line search to guarantee the global convergence. However it should be noted that only when the step size sequence $\{\alpha_k\}$ converges to 1, Newton's method is convergent with the quadratic rate. Newton's iteration with line search is as follows:

$$d_k = -G_k^{-1} g_k, \tag{3.2.10}$$

$$x_{k+1} = x_k + \alpha_k d_k, \tag{3.2.11}$$

where α_k is a step size. The formula (3.2.10)–(3.2.11) corresponds to the following algorithm.

Algorithm 3.2.3 *(Newton's Method with Line Search)*

Step 1. Initial step: given $x_0 \in R^n, \epsilon > 0$, set $k := 0$.

Step 2. Compute g_k. If $\|g_k\| \leq \epsilon$, stop and output x_k; otherwise go to Step 3.

Step 3. Solve $G_k d = -g_k$ for d_k.

Step 4. Line search step: find α_k such that

$$f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha d_k).$$

Step 5. Set $x_{k+1} = x_k + \alpha_k d_k$, $k := k + 1$, go to Step 2. □

Next, we prove the above Algorithm 3.2.3 is globally convergent.

Theorem 3.2.4 *Let $f : R^n \to R$ be twice continuously differentiable on open convex set $D \subset R^n$. Assume that for any $x_0 \in D$ there exists a constant $m > 0$ such that $f(x)$ satisfies*

$$u^T \nabla^2 f(x) u \geq m\|u\|^2, \ \forall u \in R^n, x \in L(x_0), \tag{3.2.12}$$

where $L(x_0) = \{x \mid f(x) \leq f(x_0)\}$ is the corresponding level set. Then the sequence $\{x_k\}$ generated by Algorithm 3.2.3 satisfies

1. *when $\{x_k\}$ is a finite sequence, $g_k = 0$ for some k;*
2. *when $\{x_k\}$ is an infinite sequence, $\{x_k\}$ converges to the unique minimizer x^* of f.*

Proof. First, from (3.2.12), we know that $f(x)$ is a strictly convex function on R^n, and hence its stationary point is the unique global minimizer.

Also, from the assumption, it follows that the level set $L(x_0)$ is a bounded closed convex set. Since $\{f(x_k)\}$ is monotonic descent, then $\{x_k\} \subset L(x_0)$ and $\{x_k\}$ is bounded. Therefore there exists a limit point $\bar{x} \in L(x_0)$ with $x_k \to \bar{x}$, and further $f(x_k) \to f(\bar{x})$. Also since $f \in C^2(D)$, by Theorem 2.2.4, we have $g_k \to g(\bar{x}) = 0$. Finally, note that the stationary point is unique, then the whole sequence $\{x_k\}$ converges to $\bar{x}$ which is the unique minimizer. □

Similarly, if we employ inexact line search rule (2.5.3) and (2.5.7), it follows from (2.5.22) that

$$f(x_k) - f(x_k + \alpha_k d_k) \geq \bar{\eta}\|g_k\|^2 \cos^2\langle d_k, -g_k\rangle, \tag{3.2.13}$$

where $\bar{\eta}$ is some constant independent of k. In this case the global convergence still holds.

Theorem 3.2.5 *Let $f : R^n \to R$ be twice continuously differentiable on an open convex set $D \subset R^n$. Assume that for any $x_0 \in R^n$, there exists $m > 0$ such that $f(x)$ satisfies (3.2.12) on the level set $L(x_0)$. If the line search employed satisfies (3.2.13), then the sequence $\{x_k\}$ generated from Newton's algorithm satisfies*

$$\lim_{k \to \infty} \|g_k\| = 0, \tag{3.2.14}$$

and $\{x_k\}$ converges to the unique minimizer of $f(x)$.

Proof. Since $f(x)$ satisfies (3.2.12), we see that $f(x)$ is uniformly convex on $L(x_0)$. Also, from (3.2.13), it follows that $f(x)$ is strictly monotonically descending and further that $\{x_k\}$ is bounded. Therefore there exists a constant $M > 0$ such that

$$\|G_k\| \leq M \ \forall k. \tag{3.2.15}$$

From (3.2.10), (3.2.12) and (3.2.15), it follows that

$$\begin{aligned} \cos\langle d_k, -g_k\rangle &= \frac{-d_k^T g_k}{\|d_k\|\|g_k\|} = \frac{g_k^T G_k^{-1} g_k}{\|G_k^{-1} g_k\|\|g_k\|} \\ &= \frac{d_k^T G_k d_k}{\|d_k\|\|G_k d_k\|} \geq \frac{m}{M}. \end{aligned} \tag{3.2.16}$$

Hence, by (3.2.13) and (3.2.16), we have

$$\infty > \sum_{k=0}^{\infty}[f(x_k) - f(x_{k+1})] \geq \sum_{k=0}^{\infty} \bar{\eta}\frac{m^2}{M^2}\|g_k\|^2, \tag{3.2.17}$$

which shows (3.2.14). Note that $f(x)$ is uniformly convex, then $f(x)$ has only one stationary point, and (3.2.14) indicates that $\{x_k\}$ converges to the unique minimizer x^* of f. □

3.3 Modified Newton's Method

The main difficulty faced by Newton's method is that the Hessian G_k is not positive definite. In this case, it is not sure that the model function has minimizers. When G_k is indefinite, the model function will be unbounded.

To overcome these difficulties, there are several modified schemes.

Goldstein-Price Method

Goldstein and Price [159] presented a modified method: when G_k is not positive definite, the steepest descent direction $-g_k$ is used. If we combine this strategy with the angle rule

$$\theta \leq \frac{\pi}{2} - \mu, \text{ for some } \mu > 0,$$

where θ is the angle between $-g_k$ and d_k, we can determine the direction d_k as follows:

$$d_k = \begin{cases} -G_k^{-1} g_k, & \text{if } \cos\theta \geq \eta, \\ -g_k, & \text{otherwise,} \end{cases} \tag{3.3.1}$$

where $\eta > 0$ is a given constant. Then the consultant direction d_k satisfies $\cos\theta \geq \eta$ and the angle rule is satisfied, and thus the corresponding algorithm is convergent.

Goldfeld et al. Method

Goldfeld et al. [156] presented another modified Newton's method. Their method does not substitute the steepest descent method for Newton's method, but makes the Newton's direction $-G_k^{-1}g_k$ turn to the steepest descent direction $-g_k$. More precisely, when G_k is not positive definite, one changes the model Hessian G_k to $G_k + \nu_k I$, where $\nu_k > 0$ such that $G_k + \nu_k I$ is positive definite and well-conditioned. Ideally, ν_k is not much larger than the smallest ν that makes $G_k + \nu I$ positive definite and well-conditioned. The framework of the algorithm is as follows.

Algorithm 3.3.1 *(Modified Newton's Method)*

Initial step: Given an initial point $x_0 \in R^n$.

k-th step:

(1) Set $\bar{G}_k = G_k + \nu_k I$, where

$\nu_k = 0$, *if G_k is positive definite;*

$\nu_k > 0$, *otherwise.*

(2) Solve $\bar{G}_k d = -g_k$ for d_k.

(3) Set $x_{k+1} = x_k + d_k$. □

In the above algorithm, the smallest possible ν_k is slightly larger than the magnitude of the most negative eigenvalue of G_k. We suggest applying the Gill-Murray's modified Cholesky factorization to G_k to determine ν_k, which results in

$$G_k + E = LDL^T, \tag{3.3.2}$$

where E is a diagonal matrix with nonnegative diagonal elements (see Gill, Murray and Wright [152]). If $E = 0$, set $\nu_k = 0$; if $E \neq 0$, we can use the Gerschgorin Circle Theorem 1.2.14 to compute an upper bound b_1 of ν_k:

$$b_1 = \left| \min_{1\leq i\leq n} \left\{ (G_k)_{ii} - \sum_{j\neq i} |(G_k)_{ij}| \right\} \right| \geq \left| \min_i \lambda_i \right|. \tag{3.3.3}$$

In addition, note that

$$b_2 = \max_i\{e_{ii}\} \tag{3.3.4}$$

is also an upper bound of ν_k, where e_{ii} is the i-th diagonal element of E. Then we set

$$\nu_k = \min\{b_1, b_2\}, \tag{3.3.5}$$

and get the positive definite matrix $\bar{G}_k$ and its Cholesky factorization.

In the remainder of this section, we would like to introduce another numerically stable modified Cholesky factorization due to Gill and Murray [149].

It is well-known that the Cholesky factorization $G_k = LDL^T$ of a positive definite matrix G_k can be described as follows:

$$d_{jj} = g_{jj} - \sum_{s=1}^{j-1} d_{ss} l_{js}^2, \tag{3.3.6}$$

$$l_{ij} = \frac{1}{d_{jj}} \left(g_{ij} - \sum_{s=1}^{j-1} d_{ss} l_{js} l_{is} \right), \ i \geq j+1, \tag{3.3.7}$$

where g_{ij} denote the elements of G_k, d_{jj} the diagonal elements of D. Now we ask the Cholesky factors L and D to satisfy the following two requirements: one is that all the diagonal elements of D are positive; the other is that the elements of the factors are uniformly bounded. That is,

$$d_{kk} > \delta > 0, \ \forall k \text{ and } |r_{ik}| \leq \beta, \ i > k, \tag{3.3.8}$$

where $r_{ik} = l_{ik}\sqrt{d_{kk}}$, β is a given positive number and δ is a small positive number.

Below we will describe the j-th step of this factorization. Suppose that the first $j-1$ columns of the factors have been computed, that is, for $k = 1, \cdots, j-1$, d_{kk} and l_{ik} $(i = 1, \cdots, n)$ have been computed and satisfy (3.3.8). Now we compute

$$\gamma_j = |\xi_j - \sum_{s=1}^{j-1} d_{ss} l_{js}^2|, \tag{3.3.9}$$

where ξ_j takes g_{jj} and the test value $\bar{d}$ takes

$$\bar{d} = \max\{\gamma_j, \delta\}. \tag{3.3.10}$$

In order to judge whether to accept $\bar{d}$ as the j-th element of D, we check if $r_{ij} = l_{ij}\sqrt{\bar{d}}$ satisfies (3.3.8). If yes, set $d_{jj} = \bar{d}$ and form the j-th column of L by use of $l_{ij} = r_{ij}/\sqrt{d_{jj}}$; otherwise, set

$$d_{jj} = \left| \xi_j - \sum_{s=1}^{j-1} d_{ss} l_{js}^2 \right|, \tag{3.3.11}$$

where we take $\xi_j = g_{jj} + e_{jj}$ in which e_{jj} is chosen such that $\max|r_{ij}| = \beta$, and also form the j-th column of L as above.

When the above procedure is complete, we obtain a Cholesky factorization of $\bar{G}_k$,

$$\bar{G}_k = LDL^T = G_k + E, \tag{3.3.12}$$

where E is a diagonal matrix with nonnegative diagonal elements e_{jj}. For given G_k, the nonnegative diagonal matrix E depends on the given β. Gill and Murray (1974) prove that if $n > 1$, then

$$\|E(\beta)\|_\infty \leq \left(\frac{\xi}{\beta} + (n-1)\beta \right)^2 + 2(\gamma + (n-1)\beta^2) + \delta, \tag{3.3.13}$$

where ξ and γ are respectively the maximum modules of non-diagonal elements and diagonal elements of G_k. Since, when $\beta^2 = \xi/\sqrt{n^2-1}$, the above bound is minimized, then we take β satisfying

$$\beta^2 = \max\{\gamma, \xi/\sqrt{n^2-1}, \epsilon_M\} \tag{3.3.14}$$

where ϵ_M denotes the machine precision. Also, note that adding the term ϵ_M in (3.3.14) is to prevent the case in which $\|G_k\|$ is too small.

Now we are in a position to state the modified Cholesky factorization algorithm in which $c_{is} = l_{is}d_{ss}(s = 1, \cdots, j; i = j, \cdots, n)$ are auxiliary variables saved in G_k and we need not increase the storage.

Algorithm 3.3.2 *(Modified Cholesky Factorization due to Gill and Murray (1974))*

Step 1. Compute β by (3.3.14). Given δ. Set $j := 1, c_{ii} = g_{ii}$ for $i = 1, \cdots, n$.

Step 2. Find the smallest index q such that $|c_{qq}| = \max_{j \leq i \leq n} |c_{ii}|$, exchange the q-th and the i-th rows, the q-th and the i-th columns.

Step 3. Compute the j-th row of L and find the maximum module of $l_{ij}d_{jj}$.

Set $l_{js} = c_{js}/d_{ss}, s = 1, \cdots, j-1$;
Compute $c_{ij} = g_{ij} - \sum_{s=1}^{j-1} l_{js}c_{is}, i = j+1, \cdots, n$;
Set $\theta_j = \max_{j+1\le i\le n} |c_{ij}|$ (if $j = n, \theta_j = 0$).

Step 4. Compute the j-th diagonal element of D:

$$d_{jj} = \max\{\delta, |c_{jj}|, \theta_j^2/\beta^2\};$$

Update the element e_{jj}: $e_{jj} = d_{jj} - c_{jj}$. If $j = n$, stop.

Step 5. Update $c_{ii} = c_{ii} - c_{ij}^2/d_{jj}, i = j+1, \cdots, n$;
Set $j := j+1$, go to Step 2. □

The modified Cholesky factorization above needs about $\frac{1}{6}n^3$ arithmetic operations which are almost the same as the normal Cholesky factorization.

Example 3.3.3 *Consider*

$$G_k = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 1+10^{-20} & 3 \\ 2 & 3 & 1 \end{pmatrix}. \tag{3.3.15}$$

By the above Algorithm 3.3.2, we can get $\beta^2 = 1.061$,

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0.2652 & 1 & 0 \\ 0.5303 & 0.4295 & 1 \end{pmatrix}, \ D = \begin{pmatrix} 3.771 & 0 & 0 \\ 0 & 5.750 & 0 \\ 0 & 0 & 1.121 \end{pmatrix},$$

$$E = \begin{pmatrix} 2.771 & 0 & 0 \\ 0 & 5.016 & 0 \\ 0 & 0 & 2.243 \end{pmatrix}.$$

The difference $\|\bar{G}_k - G_k\|_F = \|E\|_F \approx 6.154$. Since $d_{jj} \ge \delta$ in the modified factorization, it is guaranteed that $\bar{G}_k = G_k + E_k$ is positive definite and the condition number is uniformly bounded, i.e.,

$$\|\bar{G}_k\|\|\bar{G}_k^{-1}\| \le \kappa, \ \kappa \ge 0.$$

So, we have

$$-\frac{\nabla f(x_k)^T s_k}{\|s_k\|} \ge \frac{1}{\kappa}\|\nabla f(x_k)\|. \tag{3.3.16}$$

Thus, it follows from the inexact line search, (2.5.19) and (3.3.16) that $\{\nabla f(x_k)\}$ converges to zero.

Theorem 3.3.4 *Let $f : D \subset R^n \to R$ be twice continuously differentiable on an open set D. Let the level set $\Omega = \{x \mid f(x) \leq f(x_0)\}$ be compact. If the sequence $\{x_k\}$ is generated by the modified Newton's method, then*

$$\lim_{k\to\infty} \nabla f(x_k) = 0. \tag{3.3.17}$$

3.4 Finite-Difference Newton's Method

The finite-difference Newton's method is to use the finite-difference as an approximation of derivatives in Newton's method.

We first review the finite-difference derivative approximations.

Let $F : R^n \to R^m$. The (i, j)-component of the Jacobian $J(x)$ of $F(x)$ can be approximated by

$$a_{i,j} = \frac{f_i(x + he_j) - f_i(x)}{h}, \tag{3.4.1}$$

where $f_i(x)$ denotes the i-th component of $F(x)$, e_j the j-th unit vector, h a small perturbation of x. Equivalently, if $A_{\cdot j}$ denotes the j-th column of A, we have

$$A_{\cdot j} = \frac{F(x + he_j) - F(x)}{h}. \tag{3.4.2}$$

Theorem 3.4.1 *Let $F : R^n \to R^m$ satisfy the conditions of Theorem 1.2.22. Let the norm $\|\cdot\|$ satisfy $\|e_j\| = 1, j = 1, \cdots, n$. Then*

$$\|A_{\cdot j} - J(x)_{\cdot j}\| \leq \frac{\gamma}{2}|h|. \tag{3.4.3}$$

If the norm used is l_1 norm, then

$$\|A - J(x)\|_1 \leq \frac{\gamma}{2}|h|. \tag{3.4.4}$$

Proof. By setting $d = he_j$ in (1.2.109), we obtain

$$\|F(x + he_j) - F(x) - J(x)he_j\| \leq \frac{\gamma}{2}\|he_j\|^2 = \frac{\gamma}{2}|h|^2.$$

Dividing by h gives (3.4.3). Noting from (1.2.7) that the l_1 norm of a matrix is the maximum of the l_1 norm of a vector, we immediately get (3.4.4). □

Now, let $f : R^n \to R$. An approximation to the gradient $\nabla f(x)$ can be obtained by the forward-difference approximation, defined as

$$\frac{\partial f}{\partial x_i}(x) \approx \frac{f(x + he_i) - f(x)}{h}. \tag{3.4.5}$$

This process requires evaluation of f at $n + 1$ points: x and $x + he_i, i = 1, \cdots, n$. Obviously, it follows from (1.2.109) that

$$\frac{\partial f}{\partial x_i}(x) = \frac{f(x + he_i) - f(x)}{h} + \delta_h, \tag{3.4.6}$$

where

$$|\delta_h| \leq \frac{\gamma}{2} h. \tag{3.4.7}$$

It means there is $O(h)$ error in the forward-difference formula.

A more accurate approximation to the derivative can be obtained by using the central-difference formula, defined as

$$\frac{\partial f}{\partial x_i}(x) \approx \frac{f(x + he_i) - f(x - he_i)}{2h}.$$

The two theorems below give respectively approximations to the gradient and the Hessian of f, and describe the error bounds of these approximations.

Theorem 3.4.2 *Let $f : D \subset R^n \to R$ satisfy the conditions of Theorem 1.2.23. Let the norm used satisfy $\|e_i\| = 1, i = 1, \cdots, n$. Assume that $x + he_i, x - he_i \in D, i = 1, \cdots, n$. Also let the vector $a \in R^n$ with components a_i, be defined as*

$$a_i = \frac{f(x + he_i) - f(x - he_i)}{2h}. \tag{3.4.8}$$

Then

$$|a_i - |\nabla f(x)|_i| \leq \frac{\gamma}{6} h^2. \tag{3.4.9}$$

If the norm used is the l_∞ norm, then

$$\|a - \nabla f(x)\|_\infty \leq \frac{\gamma}{6} h^2. \tag{3.4.10}$$

Proof. Define α and β respectively as

$$\alpha = f(x + he_i) - f(x) - h[\nabla f(x)]_i - \frac{1}{2} h^2 [\nabla^2 f(x)]_{ii} \tag{3.4.11}$$

and

$$\beta = f(x - he_i) - f(x) + h[\nabla f(x)]_i - \frac{1}{2}h^2[\nabla^2 f(x)]_{ii}. \tag{3.4.12}$$

By using (1.2.110) and setting $d = \pm he_i$, we have

$$|\alpha| \leq \frac{\gamma}{6}h^3, \ |\beta| \leq \frac{\gamma}{6}h^3.$$

Then using the triangle inequality gives

$$|\alpha - \beta| \leq \frac{\gamma}{3}h^3.$$

Also, from (3.4.11)-(3.4.12) and (3.4.8), we get

$$\alpha - \beta = 2h(a_i - [\nabla f(x)]_i),$$

which gives (3.4.9). Finally, by using the definition of l_∞ norm, we get (3.4.10) immediately from (3.4.9). □

Theorem 3.4.3 *Let f satisfy the conditions of Theorem 3.4.2. Assume that $x, x + he_i, x + he_j, x + he_i + he_j \in D, 1 \leq i, j \leq n$. Also let $A \in R^{n\times n}$ with components a_{ij} defined as*

$$a_{ij} = \frac{f(x + he_i + he_j) - f(x + he_i) - f(x + he_j) + f(x)}{h^2}. \tag{3.4.13}$$

Then

$$|a_{ij} - [\nabla^2 f(x)]_{ij}| \leq \frac{5}{3}\gamma h. \tag{3.4.14}$$

If the matrix norm is l_1, l_∞, or Frobenius norm, then

$$\|A - \nabla^2 f(x)\| \leq \frac{5}{3}\gamma hn. \tag{3.4.15}$$

Proof. The proof is similar to the proof in Theorem 3.4.2. Set

$$\begin{aligned}
\alpha &= f(x + he_i + he_j) - f(x) - (he_i + he_j)^T\nabla f(x) \\
&\quad -\frac{1}{2}(he_i + he_j)^T\nabla^2 f(x)(he_i + he_j), \\
\beta &= f(x + he_i) - f(x) - (he_i)^T\nabla f(x) - \frac{1}{2}(he_i)^T\nabla^2 f(x)(he_i), \\
\eta &= f(x + he_j) - f(x) - (he_j)^T\nabla f(x) - \frac{1}{2}(he_j)^T\nabla^2 f(x)(he_j),
\end{aligned}$$

respectively. Then

$$\alpha - \beta - \eta = h^2(a_{ij} - [\nabla^2 f(x)]_{ij}). \tag{3.4.16}$$

Also, we have

$$\begin{aligned} |\alpha - \beta - \eta| &\leq |\alpha| + |\beta| + |\eta| \\ &\leq \frac{\gamma}{6}\|he_i + he_j\|^3 + \frac{\gamma}{6}\|he_i\|^3 + \frac{\gamma}{6}\|he_j\|^3 \\ &\leq \frac{5}{3}\gamma h^3. \end{aligned}$$

This inequality together with (3.4.16) gives the result (3.4.14). The inequality (3.4.15) is a consequence of (3.4.14) and definitions of norms. □

Now we are in a position to discuss the finite-difference Newton's method for nonlinear equations

$$F(x) = 0, \tag{3.4.17}$$

where $F : R^n \to R^n$ is continuously differentiable.

The Newton's method for (3.4.17) is as follows:
Solve $J(x_k)d = -F(x_k)$ for d_k;
Set $x_{k+1} = x_k + \alpha_k d_k$;
where $J(x_k)$ is the Jacobian matrix of F at x_k. When $J(x)$ is not available, we can use finite-difference derivative approximation and get the following finite-difference Newton's method for (3.4.17):

$$(A_k)_{.j} = \frac{F(x_k + h_k e_j) - F(x_k)}{h_k},\ j = 1, \cdots, n, \tag{3.4.18}$$

$$x_{k+1} := x_k - A_k^{-1}F(x_k),\ k = 0, 1, \cdots. \tag{3.4.19}$$

Theorem 3.4.4 *Let $F : R^n \to R^n$ be continuously differentiable on an open convex set $D \subset R^n$. Assume there exist $x^* \in R^n$ and $r, \beta > 0$, so that $N(x^*, r) \subset D, F(x^*) = 0, J(x^*)^{-1}$ exists and satisfies $\|J(x^*)^{-1}\| \leq \beta$, where J is Lipschitz continuous in the neighborhood $N(x^*, r) = \{x \in R^n | \ \|x - x^*\| < r\}$. Then there exist $\epsilon, h > 0$, such that if $x_0 \in N(x^*, \epsilon)$ and $\{h_k\}$ is a real sequence with $0 < |h_k| \leq h$, then the sequence $\{x_k\}$ generated from (3.4.18)-(3.4.19) is well-defined and converges to x^* linearly. If*

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

the convergence is superlinear. Furthermore, if there exists a constant c_1, such that

$$|h_k| \leq c_1 \|x_k - x^*\|, \tag{3.4.20}$$

or equivalently, there exists a constant c_2, such that

$$|h_k| \leq c_2 \|F(x_k)\|, \tag{3.4.21}$$

then the convergence rate is quadratic.

Proof. Choose ϵ and h such that, for $x_k \in N(x^*, \epsilon)$, A_k is nonsingular and $|h_k| < h$. Let $\epsilon \leq r$ and

$$\epsilon + h \leq \frac{1}{2\beta\gamma}. \tag{3.4.22}$$

Now we prove, by induction, that

$$\|x_{k+1} - x^*\| \leq \frac{1}{2}\|x_k - x^*\|, \tag{3.4.23}$$

so

$$x_{k+1} \in N(x^*, \epsilon). \tag{3.4.24}$$

For $k = 0$, we first prove A_0 is nonsingular. By assumptions and Theorem 3.4.1, we have $\|A(x) - J(x)\| \leq \frac{\gamma h}{2}$, and then

$$\begin{aligned} & \|J(x^*)^{-1}[A_0 - J(x^*)]\| \\ \leq\ & \|J(x^*)^{-1}\|\|[A_0 - J(x_0)] + [J(x_0) - J(x^*)]\| \\ \leq\ & \beta\left(\frac{\gamma h}{2} + \gamma\epsilon\right) \leq \frac{1}{2}. \end{aligned} \tag{3.4.25}$$

From Von-Neumann Theorem 1.2.5 we know that A_0 is nonsingular and that

$$\|A_0^{-1}\| \leq 2\beta. \tag{3.4.26}$$

Hence x_1 is well-defined and

$$\begin{aligned} x_1 - x^* &= -A_0^{-1}F(x_0) + x_0 - x^* \\ &= A_0^{-1}\{[F(x^*) - F(x_0) - J(x_0)(x^* - x_0)] \\ &\quad +[(J(x_0) - A_0)(x^* - x_0)]\}. \end{aligned} \tag{3.4.27}$$

Then from (3.4.26), (1.2.109) and (3.4.22), we get

$$\begin{aligned}
\|x_1 - x^*\| &\leq \|A_0^{-1}\|\{\|F(x^*) - F(x_0) - J(x_0)(x^* - x_0)\| \\
&\quad + \|A_0 - J(x_0)\|\|x^* - x_0\|\} \qquad (3.4.28)\\
&\leq 2\beta\left\{\frac{\gamma}{2}\|x^* - x_0\|^2 + \frac{\gamma}{2}h\|x_0 - x^*\|\right\} \qquad (3.4.29)\\
&\leq \beta\gamma(\epsilon + h)\|x^* - x_0\| \\
&\leq \frac{1}{2}\|x_0 - x^*\|. \qquad (3.4.30)
\end{aligned}$$

Assume that the conclusion holds for $k = j$, in the same way as $k = 0$, we can prove that the conclusion is also true for $k = j + 1$. Therefore, (3.4.23)-(3.4.24) hold. They also show the linear convergence of the iterative sequence.

The key for superlinear and quadratic convergence requires an improved bound on $\|A_0 - J(x_0)\|$. When $\lim_{k\to\infty} h_k = 0$, the second term in the bracket of (3.4.29) approaches zero, and hence

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \to 0, \text{ when } k \to \infty,$$

which implies that the method converges superlinearly. Similarly, when (3.4.20) is satisfied, it follows from (3.4.29) that the method converges to x^* quadratically. Finally, the equivalence of (3.4.20) and (3.4.21) is just a consequence of Theorem 1.2.25. □

For unconstrained optimization problem

$$\min_{x\in R^n} f(x), \qquad (3.4.31)$$

when the gradient $\nabla f(x)$ is available, we can obtain the Hessian approximation by using the forward-difference or central-difference of the gradient. In this case, the iteration scheme for the k-th step is as follows:

$$\begin{aligned}
(A)_{\cdot j} &= \frac{\nabla f(x_k + h_j e_j) - \nabla f(x_k)}{h_j}, j = 1, \cdots, n, \qquad (3.4.32)\\
A_k &= \frac{A + A^T}{2}, \qquad (3.4.33)\\
x_{k+1} &= x_k - A_k^{-1}\nabla f(x_k), \qquad (3.4.34)
\end{aligned}$$

where

$$h_j = \sqrt{\eta}\max\{|x_j|, \tilde{x}_j\}\text{sign}(x_j), \qquad (3.4.35)$$

$\tilde{x}_j$ is a typical estimation given by users, and η is a small number more than the machine accuracy.

If the standard assumptions of Theorem 3.4.4 hold, and if h_j satisfies

$$h_j = O(\|x_k - x^*\|),$$

this finite-difference Newton's method (3.4.34) maintains the quadratic convergence rate.

Sometimes, some algorithms require us to supply the Hessian matrix-vector product $\nabla^2 f(x)d$, where d is a given vector. Instead of (3.4.32), we can use

$$\nabla^2 f(x_k)d \approx \frac{\nabla f(x_k + hd) - \nabla f(x_k)}{h}, \tag{3.4.36}$$

which also has $O(h)$ approximation error. For obtaining this approximation, the cost is only evaluation of a single gradient at $x_k + hd$. However, the cost of (3.4.32) is evaluation of the gradient at $n+1$ points x_k and $x_k + h_j e_j, j = 1, \cdots, n$.

In the case that the gradient $\nabla f(x)$ is not available, we can only use the function values to approximate the Hessian. The expression (3.4.13) gives the Hessian approximation as follows:

$$(A_k)_{ij} = \frac{[f(x_k + h_i e_i + h_j e_j) - f(x_k + h_i e_i)] - [f(x_k + h_j e_j) - f(x_k)]}{h_i h_j},$$

where

$$h_j = \sqrt[3]{\eta}\max\{|x_j|, \tilde{x}_j\}\text{sign}(x_j)$$

or

$$h_j = (\tilde{\epsilon})^{1/3} x_j,$$

where $\tilde{\epsilon}$ is a machine accuracy. Using the forward-difference and central-difference, the gradient approximations are respectively

$$(\hat{g}_k)_j = \frac{f(x_k + h_j e_j) - f(x_k)}{h_j},\ j = 1, \cdots, n \tag{3.4.37}$$

and

$$(\hat{g}_k)_j = \frac{f(x_k + h_j e_j) - f(x_k - h_j e_j)}{2h_j},\ j = 1, \cdots, n. \tag{3.4.38}$$

Their approximation errors are $O(h_j)$ and $O(h_j^2)$ respectively. In this case, the finite-difference Newton's iteration is

$$x_{k+1} = x_k - A_k^{-1}\hat{g}_k, \tag{3.4.39}$$

where A_k and $\hat{g}_k$ are finite-difference approximations of $\nabla^2 f(x_k)$ and $\nabla f(x_k)$ respectively. Under the standard assumptions of Theorem 3.4.4, we have similarly

$$\begin{aligned} \|x_{k+1} - x^*\| &\leq \|A_k^{-1}\|(\frac{v}{2}\|x_k - x^*\|^2 + \|A_k - \nabla^2 f(x_k)\|\|x_k - x^*\| \\ &+ \|\hat{g}_k - \nabla f(x_k)\|). \end{aligned} \tag{3.4.40}$$

Note that there is an additional term $\|\hat{g}_k - \nabla f(x_k)\|$ than (3.4.28). If we want to get the quadratic convergence rate, it is obvious to require $\|\hat{g}_k - \nabla f(x_k)\| = O(\|x_k - x^*\|^2)$ which implies $h_j = O(\|x_k - x^*\|^2)$. Therefore, it tells us that, when using the central-difference, the iteration (3.4.39) possesses quadratic rate. If we use the forward-difference, the iteration has quadratic rate only when $h_j = O(\|x_k - x^*\|^2)$.

In general, the forward-difference scheme is practical. Although the error of the central-difference scheme is $O(h_j^2)$, as compared to the $O(h_j)$ error in forward-difference, the cost is about twice as much as that of the forward-difference. Hence, we use the central-difference scheme only for those problems which need higher accuracy. Stewart [323] gave a switch rule from forward difference to central difference. Finally, it should be mentioned that, if the gradient is available, it is better to make the best use of it.

3.5 Negative Curvature Direction Method

Another strategy for modifying Newton's method, the negative curvature direction method, is presented, because the modified Newton's methods described above are not adequate for the case in which the Hessian $\nabla^2 f(x_k)$ is indefinite and x_k is close to a saddle point.

Now, we first put forward the definition below.

Definition 3.5.1 *Let $f : R^n \to R$ be twice continuously differentiable on an open set $D \subset R^n$.*

(i) *If $\nabla^2 f(x)$ has at least a negative eigenvalue, then x is said to be an indefinite point.*

(ii) *If x is an indefinite point and d satisfies $d^T\nabla^2 f(x)d < 0$, then d is said to be a negative curvature direction of $f(x)$ at x.*

(iii) *If*

$$s^T\nabla f(x) \le 0,\ d^T\nabla f(x) \le 0,\ d^T\nabla^2 f(x)d < 0,$$

then the vector pair (s, d) is said to be a descent pair at the indefinite point x. If x is not an indefinite point and satisfies

$$s^T\nabla f(x) < 0,\ d^T\nabla f(x) \le 0,\ d^T\nabla^2 f(x)d = 0,$$

then the vector pair (s, d) is said to be a descent pair at x.

As an example of a descent pair, we can choose

$$\begin{aligned} s &= -\nabla f(x), \\ d &= \begin{cases} 0, & \text{if } \nabla^2 f(x) \ge 0, \\ -\text{sign } (u^T\nabla f(x))u, & \text{otherwise,} \end{cases} \end{aligned}$$

where u is a unit eigenvector corresponding to a negative eigenvalue of $\nabla^2 f(x)$.

Obviously, there no longer exists the descent pair if and only if $\nabla f(x) = 0$ and $\nabla^2 f(x)$ is positive semi-definite.

From the definition above, at the stationary point, the negative curvature direction must be a descent direction. At a general point, if the negative curvature direction satisfies $d^T\nabla f(x) = 0$, then both d and $-d$ are descent directions. If $d^T\nabla f(x) \le 0$, d is a descent direction, and if $d^T\nabla f(x) \ge 0$, $-d$ is a descent direction.

In this section, we first give the Gill-Murray stable Newton's method which uses negative curvature direction. Then we discuss two negative curvature direction methods: Fiacco-McCormick method and Fletcher-Freeman method. Finally, we consider the second order Armijo step rules and the second order Wolfe-Powell step rules.

3.5.1 Gill-Murray Stable Newton's Method

The basic idea of Gill-Murray stable Newton's method is: when the Hessian G_k is indefinite, one uses the modified Cholesky factorization to force the matrix G_k to be positive definite; when x_k approaches to a stationary point, use the negative curvature direction to decrease the objective function.

Let the modified Cholesky factorization be

$$\bar{G}_k = G_k + E_k = L_k D_k L_k^T,$$

where

$$D_k = \text{diag}(d_{11}, \cdots, d_{nn}),\ E_k = \text{diag}(e_{11}, \cdots, e_{nn}).$$

When $\|g_k\| \leq \epsilon$ and $\nabla^2 f(x_k)$ is not positive semi-definite, we use the following negative curvature direction algorithm.

Algorithm 3.5.2

Step 1. *Set* $\psi_j = d_{jj} - e_{jj}$, $j = 1, \cdots, n$.

Step 2. *Find the subscript* t, *such that* $\psi_t = \min\{\psi_j \mid j = 1, \cdots, n\}$.

Step 3. *If* $\psi_t \geq 0$, *stop; otherwise, solve*

$$L_k^T d = e_t \tag{3.5.1}$$

for d_k, *where* e_t *is a unit vector with the* t*-th component of* e_t *being* 1. □

Theorem 3.5.3 *Let* G_k *be the Hessian of* $f(x)$ *at* x_k *and*

$$\bar{G}_k = G_k + E_k = L_k D_k L_k^T.$$

If the direction d_k *is obtained by Algorithm 3.5.2, then* d_k *is a negative curvature direction at* x_k, *and at least one in* d_k *and* $-d_k$ *is descent direction at* x_k.

Proof. Since L_k is a unit lower triangular matrix, the solution d_k of (3.5.1) has the form

$$d_k = (\rho_1, \cdots, \rho_{t-1}, 1, 0, \cdots, 0)^T.$$

Then

$$\begin{aligned} d_k^T G_k d_k &= d_k^T \bar{G}_k d_k - d_k^T E_k d_k \\ &= d_k^T L_k D_k L_k^T d_k - d_k^T E_k d_k \\ &= e_t^T D_k e_t - \left(\sum_{r=1}^{t-1} \rho_r^2 e_{rr} + e_{tt} \right) \\ &= d_{tt} - e_{tt} - \sum_{r=1}^{t-1} \rho_r^2 e_{rr} \\ &= \psi_t - \sum_{r=1}^{t-1} \rho_r^2 e_{rr}. \end{aligned}$$

By the modified Cholesky factorization Algorithm 3.3.2, we have

$$\begin{aligned} e_{jj} &= \bar{g}_{jj} - g_{jj} = d_{jj} + \sum_{r=1}^{j-1} l_{jr}^2 d_r - g_{jj} \\ &= d_{jj} - c_{jj} \geq 0, \end{aligned}$$

which indicates that $\sum_{r=1}^{t-1} \rho_r^2 e_{rr} \geq 0$. Also, since $\psi_t < 0$, we obtain $d_k^T G_k d_k < 0$, which means d_k is a negative curvature direction, and $-d_k$ too. If $g_k^T d_k \leq 0$, then d_k is a descent direction; otherwise, $-d_k$ is a descent direction. □

The algorithm below is the Gill-Murray numerically stable Newton's method.

Algorithm 3.5.4

Step 1. *Given a starting point x_0, $\epsilon > 0$. Set $k := 1$.*

Step 2. *Compute g_k and G_k.*

Step 3. *Compute modified Cholesky factorization by using Algorithm 3.3.2*

$$G_k + E_k = L_k D_k L_k^T.$$

Step 4. *If $\|g_k\| > \epsilon$, solve $L_k D_k L_k^T d_k = -g_k$ for d_k, and go to Step 6; otherwise, go to Step 5.*

Step 5 *Perform Algorithm 3.5.2. If it cannot produce d_k (i.e., $\psi_t \geq 0$), stop; otherwise, find d_k and set*

$$d_k = \begin{cases} -d_k, & \text{if } g_k^T d_k > 0, \\ d_k, & \text{otherwise.} \end{cases}$$

Step 6. *Compute line search factor α_k, and set $x_{k+1} = x_k + \alpha_k d_k$.*

Step 7. *If $f(x_{k+1}) \geq f(x_k)$, stop; otherwise, set $k = k + 1$, and go to Step 2.* □

About the convergence of the algorithm above, we have the following theorem.

Theorem 3.5.5 *Let $f : R^n \to R$ be twice continuously differentiable on an open set D. Assume there exists $\bar{x} \in D \subset R^n$ such that the level set*

$$L(\bar{x}) = \{x \mid f(x) \leq f(\bar{x})\}$$

is a bounded closed convex set. Assume that we pick $\epsilon = 0$ in Algorithm 3.5.4, and the starting point $x_0 \in L(\bar{x})$. Then the sequence $\{x_k\}$ generated from Algorithm 3.5.4 satisfies

(i) *when $\{x_k\}$ is a finite sequence, its last element must be the stationary point of $f(x)$;*

(ii) *when $\{x_k\}$ is an infinite sequence, it must have accumulation points, and all accumulation points are the stationary points of $f(x)$.*

The proof is omitted. We refer the interested reader to the original paper Gill and Murray [147].

3.5.2 Fiacco-McCormick Method

The idea of the negative curvature direction method was first presented by Fiacco and McCormick [122] who dealt with the case that the Hessian G_k has negative eigenvalues and employed the exact line search. The idea is simply to go forward along a negative curvature direction and decrease the objective function.

When

$$d_k^T g_k \leq 0 \text{ and } d_k^T G_k d_k < 0, \tag{3.5.2}$$

$$f(x_k + d_k) \approx f(x_k) + d_k^T g_k + \frac{1}{2} d_k^T G_k d_k$$

will be descending. Since G_k is indefinite, the Fiacco-McCormick method uses the decomposition

$$G_k = LDL^T, \tag{3.5.3}$$

where L is a unit lower triangular, and D is a diagonal matrix. If G_k is positive definite, the d_k generated from this decomposition (3.5.3) is a descent direction. However, if there exists a negative d_{ii}, then solve

$$L^T t = a, \tag{3.5.4}$$

where the components a_i of the vector a is defined as

$$a_i = \begin{cases} 1, & d_{ii} \le 0, \\ 0, & d_{ii} > 0. \end{cases} \tag{3.5.5}$$

It is easy to show that

$$d_k = \begin{cases} t, & g_k^T t \le 0, \\ -t, & g_k^T t > 0, \end{cases} \tag{3.5.6}$$

is a negative curvature direction satisfying (3.5.2).

Unfortunately, the decomposition (3.5.3) may be potentially unstable, amplify the rounding errors, and even do not exist. Hence, Fletcher and Freeman [135] employ a stable symmetric indefinite factorization.

3.5.3 Fletcher-Freeman Method

Fletcher and Freeman [135], instead, employ a stable symmetric indefinite factorization due to Bunch and Parlett [33]. For any symmetric matrix G_k, there exists a permutation matrix, such that

$$P^T G_k P = L D L^T, \tag{3.5.7}$$

where L is unit lower triangular, D is a block diagonal matrix with blocks of dimension 1 or 2. The aim to use the permutation matrix is to maintain the symmetricity and numerical stability. Contrasting with the factorization (3.5.3), the factorization (3.5.7) always exists and can be computed by a numerically stable process. Now, for 1×1 pivot case, let A be an $n \times n$ matrix

$$A = A^{(0)} = \begin{bmatrix} a_{11} & \vec{a}_{21}^T \\ \vec{a}_{21} & A_{22} \end{bmatrix}, \tag{3.5.8}$$

where $\vec{a}_{21}$ is $(n-1) \times 1$ vector, A_{22} is an $(n-1) \times (n-1)$ matrix. Eliminating one row and one column yields a reduced matrix $A^{(1)}$:

$$A^{(1)} = A^{(0)} - d_{11} l_1 l_1^T = \begin{bmatrix} 0 & 0^T \\ 0 & A_{22} - \vec{a}_{21}\vec{a}_{21}^T/d_{11} \end{bmatrix}, \tag{3.5.9}$$

where

$$d_{11} = a_{11},\ l_1 = \frac{1}{d_{11}} \begin{bmatrix} a_{11} \\ \vec{a}_{21} \end{bmatrix} = \begin{bmatrix} 1 \\ \vec{a}_{21}/d_{11} \end{bmatrix}. \tag{3.5.10}$$

For 2×2 pivot case, let

$$A^{(0)} = \begin{bmatrix} A_{11} & A_{21}^T \\ A_{21} & A_{22} \end{bmatrix}, \tag{3.5.11}$$

where A_{11} is a 2×2 block matrix, A_{21} is an $(n-2) \times 2$ matrix, and A_{22} is an $(n-2) \times (n-2)$ matrix. Eliminating two rows and two columns yields a reduced matrix $A^{(2)}$:

$$\begin{aligned} A^{(2)} &= A^{(0)} - L_1 D_1 L_1^T = A^{(0)} - \begin{bmatrix} I \\ L_{21} \end{bmatrix} D_1 [I \; L_{21}^T] \\ &= \begin{bmatrix} 0 & 0 \\ 0 & A_{22} - A_{21} D_1^{-1} A_{21}^T \end{bmatrix}, \end{aligned} \tag{3.5.12}$$

where

$$D_1 = A_{11}, \; L_1 = \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} D_1^{-1} = \begin{bmatrix} I \\ A_{21} A_{11}^{-1} \end{bmatrix} \triangleq \begin{bmatrix} I \\ L_{21} \end{bmatrix}. \tag{3.5.13}$$

Next step, we will apply the same process to the remaining matrix $A_{22} - \vec{a}_{21}\vec{a}_{21}^T/d_{11}$ or $A_{22} - A_{21}D_1^{-1}A_{21}^T$ with dimension $(n-1) \times (n-1)$ or $(n-2) \times (n-2)$ respectively. Finally, this recursive procedure gives (3.5.7).

In all the iterations, the algorithm has to identify the pivot block between two pivoting forms. A natural problem is how to identify 1×1 submatrix a_{11} or 2×2 block submatrix A_{11} as a pivot block. Now we describe a criteria as follows. First, compute the largest-magnitude diagonal and the largest-magnitude off-diagonal elements, denoting their respective magnitude by ξ_{dia} and ξ_{off}. If the growth ratio ξ_{dia}/ξ_{off} is acceptable, we choose the diagonal element with largest-magnitude as a pivot and perform row-column exchange such that a_{11} is just the element. Otherwise, we choose the off-diagonal element, say a_{ij}, whose magnitude is ξ_{off}, and choose the corresponding 2×2 block

$$\begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix}$$

as a pivot block. Then we perform row-column exchange such that A_{11} is this 2×2 block.

This decomposition needs $n^3/6 + O(n^2)$ multiplications. Maybe the expensive computation is a disadvantage of this method. A more economical improvement is presented by Bunch and Kaufman [32]. The interested reader

may consult that paper. The forms of L and D produced by the decomposition are a block lower triangular matrix and a block diagonal matrix, for example,

$$D = \begin{bmatrix} * & & & & & \\ & * & * & & & \\ & * & * & & & \\ & & & * & * & \\ & & & * & * & \\ & & & & & * \end{bmatrix}, \; L = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & 0 & 1 & & & \\ & * & * & 1 & & \\ & * & * & 0 & 1 & \\ & * & * & * & * & 1 \end{bmatrix}.$$

The decomposition above is said to be Bunch-Parlett factorization, in brief, B-P factorization which can generate negative curvature direction.

Let G_k have symmetric indefinite factorization

$$G_k = LDL^T. \tag{3.5.14}$$

We solve the triangular system of equations

$$L^T t = a, \tag{3.5.15}$$

where, in the case of 1×1 pivot, the components of a are

$$a_i = \begin{cases} 1, & d_{ii} \le 0, \\ 0, & d_{ii} > 0; \end{cases} \tag{3.5.16}$$

in the case of 2×2 pivot, $\begin{pmatrix} a_i \\ a_{i+1} \end{pmatrix}$ is the unit eigenvector corresponding to the negative eigenvalue of $\begin{bmatrix} d_{ii} & d_{i,i+1} \\ d_{i+1,i} & d_{i+1,i+1} \end{bmatrix}$. Set

$$d_k = \begin{cases} t, & \text{when } g_k^T t \le 0, \\ -t, & \text{when } g_k^T t > 0, \end{cases} \tag{3.5.17}$$

then d_k is the negative curvature direction satisfying (3.5.2). In fact, we have

$$d_k^T G_k d_k = d_k^T LDL^T d_k = a^T Da = \sum_{i:\lambda_i<0} \lambda_i < 0, \tag{3.5.18}$$

and

$$d_k^T g_k \le 0. \tag{3.5.19}$$

In addition, when D has negative eigenvalues, the direction d_k can also be computed by

$$d_k = -L^{-T}\tilde{D}^{+}L^{-1}g_k, \tag{3.5.20}$$

where $\tilde{D}$ is the positive part of D, i.e.,

$$\tilde{D}_i = \begin{cases} d_{ii}, & \text{when } d_{ii} > 0, \\ 0, & \text{otherwise,} \end{cases}$$

and $\tilde{D}^{+}$ is the generalized inverse of $\tilde{D}$.

When D contains at least one zero eigenvalue, the direction d_k can be computed by

$$G_k d_k = LDL^T d_k = 0,\ g_k^T d_k < 0. \tag{3.5.21}$$

When all the eigenvalues of D are positive, all blocks of D are 1×1 elements. In this case, B-P decomposition is reduced to usual Cholesky factorization, and the direction produced is usual Newton's direction

$$d_k = -L^{-T}D^{-1}L^{-1}g_k.$$

It is not difficult to see that the negative curvature descent direction determined by (3.5.17) is limited in some subspace; the direction from (3.5.20) is a Newton's direction limited in the subspace of positive curvature direction. Although the idea of using negative curvature directions is in some ways attractive, Fletcher and Freeman [135] find that it is not satisfactory to use such directions on successive iterations and that if we alternate positive curvature and negative curvature search, i.e., alternate (3.5.17) and (3.5.20), we can get better results. Similarly, if one continuously meets zero eigenvalue, alternating (3.5.20) and (3.5.21) will give better results.

3.5.4 Second-Order Step Rules

Second-Order Armijo Step Rule – McCormick Method

In §2.5 we have discussed Armijo line search rule. Consider

$$\min f(x),\ x \in \mathcal{D} \subset R^n, \tag{3.5.22}$$

where $f : R^n \to R$ is a continuously differentiable function in the open set $\mathcal{D}$.

Given $\beta \in (0,1)$ and $\rho \in (0,1), m_k$ is the least nonnegative integer m such that

$$f(x_k + \beta^m \tau d_k) \leq f(x_k) + \rho \beta^m \tau g_k^T d_k, \tag{3.5.23}$$

where $\tau > 0$, or require α to satisfy

$$f(x_k + \alpha d_k) \leq f(x_k) + \rho \alpha g_k^T d_k. \tag{3.5.24}$$

For the steepest descent method

$$x_{k+1} = x_k - 2^{-i} g_k, \tag{3.5.25}$$

the Armijo rule is

$$f(x_{k+1}) \leq f(x_k) - \rho 2^{-i} \|g_k\|^2, \ \rho \in (0,1). \tag{3.5.26}$$

Instead of using only one descent direction and searching in a line determined by that direction, we search along a curve of the form

$$x(\alpha) = x_k + \phi_1(\alpha) s_k + \phi_2(\alpha) d_k, \tag{3.5.27}$$

where (s_k, d_k) is a descent pair at x_k defined in Definition 3.5.1, $\phi_1(\alpha)$ and $\phi_2(\alpha)$ are polynomials with $\phi_1(0) = \phi_2(0) = 0$.

If we set $\Phi(\alpha) = f(x(\alpha))$ and assume that $\rho \in (0,1)$, there is an $\bar{\alpha} > 0$ such that

$$\Phi(\alpha) \leq \Phi(0) + \rho[\Phi'(0)\alpha + \frac{1}{2}\Phi''(0)\alpha^2] \tag{3.5.28}$$

for all $\alpha \in [0, \bar{\alpha}]$ provided that either $\Phi'(0) < 0$ or $\Phi'(0) = 0$ and $\Phi''(0) < 0$.

Normally, in (3.5.27) we choose $\phi_1(\alpha)$ and $\phi_2(\alpha)$ as lower-order polynomials. The simplest functions of this type are

$$\phi_1(\alpha) = \alpha^2, \ \phi_2(\alpha) = \alpha,$$

which lead to the iteration

$$x(\alpha) = x_k + \alpha^2 s_k + \alpha d_k. \tag{3.5.29}$$

If we set $\alpha = \gamma^i, \ \gamma \in (0,1)$, (3.5.29) becomes

$$x_k(i) = x_k + \gamma^{2i} s_k + \gamma^i d_k \in \mathcal{D}. \tag{3.5.30}$$

The second-order Armijo rule requires us to find $i(k)$ which is the smallest nonnegative integer i such that

$$f(x_k(i)) \leq f(x_k) + \rho\gamma^{2i}[g_k^T s_k + \frac{1}{2} d_k^T G_k d_k], \tag{3.5.31}$$

where $\rho \in (0,1)$, and set $x_{k+1} = x_k(i(k))$. Typically, McCormick [203] chooses $\gamma^2 = \frac{1}{2}$ in (3.5.30).

There exists a finite $i(k)$ satisfying (3.5.31) provided that

$$s_k^T g_k < 0, \text{ whenever } g_k \neq 0 \tag{3.5.32}$$

and

$$d_k^T G_k d_k < 0, \text{ whenever } g_k = 0. \tag{3.5.33}$$

Only if x_k is a point satisfying the second-order optimal condition, there does not exist the descent pair satisfying (3.5.32)-(3.5.33), and the algorithm terminates. The following is the convergence theorem of the second-order Armijo rule.

Theorem 3.5.6 *Let $f : R^n \to R$ be twice continuously differentiable on the open set $\mathcal{D}$, and assume that for some $x_0 \in \mathcal{D}$, the level set*

$$L(x_0) = \{x \in \mathcal{D} \ : \ f(x) \leq f(x_0)\}$$

is compact. Suppose that $\{\|s_k\|\}$ and $\{\|d_k\|\}$ are bounded. If $\{x_k\}$ satisfies (3.5.30) and (3.5.31), then

$$\lim_{k\to\infty} g_k^T s_k = 0 \tag{3.5.34}$$

and

$$\lim_{k\to\infty} d_k^T G_k d_k = 0. \tag{3.5.35}$$

Proof. The sequence $\{f(x_k)\}$ is decreasing and bounded below due to the continuity of f and the compactness of $L(x_0)$. Thus $\{f(x_k) - f(x_{k+1})\}$ converges to zero. Let $i(k)$ be the smallest nonnegative integer such that (3.5.30)-(3.5.31) hold, then there are two cases to consider.

Case 1. Suppose the integer sequence $\{i(k)\}$ is bounded above by $\beta \geq 0$. Then

$$f(x_k) - f(x_{k+1}) \geq -\rho\gamma^{2\beta}[g_k^T s_k + \frac{1}{2} d_k^T G_k d_k]. \tag{3.5.36}$$

Since $-g_k^T s_k \geq 0$ and $-d_k^T G_k d_k \geq 0$, the conclusion follows.

Case 2. Suppose that the integer $\{i(k)\}$ is not bounded above. So, without loss of generality, we can assume that $\lim_{k\to+\infty} i(k) = +\infty$. By (3.5.30)-(3.5.31),

$$f(x(i(k)-1)) - f(x_k) > \rho\gamma^{2[i(k)-1]}[g_k^T s_k + \frac{1}{2} d_k^T G_k d_k]. \tag{3.5.37}$$

For convenience, let

$$p_k = \gamma^{2[i(k)-1]} s_k + \gamma^{i(k)-1} d_k.$$

By using Taylor's theorem and noting that $\nabla^2 f(x)$ is continuous, we have

$$f(x(i(k)-1)) - f(x_k) = p_k^T g_k + \frac{1}{2} p_k^T G_k p_k + o(\gamma^{2[i(k)-1]}). \tag{3.5.38}$$

Combining (3.5.37) and (3.5.38) gives

$$o(\gamma^{2[i(k)-1]}) > (1-\rho)\gamma^{2[i(k)-1]}[-g_k^T s_k - \frac{1}{2} d_k^T G_k d_k]. \tag{3.5.39}$$

Dividing by $(1-\rho)\gamma^{2[i(k)-1]}$ and taking limits yields

$$g_k^T s_k \to 0 \text{ and } d_k^T G_k d_k \to 0. \quad \square$$

Furthermore, we have the following result.

Theorem 3.5.7 *Assume that the conditions in Theorem 3.5.6 hold. In addition, suppose there exist positive constants c_1, c_2, c_3, such that*

$$\|s_k\| \geq c_3 \|g_k\|, \tag{3.5.40}$$
$$d_k^T G_k d_k \leq c_2 \lambda_{G_k}, \tag{3.5.41}$$
$$-s_k^T g_k \geq c_1 \|s_k\| \|g_k\|, \tag{3.5.42}$$

where λ_{G_k} is the most negative eigenvalue of G_k. Then the accumulation point x^* of $\{x_k\}$ satisfies $\nabla f(x^*) = 0$, and $\nabla^2 f(x^*)$ is positive semi-definite with at least one zero eigenvalue.

Proof. From Theorem 3.5.6, we have

$$g_k^T s_k \to 0 \text{ and } d_k^T G_k d_k \to 0.$$

Using (3.5.42) and (3.5.40) gives

$$-g_k^T s_k \geq c_1 c_3 \|g_k\|^2.$$

Thus we get $\|g_k\| \to 0$. Also, it follows from (3.5.41) that $\bar{d}^T \nabla^2 f(x^*)\bar{d} = 0$ with $\bar{d}$ a limit eigenvector of a subsequence of $\{d_k\}$. Therefore, $\nabla^2 f(x^*)$ is positive semi-definite with at least one zero eigenvalue. □

Second-Order Armijo Step Rule — Goldfarb Method

Goldfarb [154] thinks that the iteration

$$x_k(\alpha) = x_k + \alpha^2 s_k + \alpha d_k \tag{3.5.43}$$

is not ideal. The form (3.5.43) may be good in the neighborhood of a saddle point. However, far from a saddle point, it is not a good approach. Then Goldfarb [154] put forward a similar second-order Armijo rule based on the iteration of the form

$$x_k(\alpha) = x_k + \alpha s_k + \alpha^2 d_k, \tag{3.5.44}$$

and gives the following algorithm:

For given γ and ρ, where $0 < \gamma, \rho < 1$, and an initial point x_0, determine x_{k+1}, for $k = 0, 1, \cdots$, as follows:

Choose a descent pair (s_k, d_k) at x_k. If none exists, stop. Otherwise, let $i(k) + 1$ be the smallest nonnegative integer such that

$$f(x_k(\gamma^i)) - f(x_k) \leq \rho[\gamma^i s_k^T g_k + \frac{1}{2}\gamma^{4i} d_k^T G_k d_k] \tag{3.5.45}$$

and set

$$x_{k+1} = x_k(\gamma^{i(k)+1}). \tag{3.5.46}$$

In very much the same manner as Theorem 3.5.6 and Theorem 3.5.7, we have the convergence theorems. So we give them as follows without proof.

Theorem 3.5.8 *Let $f : R^n \to R$ have two continuous derivatives on the open set $\mathcal{D}$ and let the level set $S = \{x \mid f(x) \leq f(x_0)\}$ be a compact subset of $\mathcal{D}$ for a given $x_0 \in \mathcal{D}$. Suppose that an admissible sequence of descent pairs $\{(s_k, d_k)\}$ is used in the above algorithm, and that*

$$-s_k^T g_k \geq c_1 \|s_k\|^2, \tag{3.5.47}$$

$$s_k^T G_k s_k \leq c_2 \|s_k\|^2, \tag{3.5.48}$$

where $0 < c_1, c_2 < \infty$. Then $g_k \to 0, s_k \to 0, \lambda_k \to 0$, and $d_k \to 0$.

Theorem 3.5.9 *In addition to the assumptions of Theorem 3.5.8, assume that the set of stationary points of $f(x)$ in the level set L is finite. Then, if $\{x_k\}$ is the sequence obtained by the second-order Armijo steplength algorithm (3.5.45) and (3.5.46), we have*

$$\lim_{k\to\infty} x_k = x^*,\ g(x^*) = 0,\ G(x^*) \geq 0. \tag{3.5.49}$$

Moreover, if infinitely many $G_k \not\geq 0$, then $G(x^*)$ has at least one eigenvalue equal to zero.

Second-Order Wolfe-Powell Step Rule — Moré-Sorensen Rule

Consider the iteration of the form

$$x(\alpha) = x_k + \alpha^2 s_k + \alpha d_k, \tag{3.5.50}$$

where (s_k, d_k) is a descent pair at x_k. Replacing Wolfe-Powell step rule (2.5.3) and (2.5.7), we ask α to satisfy

$$f(x(\alpha)) \leq f(x) + \rho\alpha^2[\nabla f(x)^T s + \frac{1}{2} d^T \nabla^2 f(x) d], \tag{3.5.51}$$

$$\nabla f(x(\alpha))^T x'(\alpha) \geq \sigma[\nabla f(x)^T d + 2\alpha \nabla f(x)^T s + \alpha d^T \nabla^2 f(x) d], \tag{3.5.52}$$

where $0 < \rho \leq \sigma < 1$. When $d = 0$, these conditions reduce to those of (2.5.3) and (2.5.7). The conditions (3.5.51) and (3.5.52) are said to be the second-order Wolfe-Powell step rule which is contributed by Moré and Sorensen [221].

If (s_k, d_k) is a descent pair at x_k and we set

$$\Phi_k(\alpha) = f(x_k + \alpha^2 s_k + \alpha d_k), \tag{3.5.53}$$

then (3.5.51) and (3.5.52) are equivalent to

$$\Phi_k(\alpha_k) \leq \Phi_k(0) + \frac{1}{2}\rho\Phi_k''(0)\alpha_k^2, \tag{3.5.54}$$

$$\Phi_k'(\alpha_k) \geq \sigma[\Phi_k'(0) + \Phi_k''(0)\alpha_k]. \tag{3.5.55}$$

The second order Wolfe-Powell step rule has a geometric interpretation as shown in Figure 3.5.1.

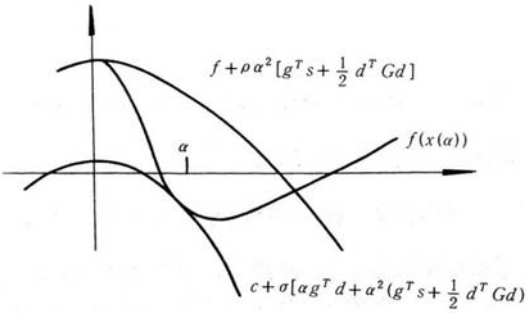


Figure 3.5.1 Second-order Wolfe-Powell step rule

Similar to the preceding discussion, we now give the following convergence results.

Theorem 3.5.10 *Let $f : R^n \to R$ have twice continuous derivatives on the open set $\mathcal{D}$, and assume that for some $x_0 \in \mathcal{D}$, the level set*

$$L(x_0) = \{x \in \mathcal{D} \mid f(x) \leq f(x_0)\}$$

is a compact subset of $\mathcal{D}$. If $\{x_k\}$ satisfies (3.5.50)-(3.5.52), then

$$\lim_{k\to\infty} g_k^T s_k = 0 \text{ and } \lim_{k\to\infty} d_k^T G_k d_k = 0. \tag{3.5.56}$$

Proof. From (3.5.53) we have $\Phi_k'(0) = g_k^T d_k$ and

$$\Phi_k''(0) = 2g_k^T s_k + d_k^T G_k d_k.$$

Since (s_k, d_k) is a descent pair, $\Phi_k'(0) \leq 0$ and $\Phi_k''(0) < 0$. Thus (3.5.51) implies that $\{x_k\} \subset L(x_0)$. By the continuity of f and compactness of $L(x_0)$ we have that $\{f_k - f_{k+1}\}$ converges to zero. Since

$$f_k - f_{k+1} \geq -\frac{1}{2}\rho\Phi_k''(0)\alpha_k^2 \geq 0,$$

it follows that

$$\lim_{k\to\infty} \alpha_k^2 g_k^T s_k = 0 \tag{3.5.57}$$

and

$$\lim_{k\to\infty} \alpha_k^2 d_k^T G_k d_k = 0. \tag{3.5.58}$$

From (3.5.55) we have

$$\Phi_k'(\alpha_k) - \Phi_k'(0) - \alpha_k \Phi_k''(0) \geq -(1-\sigma)[\Phi_k'(0) + \Phi_k''(0)\alpha_k],$$

and hence

$$\Phi_k'(\alpha_k) - \Phi_k'(0) - \alpha_k \Phi_k''(0) \geq -(1-\sigma)\Phi_k''(0)\alpha_k.$$

An application of the mean-value theorem yields that for some $\theta_k \in (0, \alpha_k)$,

$$\Phi_k''(\theta_k) - \Phi_k''(0) \geq -(1-\sigma)\Phi_k''(0). \tag{3.5.59}$$

In the following, we prove (3.5.56) by contradiction. Suppose either the first equality or the second equality does not hold, then there is a subsequence $\{k_i\}$ and $\eta > 0$ such that

$$-\Phi_{k_i}''(0) \geq \eta > 0. \tag{3.5.60}$$

Hence (3.5.59) implies that $\{\alpha_{k_i}\}$ does not converge to zero. However, if $\{\alpha_{k_i}\}$ does not converge to zero and (3.5.60) holds, then (3.5.57) and (3.5.58) cannot be satisfied. This contradiction establishes the theorem. □

Furthermore, we have

Theorem 3.5.11 *Let $f : R^n \to R$ have twice continuous derivatives on the open set $\mathcal{D}$, and assume that, for some $x_0 \in \mathcal{D}$, the level set $L(x_0) = \{x \in \mathcal{D} \mid f(x) \leq f(x_0)\}$ is compact. In addition, assume that f has a finite number of critical points in $L(x_0)$. Then, if $\{x_k\}$ is a sequence obtained by the second-order step rule (3.5.50)-(3.5.52), we have*

$$\lim_{k\to\infty} x_k = x^*,\ g(x^*) = 0,\ G(x^*) \geq 0. \tag{3.5.61}$$

Moreover, if infinitely many $G_k \not\geq 0$, then $G(x^*)$ has at least one eigenvalue equal to zero.

Proof. It is similar to the proof of Theorem 3.5.7. □

Determine Descent Pair (s_k, d_k)

Finally, we mention a way to obtain the descent pair (s_k, d_k) which satisfies all of the requirements of Theorem 3.5.10 and 3.5.11. First, consider computing s_k. Assume that

$$G_k = L_k D_k L_k^T$$

is the Bunch-Parlett symmetric indefinite factorization where we omit the permutations, L_k is a unit lower triangular matrix, D_k a block diagonal matrix with 1×1 or 2×2 diagonal block. Let

$$D_k = U_k \Lambda_k U_k^T$$

be the spectral decomposition of D_k. Set

$$\bar{\lambda}_j^{(k)} = \max \left\{ |\lambda_j^{(k)}|,\ \epsilon n \max_{1 \le i \le n} |\lambda_i^{(k)}|,\ \epsilon \right\}, \quad j = 1, \cdots, n,$$

$$\bar{\Lambda}_k = \mathrm{diag}(\bar{\lambda}_1^{(k)}, \cdots, \bar{\lambda}_n^{(k)}),$$

where ϵ is the relative machine precision. Set

$$\bar{D}_k = U_k \bar{\Lambda}_k U_k^T.$$

We obtain s_k as the solution of

$$L_k \bar{D}_k L_k^T s = -g_k.$$

Next, the negative curvature direction d_k is obtained as the solution of

$$L_k^T d_k = \pm |\min\{\lambda(D_k), 0\}|^{\frac{1}{2}} z_k,$$

where $\lambda(D_k)$ is the smallest eigenvalue of D_k and z_k the corresponding unit eigenvector of D_k. The other way to obtain a negative curvature direction d_k is to solve

$$L_k^T d_k = \pm \sum_{\lambda_j(D_k) \le 0} z_j.$$

3.6 Inexact Newton's Method

As mentioned before, the pure Newton's method is expensive in each iteration, especially when the dimension n is large. Also, the quadratic model used to derive the Newton equation may not provide a good prediction of the behavior of the function, especially when the iterate x_k is remote from the solution x^*. In this section, we consider a class of inexact Newton's methods in which we only approximately solve the Newton equation. In the following, we discuss this class of methods for solving nonlinear equations $F(x) = 0$. It

is not difficult for readers to deal with unconstrained optimization problems by using this way.

Consider solving the nonlinear equations

$$F(x) = 0, \tag{3.6.1}$$

where $F : R^n \to R^n$ is assumed to have the following properties:

A1 There exists x^* such that $F(x^*) = 0$.

A2 F is continuously differentiable in the neighborhood of x^*.

A3 $F'(x^*)$ is nonsingular.

Recall that the basic Newton's step is obtained by solving

$$F'(x_k)s_k = -F(x_k) \tag{3.6.2}$$

and setting

$$x_{k+1} = x_k + s_k. \tag{3.6.3}$$

Now, we consider inexact Newton's method: solve

$$F'(x_k)s_k = -F(x_k) + r_k, \tag{3.6.4}$$

where

$$\|r_k\| \le \eta_k \|F(x_k)\|. \tag{3.6.5}$$

Set

$$x_{k+1} = x_k + s_k. \tag{3.6.6}$$

Here, $r_k = F'(x_k)s_k + F(x_k)$ denotes the residual, and $\{\eta_k\}$ (with $0 < \eta_k < 1$) is a forcing sequence which controls the inexactness.

Next, we study the local convergence of inexact Newton's methods.

Lemma 3.6.1 *Let $F : D \subset R^n \to R^n$ be continuously differentiable in a neighborhood of $x^* \in D$, and let $F'(x^*)$ be nonsingular. Then there exist $\delta > 0, \xi > 0$, and $\epsilon > 0$, such that when $\|y - x^*\| < \delta$ and $y \in D$, $F'(y)$ is nonsingular and*

$$\|F'(y)^{-1}\| \le \xi. \tag{3.6.7}$$

Also, $F'(y)^{-1}$ is continuous at x^*, that is

$$\|F'(y)^{-1} - F'(x^*)^{-1}\| < \epsilon. \tag{3.6.8}$$

Proof. Set $\alpha = \|F'(x^*)^{-1}\|$. For a given $\beta < \alpha^{-1}$, choose δ such that when $\|y - x^*\| < \delta$ with $y \in D$,

$$\|F'(x^*) - F'(y)\| \leq \beta.$$

It follows from Von-Neumann Theorem 1.2.5 that $F'(y)$ is invertible, and (3.6.7) holds with $\xi = \alpha/(1 - \beta\alpha)$. Thus,

$$\begin{aligned} \|F'(x^*)^{-1} - F'(y)^{-1}\| &= \|F'(x^*)^{-1}(F'(y) - F'(x^*))F'(y)^{-1}\| \\ &\leq \alpha\xi\|F'(x^*) - F'(y)\| \\ &\leq \alpha\beta\xi \\ &\triangleq \epsilon, \end{aligned}$$

which says that the continuity of F' guarantees the continuity of $(F')^{-1}$. □

In the following, we establish the linear convergence in Theorem 3.6.2 and superlinear convergence in Theorem 3.6.4.

Theorem 3.6.2 *Let $F : R^n \to R^n$ satisfy the properties (A1)–(A3). Assume that the sequence $\{\eta_k\}$ satisfies $0 \leq \eta_k \leq \eta < t < 1$. Then, for some $\epsilon > 0$, if the starting point x_0 is sufficiently near x^*, the sequence $\{x_k\}$ generated by inexact Newton's method (3.6.4)–(3.6.6) converges to x^*, and the convergence rate is linear, i.e.,*

$$\|x_{k+1} - x^*\|_* \leq t\|x_k - x^*\|_*, \tag{3.6.9}$$

where $\|y\|_* = \|F'(x^*)y\|$.

Proof. Since $F'(x^*)$ is nonsingular, for $y \in R^n$, we have

$$\frac{1}{\mu}\|y\| \leq \|y\|_* \leq \mu\|y\|, \tag{3.6.10}$$

where

$$\mu = \max\{\|F'(x^*)\|, \|F'(x^*)^{-1}\|\}. \tag{3.6.11}$$

Since $\eta < t$, there exists sufficiently small $\gamma > 0$, such that

$$(1 + \gamma\mu)[\eta(1 + \mu\gamma) + 2\mu\gamma] \leq t. \tag{3.6.12}$$

Now choose $\epsilon > 0$ sufficiently small, such that if $\|y - x^*\| \leq \mu^2\epsilon$, we have

$$\|F'(y) - F'(x^*)\| \leq \gamma, \tag{3.6.13}$$

$$\|F'(y)^{-1} - F'(x^*)^{-1}\| \leq \gamma, \tag{3.6.14}$$

$$\|F(y) - F(x^*) - F'(x^*)(y - x^*)\| \leq \gamma\|y - x^*\|. \tag{3.6.15}$$

Let $\|x_0 - x^*\| \leq \epsilon$. We now prove (3.6.9) by induction. By using (3.6.10)–(3.6.11) and assumption of the induction, we have

$$\begin{aligned} \|x_k - x^*\| &\leq \mu\|x_k - x^*\|_* \leq \mu t^k \|x_0 - x^*\|_* \\ &\leq \mu^2 \|x_0 - x^*\| \leq \mu^2 \epsilon. \end{aligned}$$

Then, when $y = x_k$, (3.6.13)–(3.6.15) hold. Since

$$\begin{aligned} & F'(x^*)(x_{k+1} - x^*) \\ = \; & F'(x^*)(x_k - x^* - F'(x_k)^{-1}F(x_k) + F'(x_k)^{-1} r_k) \\ = \; & F'(x^*)F'(x_k)^{-1}[F'(x_k)(x_k - x^*) - F(x_k) + r_k] \\ = \; & [I + F'(x^*)(F'(x_k)^{-1} - F'(x^*)^{-1})][r_k + (F'(x_k) - F'(x^*))(x_k - x^*) \\ & -(F(x_k) - F(x^*) - F'(x^*)(x_k - x^*))], \end{aligned} \tag{3.6.16}$$

by taking norms and using (3.6.11), (3.6.14), (3.6.5), (3.6.13) and (3.6.15), we obtain

$$\begin{aligned} & \|x_{k+1} - x_k\|_* \\ \leq \; & [1 + \|F'(x^*)\|\|F'(x_k)^{-1} - F'(x^*)^{-1}\|][\|r_k\| + \\ & \|F'(x_k) - F'(x^*)\|\|x_k - x^*\| + \|F(x_k) - F(x^*) - F'(x^*)(x_k - x^*)\|] \\ \leq \; & (1 + \mu\gamma)[\eta_k \|F(x_k)\| + \gamma\|x_k - x^*\| + \gamma\|x_k - x^*\|]. \end{aligned} \tag{3.6.17}$$

Note that

$$F(x_k) = [F'(x^*)(x_k - x^*)] + [F(x_k) - F(x^*) - F'(x^*)(x_k - x^*)],$$

taking the norm gives

$$\|F(x_k)\| \leq \|x_k - x^*\|_* + \gamma\|x_k - x^*\|. \tag{3.6.18}$$

Substituting (3.6.18) into (3.6.17) and using (3.6.10) and (3.6.12) yield

$$\begin{aligned} \|x_{k+1} - x^*\|_* &\leq (1 + \mu\gamma)[\eta_k(\|x_k - x^*\|_* + \gamma\|x_k - x^*\|) + 2\gamma\|x_k - x^*\|] \\ &\leq (1 + \mu\gamma)[\eta(1 + \mu\gamma) + 2\mu\gamma]\|x_k - x^*\|_* \\ &\leq t\|x_k - x^*\|_*. \quad \square \end{aligned}$$

Below, we discuss the superlinear convergence rate of the inexact Newton's methods. We first give a lemma.

Lemma 3.6.3 *Let*

$$\alpha = \max\{\|F'(x^*)\| + \frac{1}{2\beta}, 2\beta\},$$

where $\beta = \|F'(x^*)^{-1}\|$. *Then, for* $\|y - x^*\|$ *sufficiently small, the inequality*

$$\frac{1}{\alpha}\|y - x^*\| \leq \|F(y)\| \leq \alpha\|y - x^*\| \tag{3.6.19}$$

holds.

Proof. From the continuous differentiability of F, we know that there exists a sufficiently small $\delta > 0$, such that when $\|y - x^*\| < \delta$,

$$\|F(y) - F(x^*) - F'(x^*)(y - x^*)\| \leq \frac{1}{2\beta}\|y - x^*\| \tag{3.6.20}$$

holds. Note that

$$F(y) = [F'(x^*)(y - x^*)] + [F(y) - F(x^*) - F'(x^*)(y - x^*)],$$

and take norms, then we have

$$\begin{aligned} \|F(y)\| &\leq \|F'(x^*)\|\|y - x^*\| + \|F(y) - F(x^*) - F'(x^*)(y - x^*)\| \\ &\leq \left(\|F'(x^*)\| + \frac{1}{2\beta}\right)\|y - x^*\| \end{aligned} \tag{3.6.21}$$

and

$$\begin{aligned} \|F(y)\| &\geq \|F'(x^*)^{-1}\|^{-1}\|y - x^*\| - \|F(y) - F(x^*) - F'(x^*)(y - x^*)\| \\ &\geq \left(\|F'(x^*)^{-1}\|^{-1} - \frac{1}{2\beta}\right)\|y - x^*\| \\ &= \frac{1}{2\beta}\|y - x^*\|. \end{aligned} \tag{3.6.22}$$

Combining (3.6.21) and (3.6.22) gives (3.6.19). □

Theorem 3.6.4 *Let the assumptions of Theorem 3.6.2 be satisfied. Assume that the sequence* $\{x_k\}$ *generated by the inexact Newton's method converges to* x^*, *then, if and only if*

$$\|r_k\| = o(\|F(x_k)\|), \quad k \to \infty, \tag{3.6.23}$$

$\{x_k\}$ *converges to* x^* *superlinearly.*

Proof. Assume that $\{x_k\}$ converges to x^* superlinearly. Since

$$\begin{aligned} r_k &= F(x_k) + F'(x_k)(x_{k+1} - x_k) \\ &= [F(x_k) - F(x^*) - F'(x^*)(x_k - x^*)] - [F'(x_k) - F'(x^*)](x_k - x^*) \\ &\quad +[F'(x^*) + (F'(x_k) - F'(x^*))](x_{k+1} - x^*), \end{aligned}$$

taking norms and using property (A1)-(A3) and the superlinear convergence property of $\{x_k\}$ yield

$$\begin{aligned} \|r_k\| &\leq \|F(x_k) - F(x^*) - F'(x^*)(x_k - x^*)\| + \|F'(x_k) - F'(x^*)\|\|x_k - x^*\| \\ &\quad +[\|F'(x^*)\| + \|F'(x_k) - F'(x^*)\|]\|x_{k+1} - x^*\| \\ &= o(\|x_k - x^*\|) + o(1)\|x_k - x^*\| \\ &\quad +[\|F'(x^*)\| + o(1)]o(\|x_k - x^*\|). \end{aligned} \tag{3.6.24}$$

Thus, by use of Lemma 3.6.3, we have, when $k \to \infty$, that

$$\|r_k\| = o(\|x_k - x^*\|) = o(\|F(x_k)\|). \tag{3.6.25}$$

Conversely, assume that $\|r_k\| = o(\|F(x_k)\|)$. From (3.6.16), it follows that

$$\begin{aligned} & \|x_{k+1} - x^*\| \\ \leq\ & (\|F'(x^*)^{-1}\| + \|F'(x_k)^{-1} - F'(x^*)^{-1}\|)(\|r_k\| \\ & +\|F'(x_k) - F'(x^*)\|\|x_k - x^*\| + \|F(x_k) - F(x^*) - F'(x^*)(x_k - x^*)\|) \\ =\ & (\|F'(x^*)^{-1}\| + o(1))(o(\|F(x_k)\|) + o(1)\|x_k - x^*\| + o(\|x_k - x^*\|)). \end{aligned}$$

Therefore, we get from Lemma 3.6.3 that

$$\begin{aligned} \|x_{k+1} - x^*\| &= o(\|F(x_k)\|) + o(\|x_k - x^*\|) \\ &= o(\|x_k - x^*\|), \end{aligned}$$

which shows the superlinear convergence of sequence $\{x_k\}$. □

The following corollary indicates that when $\{\eta_k\} \to 0$, the sequence $\{x_k\}$ converges to x^* superlinearly.

Corollary 3.6.5 *Assume that the sequence $\{x_k\}$ generated by inexact Newton's method converges to x^*. Then, if sequence $\{\eta_k\}$ converges to zero, the sequence $\{x_k\}$ converges to x^* superlinearly.*

Proof. If $\lim_{k\to\infty} \eta_k = 0$, then

$$\limsup_{k\to\infty} \frac{\|r_k\|}{\|F(x_k)\|} = 0,$$

which means that $\|r_k\| = o(\|F(x_k)\|)$. Then the conclusion is obtained from Theorem 3.6.4. □

There are several proofs of local convergence for the inexact Newton's method. Below, we give outlines of other proofs.

The outline of the second proof is as follows.

From (3.6.4)–(3.6.6), we have

$$\begin{aligned} & x_{k+1} - x^* \\ = \; & x_k - x^* - F'(x_k)^{-1}F(x_k) + F'(x_k)^{-1}r_k \\ = \; & F'(x_k)^{-1}[F'(x_k)(x_k - x^*) - F(x_k) + F(x^*) + r_k]. \end{aligned} \tag{3.6.26}$$

Taking norms, and using (3.6.7), (3.6.15) and Lipschitzian continuity of $F(x)$, i.e., $\|F(x_k)\| = \|F(x_k) - F(x^*)\| \le L\|x_k - x^*\|$, we obtain

$$\begin{aligned} \|x_{k+1} - x^*\| &\le \xi[\gamma\|x_k - x^*\| + \eta_k L\|x_k - x^*\|] \\ &\le \xi(\gamma + \eta_k L)\|x_k - x^*\|. \end{aligned} \tag{3.6.27}$$

If we choose γ and η_k such that $\xi(\gamma + \eta_k L) < 1$, then $\{x_k\}$ converges to x^* linearly. If we choose $\eta_k \to 0$ and note that γ is sufficiently small, then $\xi(\gamma + \eta_k L) \to 0$, and thus the sequence $\{x_k\}$ converges to x^* superlinearly.

The third proof is as follows.

Theorem 3.6.6 *Let $F : R^n \to R^n$ satisfy the properties (A1)–(A3). Assume that the sequence $\{\eta_k\}$ satisfies $0 \le \eta_k \le \eta < 1$. Then, for some $\epsilon > 0$, if the starting point x_0 is sufficiently near x^*, the sequence $\{x_k\}$ generated by inexact Newton's method (3.6.4)–(3.6.6) converges to x^*, and the convergence rate is linear, i.e., for all k sufficiently large,*

$$\|x_{k+1} - x^*\| \le c\|x_k - x^*\| \tag{3.6.28}$$

for some constant $0 < c < 1$.

Furthermore, if $\eta_k \to 0$, then the sequence $\{x_k\}$ converges to x^* superlinearly. If $\eta_k = O(\|F(x_k)\|)$, then the sequence converges to x^* quadratically.

Proof. From (3.6.4),

$$s_k = F'(x_k)^{-1}[-F(x_k) + r_k].$$

Taking norms and using (3.6.7) and (3.6.5), we obtain

$$\|s_k\| \le \xi(\|F(x_k)\| + \|r_k\|) \le \xi(1+\eta)\|F(x_k)\| \le 2\xi\|F(x_k)\|. \tag{3.6.29}$$

By using Taylor's theorem, (3.6.4) and the above expression, we have

$$\begin{aligned} F(x_{k+1}) &= F(x_k) + F'(x_k)s_k + O(\|s_k\|^2) \\ &= r_k + O(\|F(x_k)\|^2). \end{aligned} \tag{3.6.30}$$

By taking norms and using (3.6.5), we get

$$\|F(x_{k+1})\| \le \eta_k\|F(x_k)\| + O(\|F(x_k)\|^2). \tag{3.6.31}$$

Dividing both sides by $\|F(x_k)\|$, passing to the $\limsup, k \to \infty$, noting that $\eta_k \le \eta < 1$, we deduce

$$\limsup_{k\to\infty} \frac{\|F(x_{k+1})\|}{\|F(x_k)\|} \le \eta < 1. \tag{3.6.32}$$

By using Corollary 1.2.26 (or Lemma 3.6.3) we immediately obtain

$$\limsup_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \le C \limsup_{k\to\infty} \frac{\|F(x_{k+1})\|}{\|F(x_k)\|} \tag{3.6.33}$$

for some constant C. When $\{x_k\}$ is sufficiently close to x^* and $C\eta < 1$, the sequence $\{x_k\}$ converges to x^* locally and linearly.

Furthermore, if $\eta_k \to 0$, then

$$\limsup_{k\to\infty} \frac{\|r_k\|}{\|F(x_k)\|} = 0,$$

i.e., $\|r_k\| = o(\|F(x_k)\|)$. By using (3.6.30) and taking norms, we have

$$\limsup_{k\to\infty} \frac{\|F(x_{k+1})\|}{\|F(x_k)\|} = 0 \tag{3.6.34}$$

which indicates the superlinear convergence in the function value sequence $\{F(x_k)\}$. It is easy to see that

$$\limsup_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0 \tag{3.6.35}$$

by use of Corollary 1.2.26 (or (3.6.19)).

If

$$\eta_k = O(\|F(x_k)\|), \tag{3.6.36}$$

then there exists some constant c_1 such that $\eta_k \leq c_1\|F(x_k)\|$. By using (3.6.5) we get that

$$\limsup_{k\to\infty} \frac{\|r_k\|}{\|F(x_k)\|^2} \leq c_1, \tag{3.6.37}$$

which shows that

$$r_k = O(\|F'(x_k)\|^2). \tag{3.6.38}$$

We have immediately from (3.6.30) that

$$\limsup_{k\to\infty} \frac{\|F(x_{k+1})\|}{\|F(x_k)\|^2} = c \tag{3.6.39}$$

for some constant c, which means quadratic convergence of $\{F(x_k)\}$. And therefore we have that

$$\limsup_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^2} = c. \quad \square \tag{3.6.40}$$

It is easy to apply the above result to unconstrained optimization problem $\min_{x\in R^n} f(x)$. In fact, instead of (3.6.4)–(3.6.5), we use

$$\nabla^2 f(x_k)s_k = -\nabla f(x_k) + r_k, \tag{3.6.41}$$

where

$$\|r_k\| \leq \eta_k\|\nabla f(x_k)\|, \tag{3.6.42}$$

and then we can get the same results for unconstrained optimization problems. Similar to the above discussion we have the following theorem.

Theorem 3.6.7 *Suppose that $\nabla f(x)$ is continuously differentiable in a neighborhood of a minimizer x^*, and assume that $\nabla^2 f(x^*)$ is positive definite. Consider the iteration $x_{k+1} = x_k + s_k$, where s_k is an inexact Newton step satisfying (3.6.41) and (3.6.42). Assume that the sequence $\{\eta_k\}$ satisfies $0 \leq \eta_k \leq \eta < 1$. Then, if the starting point x_0 is sufficiently near x^*, the sequence $\{x_k\}$ converges to x^* linearly, i.e., for all k sufficiently large,*

$$\|x_{k+1} - x^*\| \leq c\|x_k - x^*\| \tag{3.6.43}$$

for some constant $0 < c < 1$.

The sequence $\{x_k\}$ converges to x^* superlinearly if $\|r_k\| = o(\|\nabla f(x_k)\|)$.

The sequence $\{x_k\}$ converges to x^* quadratically if $\|r_k\| = O(\|\nabla f(x_k)\|^2)$.

About the implementation of the inexact Newton's method, we can generate the search direction by applying the conjugate gradient method to the Newton's equation $\nabla^2 f(x_k)s_k = -\nabla f(x_k)$, and then ask that the termination test (3.6.42) be satisfied.

Inexact Newton's method is an efficient method, especially for large scale nonlinear equations and optimization problems. Inexact Newton's method was due to Dembo, Eisenstat and Steihaug [83]. The other important works about this method can be found in Steihaug [321], Dennis and Walker [99], Ypma [365], and Nash [229].

Exercises

1. Let $f(x) = \frac{3}{2}x_1^2 + \frac{1}{2}x_2^2 - x_1x_2 - 2x_1$. Let the initial point $x^{(0)} = (-2, 4)^T$. Minimize $f(x)$ by use of the steepest descent method and Newton's method, respectively.

2. Let
(1) $f(x) = \frac{1}{2}(x_1^2 + 9x_2^2)$;
(2) $f(x) = \frac{1}{2}(x_1^2 + 10^4 x_2^2)$.
Discuss the convergence rate of the steepest descent method.

3. Let $f(x) = \frac{1}{2}x^T x + \frac{1}{4}\sigma(x^T Ax)^2$, where

$$A = \begin{bmatrix} 5 & 1 & 0 & \frac{1}{2} \\ 1 & 4 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 3 & 0 \\ \frac{1}{2} & 0 & 0 & 2 \end{bmatrix}.$$

Let (1) $x^{(0)} = (\cos 70^\circ, \sin 70^\circ, \cos 70^\circ, \sin 70^\circ)^T$;
(2) $x^{(0)} = (\cos 50^\circ, \sin 50^\circ, \cos 50^\circ, \sin 50^\circ)^T$.
In the case of $\sigma = 1$ and $\sigma = 10^4$, discuss the numerical results and behavior of convergence rate of pure Newton's method and Newton's method with line search respectively.

4. Minimize the Rosenbrock function $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ by the steepest descent method and Newton's method respectively, where $x^{(0)} = (-1.2, 1)^T, x^* = (1, 1)^T, f(x^*) = 0$.

5. By your opinion, state the reasons that the steepest descent method converges slowly.

6. Prove the convergence of the inexact Newton methods.

Chapter 4

Conjugate Gradient Method

In the preceding chapter we have discussed the steepest descent method and the Newton method. In this chapter we introduce the conjugate gradient method which is one between the steepest descent method and the Newton method. The conjugate gradient method deflects the direction of the steepest descent method by adding to it a positive multiple of the direction used in the last step. This method only requires the first-order derivatives but overcomes the steepest descent method's shortcoming of slow convergence. At the same time, the method need not save and compute the second-order derivatives which are needed by Newton method. In particular, since it does not require the Hessian matrix or its approximation, it is widely used to solve large scale optimization problems.

In this chapter, we will discuss the derivation, the properties, the algorithm and numerical experiments, and the convergence of the conjugate gradient method. Note that the restarting and preconditioning are very important to improve the conjugate gradient method. As a beginning, we first introduce the concept of conjugate directions and the conjugate direction method.

4.1 Conjugate Direction Methods

One of the main properties of the conjugate gradient method is that its directions are conjugate. Now, we first introduce conjugate directions and conjugate direction methods.

Definition 4.1.1 *Let G be an $n \times n$ symmetric and positive definite matrix, $d_1, d_2, \cdots, d_m \in R^n$ be non-zero vectors, $m \leq n$. If*

$$d_i^T G d_j = 0, \ \forall i \neq j, \tag{4.1.1}$$

the vectors $d_1, d_2, \cdots, d_m$ are called G-conjugate or simply conjugate.

Obviously, if vectors $d_1, \cdots, d_m$ are G-conjugate, then they are linearly independent. If $G = I$, the conjugacy is equivalent to the usual orthogonality.

A general conjugate direction method has the following steps:

Algorithm 4.1.2 *(General Conjugate Direction Method)*

Step 1. *Given an initial point $x_0, \epsilon > 0, k := 0$. Compute $g_0 = g(x_0)$; Compute d_0 such that $d_0^T g_0 < 0$.*

Step 2. *If $\|g_k\| \leq \epsilon$, stop.*

Step 3. *Compute α_k such that*

$$f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha d_k).$$

Set $x_{k+1} = x_k + \alpha_k d_k$.

Step 4. *Compute d_{k+1} by some conjugate direction method, such that $d_{k+1}^T G d_j = 0, j = 0, 1, \cdots, k$.*

Step 5. *Set $k := k + 1$, go to Step 2.* □

The conjugate direction method is an important class of optimization methods. The following theorem shows that, under exact line search, the conjugate direction methods have quadratic termination property, which means that the method terminates in at most n steps when it is applied to a quadratic function with positive definite Hessian.

Theorem 4.1.3 *(Principal Theorem of Conjugate Direction Method) For a quadratic function with positive definite Hessian G, the conjugate direction method terminates in at most n exact line searches. Each x_{i+1} is the minimizer in the subspace generated by x_0 and the directions $d_0, \cdots, d_i$, that is $\{x \mid x = x_0 + \sum_{j=0}^{i} \alpha_j d_j\}$.*

Proof. Since G is positive definite and the conjugate directions $d_0, d_1, \cdots$ are linearly independent, it is enough to prove for all $i \leq n-1$ that

$$g_{i+1}^T d_j = 0, \ j = 0, \cdots, i. \tag{4.1.2}$$

(Note that if (4.1.2) holds, we immediately have $g_n^T d_j = 0, j = 0, \cdots, n-1$ and $g_n = 0$, therefore x_n is a minimizer.)

To prove (4.1.2), we consider two cases $j < i$ and $j = i$. Keep in mind that

$$y_k \stackrel{Def}{=} g_{k+1} - g_k = G(x_{k+1} - x_k) = \alpha_k G d_k. \tag{4.1.3}$$

When $j < i$, by use of exact line search and the conjugacy, we have

$$\begin{aligned} g_{i+1}^T d_j &= g_{j+1}^T d_j + \sum_{k=j+1}^{i} y_k^T d_j \\ &= g_{j+1}^T d_j + \sum_{k=j+1}^{i} \alpha_k d_k^T G d_j \\ &= 0. \end{aligned} \tag{4.1.4}$$

When $j = i$, (4.1.2) is a direct result from the exact line search. Thus (4.1.2) holds and we complete the proof. □

This theorem is simple but important. All conjugate direction methods rely on this theorem. We reemphasize that, under exact line search, all conjugate direction methods satisfy (4.1.2), and have quadratic termination property. This shows that conjugacy plus exact line search implies quadratic termination.

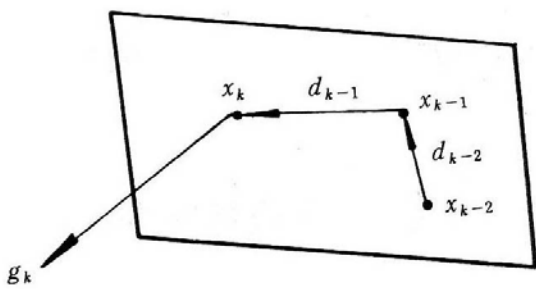


Figure 4.1.1 The gradient of conjugate direction method satisfies (4.1.2)

4.2 Conjugate Gradient Method

4.2.1 Conjugate Gradient Method

In the conjugate direction method described in §4.1, there is not an explicit procedure for generating a conjugate system of vectors $d_1, d_2, \cdots$. In this section we describe a method for generating mutually conjugate direction vectors, which is theoretically appealing and computationally effective. This method is called the conjugate gradient method.

In conjugate direction methods, the conjugate gradient method is of particular importance. Now it is widely used to solve large scale optimization problems. The conjugate gradient method was originally proposed by Hestenes and Stiefel [173] in the 1950s to solve linear systems. Since solving a linear system is equivalent to minimizing a positive definite quadratic function, Fletcher and Reeves [138] in the 1960s modified it and developed a conjugate gradient method for unconstrained minimization. By means of conjugacy, the conjugate gradient method makes the steepest descent direction have conjugacy, and thus increases the efficiency and reliability of the algorithm.

Now we derive the conjugate gradient method for the quadratic case.

Let

$$f(x) = \frac{1}{2}x^T Gx + b^T x + c, \tag{4.2.1}$$

where G is an $n \times n$ symmetric positive definite matrix, $b \in R^n$ and c is a real number. Obviously, the gradient of $f(x)$ is

$$g(x) = Gx + b. \tag{4.2.2}$$

Set

$$d_0 = -g_0, \tag{4.2.3}$$

then we have

$$x_1 = x_0 + \alpha_0 d_0, \tag{4.2.4}$$

where α_0 is generated by an exact line search. Then we have

$$g_1^T d_0 = 0. \tag{4.2.5}$$

Set

$$d_1 = -g_1 + \beta_0 d_0, \tag{4.2.6}$$

and choose β_0 such that

$$d_1^T G d_0 = 0. \tag{4.2.7}$$

It follows from multiplying (4.2.6) by $d_0^T G$ that

$$\beta_0 = \frac{g_1^T G d_0}{d_0^T G d_0} = \frac{g_1^T (g_1 - g_0)}{d_0^T (g_1 - g_0)} = \frac{g_1^T g_1}{g_0^T g_0}. \tag{4.2.8}$$

In general, in the k-th iteration, set

$$d_k = -g_k + \sum_{i=0}^{k-1} \beta_i d_i. \tag{4.2.9}$$

Choosing β_i such that $d_k^T G d_i = 0, i = 0, 1, \cdots, k-1$, and noticing from Theorem 4.1.3 that

$$g_k^T d_i = 0,\ g_k^T g_i = 0,\ i = 0, 1, \cdots, k-1, \tag{4.2.10}$$

it follows from multiplying (4.2.9) by $d_j^T G$, $(j = 0, 1, \cdots, k-1)$ that

$$\beta_j = \frac{g_k^T G d_j}{d_j^T G d_j} = \frac{g_k^T (g_{j+1} - g_j)}{d_j^T (g_{j+1} - g_j)},\ j = 0, 1, \cdots, k-1. \tag{4.2.11}$$

Then

$$\beta_j = 0,\ j = 0, 1, \cdots, k-2, \tag{4.2.12}$$

$$\beta_{k-1} = \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T (g_k - g_{k-1})} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}}. \tag{4.2.13}$$

The above derivation establishes the iterative scheme of the conjugate gradient method:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{4.2.14}$$

$$d_k = -g_k + \beta_{k-1} d_{k-1}, \tag{4.2.15}$$

where

$$\beta_{k-1} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}},\ \text{(F-R Formula)} \tag{4.2.16}$$

and α_k is an exact step size, in particular, for the quadratic case,

$$\alpha_k = \frac{-g_k^T d_k}{d_k^T G d_k}. \tag{4.2.17}$$

The other famous formulas of β_k are as follows:

$$\beta_{k-1} = \frac{g_k^T(g_k - g_{k-1})}{d_{k-1}^T(g_k - g_{k-1})}, \text{ (H-S or C-W Formula)} \quad (4.2.18)$$

$$\beta_{k-1} = \frac{g_k^T(g_k - g_{k-1})}{g_{k-1}^T g_{k-1}}, \text{ (PRP Formula)} \quad (4.2.19)$$

$$\beta_{k-1} = -\frac{g_k^T g_k}{d_{k-1}^T g_{k-1}}, \text{ (Dixon Formula)} \quad (4.2.20)$$

$$\beta_{k-1} = \frac{g_k^T g_k}{d_{k-1}^T(g_k - g_{k-1})}, \text{ (D-Y Formula)} \quad (4.2.21)$$

where F-R, H-S (or C-W), PRP, Dixon and D-Y formula refer respectively Fletcher-Reeves formula, Hestenes-Stiefel (or Crowder-Wolfe) formula, Polak-Ribière-Polyak formula, Dixon formula and Dai-Yuan Formula. It is easy to see that these formulas are equivalent in the sense that all yield the same search directions when used in minimizing a quadratic function with positive definite Hessian matrix. However, for a general nonlinear function with inexact line search, their behavior is markedly different. Some descriptions will be given later in this subsection.

From (4.2.14)-(4.2.16), we can see that the conjugate gradient method is only a little more complex than the steepest descent method, but it has quadratic termination property and need not compute the Hessian or its approximation matrix. Besides, we will learn below that the conjugate gradient method has global convergence and n-step local quadratic convergence. Hence this method is very attractive especially for large scale optimization problems.

The following theorem includes the main properties of a conjugate gradient method.

Theorem 4.2.1 *(Property theorem of conjugate gradient method) For positive definite quadratic function (4.2.1), the conjugate gradient method (4.2.14)-(4.2.16) with exact line searches terminates after $m \leq n$ steps, and the following properties hold for all i, $(0 \leq i \leq m)$,*

$$d_i^T G d_j = 0, \ j = 0, 1, \cdots, i-1, \quad (4.2.22)$$

$$g_i^T g_j = 0, \ j = 0, 1, \cdots, i-1, \quad (4.2.23)$$

$$d_i^T g_i = -g_i^T g_i, \quad (4.2.24)$$

$$[g_0, g_1, \cdots, g_i] = [g_0, Gg_0, \cdots, G^i g_0], \tag{4.2.25}$$

$$[d_0, d_1, \cdots, d_i] = [g_0, Gg_0, \cdots, G^i g_0]. \tag{4.2.26}$$

where m is the number of distinct eigenvalues of G.

Proof. We prove (4.2.22)–(4.2.24) by induction. For $i = 1$, it is trivial. Suppose (4.2.22)–(4.2.24) hold for some $i < m$. We show that they also hold for $i + 1$.

For quadratic function (4.2.1), we have obviously

$$g_{i+1} = g_i + G(x_{i+1} - x_i) = g_i + \alpha_i G d_i. \tag{4.2.27}$$

From (4.2.17), α_i can be written as

$$\alpha_i = \frac{g_i^T g_i}{d_i^T G d_i} \neq 0. \tag{4.2.28}$$

Using (4.2.27) and (4.2.15) gives

$$\begin{aligned} g_{i+1}^T g_j &= g_i^T g_j + \alpha_i d_i^T G g_j \\ &= g_i^T g_j - \alpha_i d_i^T G(d_j - \beta_{j-1} d_{j-1}). \end{aligned} \tag{4.2.29}$$

When $j = i$, (4.2.29) becomes

$$g_{i+1}^T g_i = g_i^T g_i - \frac{g_i^T g_i}{d_i^T G d_i} d_i^T G d_i = 0.$$

When $j < i$, (4.2.29) is zero directly by induction hypothesis. So, (4.2.23) follows.

Now, from (4.2.15) and (4.2.27), it follows that

$$\begin{aligned} d_{i+1}^T G d_j &= -g_{i+1}^T G d_j + \beta_i d_i^T G d_j \\ &= g_{i+1}^T (g_j - g_{j+1})/\alpha_j + \beta_i d_i^T G d_j. \end{aligned} \tag{4.2.30}$$

When $j = i$, it follows from (4.2.30), (4.2.23), (4.2.28) and (4.2.16) that

$$d_{i+1}^T G d_i = -\frac{g_{i+1}^T g_{i+1}}{g_i^T g_i} d_i^T G d_i + \frac{g_{i+1}^T g_{i+1}}{g_i^T g_i} d_i^T G d_i = 0.$$

When $j < i$, (4.2.30) is also zero from induction hypothesis. Then (4.2.22) follows.

Also, from (4.2.15) and the exact line search, we have

$$\begin{aligned} d_{i+1}^T g_{i+1} &= -g_{i+1}^T g_{i+1} + \beta_i d_i^T g_{i+1} \\ &= -g_{i+1}^T g_{i+1}, \end{aligned}$$

which shows (4.2.24) holds for $i+1$.

Finally, we show (4.2.25) and (4.2.26) by induction. It is trivial for $i = 0$. Now suppose they hold for some i, and we prove that they hold also for $i+1$.

From the induction hypothesis, both g_i and Gd_i belong to

$$[g_0, Gg_0, \cdots, G^i g_0, G^{i+1} g_0].$$

Then it follows from (4.2.27) that $g_{i+1} \in [g_0, Gg_0, \cdots, G^{i+1} g_0]$. Furthermore, we need to show

$$g_{i+1} \notin [g_0, Gg_0, \cdots, G^i g_0] = [d_0, \cdots, d_i].$$

In fact, since vectors $d_0, \cdots, d_i$ are conjugate, it follows from Theorem 4.1.3 that $g_{i+1} \perp [d_0, \cdots, d_i]$. If $g_{i+1} \in [g_0, Gg_0, \cdots, G^i g_0] = [d_0, \cdots, d_i]$, then it results in $g_{i+1} = 0$. This is a contradiction. Therefore (4.2.25) follows.

Similarly, by (4.2.15) and induction hypothesis, we can get (4.2.26). □

In this theorem, (4.2.22)–(4.2.24) represent respectively conjugacy of directions, orthogonality of gradients, and descent condition. (4.2.25)–(4.2.26) give some relations between direction vectors and gradients. Usually, The subspace $[g_0, Gg_0, \cdots, G^i g_0]$ is called the Krylov subspace.

Recall please the convergence rate (3.1.7), (3.1.8), and (3.1.9) of the steepest descent method for quadratic functions in Theorem 3.1.5. Similarly, for quadratic functions, we can also obtain the following facts for the conjugate gradient method:

Fact 1:

$$\frac{\|x_k - x^*\|_G}{\|x_0 - x^*\|_G} \leq \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^k, \tag{4.2.31}$$

where κ is the spectral condition number of G.

Fact 2: starting from x_1, the iterate x_{k+2} of the conjugate gradient method after $k+1$ iterations satisfies

$$E(x_{k+2}) \leq \left(\frac{\lambda_{k+1} - \lambda_n}{\lambda_{k+1} + \lambda_n}\right)^2 E(x_1) = \left(\frac{1 - \lambda_n/\lambda_{k+1}}{1 + \lambda_n/\lambda_{k+1}}\right)^2 E(x_1), \tag{4.2.32}$$

where $E(x)$ is defined by

$$E(x) = \frac{1}{2}(x - x^*)^T G(x - x^*),$$

and the eigenvalues λ_i of G satisfy

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k \geq \lambda_{k+1} \geq \cdots \geq \lambda_n > 0.$$

Clearly, after the first iteration ($k = 0$), the obtained iterate x_2 satisfies

$$E(x_2) \leq \left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}\right)^2 E(x_1)$$

which is the same as convergence rate (3.1.8) of the steepest descent method, this is because, at the first iteration, the direction of the conjugate gradient method just is the steepest descent direction. However, after the second iteration ($k = 1$), we have

$$E(x_3) \leq \left(\frac{\lambda_2 - \lambda_n}{\lambda_2 + \lambda_n}\right)^2 E(x_1).$$

At this time, the influence of the largest eigenvalue λ_1 has been removed. The formula (4.2.32) indicates that after each additional iteration of the conjugate gradient method, the influence of one bigger eigenvalue will be removed.

Next, we would like to discuss restart strategy. Since the direction d_k after n steps is no longer conjugate for general non-quadratic functions, it is suitable to reset periodically d_k to the steepest descent direction, i.e., set

$$d_{cn} = -g_{cn},\ c = 1, 2, \cdots.$$

This strategy is called restart. With this strategy, the resultant x_{n-1} is nearer to x^* than x_0. Especially, when the iterate enters from an area in which non-quadratic behavior is strong into a neighborhood in which a quadratic model function approximates $f(x)$ well, the restart method is able to converge rapidly. For large scale problems, restart strategy will be used more frequently, for example, every k iterations restart, where $k < n$, even $k \ll n$.

Notice that the restart conjugate gradient method permits inexact line search. However, some control measures are needed so that the resultant direction is descending. In fact, we have

$$g_k^T d_k = -g_k^T g_k + \beta_{k-1} g_k^T d_{k-1}. \tag{4.2.33}$$

If exact line search was used in the previous iteration, then $g_k^T d_{k-1} = 0$, and hence $g_k^T d_k = -g_k^T g_k < 0$ which guarantees that d_k is a descent direction. However, if inexact line search was used in the previous iteration, the quantity $\beta_{k-1} g_k^T d_{k-1}$ may be positive and larger than $-g_k^T g_k$, consequently $-g_k^T g_k + \beta_{k-1} g_k^T d_{k-1}$ is possibly larger than zero. In this case d_k will not be a descent direction. A typical remedy for such an eventuality is to restart the algorithm with d_k as the steepest descent direction $-g_k$. However, frequently setting d_k to the steepest descent direction will lessen the efficiency of the algorithm, and make the behavior of the algorithm incline to a steepest descent method. This situation requires care. The following control measure can be used to overcome this difficulty.

Let $\bar{g}_{k+1}, \bar{d}_{k+1}$ and $\bar{\beta}_k$ denote the computed values of g_{k+1}, d_{k+1} and β_k at $x_k + \alpha^j d_k$ respectively, where $\{\alpha^j\}$ is a test step size sequence generated from a step size algorithm. If

$$-\bar{g}_{k+1}^T \bar{d}_{k+1} \geq \sigma \|\bar{g}_{k+1}\|_2 \|\bar{d}_{k+1}\|_2, \tag{4.2.34}$$

where σ is a small positive number, then α^j is accepted as α_k. If (4.2.34) is not satisfied at any trial points, we will use exact line search to produce α_k.

The following algorithm is a restart conjugate gradient method with exact line search.

Algorithm 4.2.2 *(Restart F-R Conjugate Gradient Method)*

Step 0. *Given x_0, $\epsilon > 0$.*

Step 1. *Set $k = 0$. Compute $g_0 = g(x_0)$.*

Step 2. *If $\|g_0\| \leq \epsilon$, stop; otherwise, set $d_0 = -g_0$.*

Step 3. *Compute step size α_k, such that*

$$f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} \{f(x_k + \alpha d_k)\}.$$

Step 4. *Set $x_{k+1} = x_k + \alpha_k d_k, k := k + 1$.*

Step 5. *Compute $g_k = g(x_k)$. If $\|g_k\| \leq \epsilon$, stop; otherwise go to Step 6.*

Step 6. *If $k = n$, set $x_0 = x_k$, and go to Step 1; otherwise, go to Step 7.*

Step 7. *Compute* $\beta = g_k^T g_k / g_{k-1}^T g_{k-1}, d_k = -g_k + \beta d_{k-1}$.

Step 8. *If* $d_k^T g_k > 0$, *set* $x_0 = x_k$, *and go to Step 1; otherwise go to Step 3.* □

(4.2.18)–(4.2.20) are common formulas of the conjugate gradient method. The Fletcher-Reeves formula (4.2.16) is the first presented in 1964 for solving optimization problems and now is the most widely used in practice. However, in general, this formula does not have the descent property and is often used in conjunction with exact line search. Dixon's formula (4.2.20) has descent property. If we employ inexact line search

$$|g_{k+1}^T d_k| \leq -\sigma g_k^T d_k, \ 0 < \sigma < 1,$$

Dixon's formula satisfies

$$d_k^T g_k < 0, \text{ if } g_k \neq 0.$$

The Polak-Ribiere-Polyak (PRP) formula (4.2.19) has a characteristic that it can restart automatically. When the algorithm goes slowly and $g_{k+1} \approx g_k$, PRP formula will produce $\beta_k \approx 0$ and thus $d_{k+1} \approx -g_{k+1}$. This indicates that the algorithm has a tendency of restarting automatically, so that it can overcome some shortcomings of going forward slowly. Various numerical experiments show that PRP formula is more robust and efficient than other existing formulas for solving optimization problems.

4.2.2 Beale's Three-Term Conjugate Gradient Method

Beale [10] considered the three-term conjugate gradient method. The idea is as follows. When frequently periodic restarts with the steepest descent direction are used, the reduction at the restart iteration is often poor compared with the reduction that would have occurred without restarting. However, if the restart direction is taken as an arbitrary vector, the required conjugacy relations may not hold. Now we consider restarting at x_t, and take the direction d_t generated by the algorithm as the restarting direction to begin the new cycle, and require the constructed sequence of directions to satisfy the conjugacy.

Set

$$d_{t+1} = -g_{t+1} + \beta_t d_t, \tag{4.2.35}$$

$$d_k = -g_k + \gamma_{k-1} d_t + \beta_{t+1} d_{t+1} + \cdots + \beta_{k-1} d_{k-1}, \tag{4.2.36}$$

where $n + t - 1 \geq k \geq t + 2$. Similar to the derivation of the traditional conjugate gradient method, by means of conjugacy between d_{t+1} and d_t, d_k and $d_t, d_{t+1}, \cdots, d_{k-1}$, we can get the following relation:

$$\beta_{k-1} = \frac{g_k^T G d_{k-1}}{d_{k-1}^T G d_{k-1}},\ \gamma_{k-1} = \frac{g_k^T G d_t}{d_t^T G d_t},$$

$$\beta_j = 0,\ j = t+1, \cdots, k-2.$$

Then (4.2.36) can be reduced as

$$d_k = -g_k + \beta_{k-1} d_{k-1} + \gamma_{k-1} d_t, \tag{4.2.37}$$

where

$$\beta_{k-1} = \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T (g_k - g_{k-1})}, \tag{4.2.38}$$

$$\gamma_{k-1} = \begin{cases} 0, & \text{if } k = t+1; \\ \frac{g_k^T (g_{t+1} - g_t)}{d_t^T (g_{t+1} - g_t)}, & \text{if } k > t+1. \end{cases} \tag{4.2.39}$$

Note that β_{k-1} in (4.2.38) can be represented as any formula in (4.2.18)-(4.2.21), for example,

$$\beta_{k-1} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}}$$

which is F-R formula.

Note also that in Beale's three-term formula (4.2.37), d_k may not be a descent direction, even if exact line searches are made. In order to make d_k be sufficient downhill and make two consecutive gradients not be far from orthogonal, we may impose some control measures as follows,

$$-g_k^T d_k \geq \sigma \|g_k\| \|d_k\|, \tag{4.2.40}$$

where σ is a small positive number, and

$$|g_{k-1}^T g_k| < 0.2 \|g_k\|^2. \tag{4.2.41}$$

Since the iterate x_k generated from (4.2.37)-(4.2.39) is a minimizer of the linear manifold

$$\begin{aligned} \mathcal{B}_{k-1} &= x_t + [d_t, d_{t+1}, \cdots, d_{k-1}] \\ &= x_t + [d_t, g_{t+1}, \cdots, g_{k-1}], \end{aligned}$$

and hence

$$g_k \perp [d_t, d_{t+1}, \cdots, d_{k-1}]$$

and

$$g_k \perp [d_t, g_{t+1}, \cdots, g_{k-1}].$$

Below, we give Beale's three-term conjugate gradient algorithm.

Algorithm 4.2.3 *(Beale's three-term CG method)*

Step 1. *Given x_0, set $k = 0, t = 0$, evaluate $g_0 = g(x_0)$. If $\|g_0\| \leq \epsilon$, stop; otherwise set $d_0 = -g_0$.*

Step 2. *Compute α_k by exact line search.*

Step 3. *Set $x_{k+1} = x_k + \alpha_k d_k$, set $k := k + 1$, evaluate $g_k = g(x_k)$.*

Step 4. *If $\|g_k\| \leq \epsilon$, stop; otherwise go to Step 5.*

Step 5. *If both conditions*

$$|g_{k-1}^T g_k| \geq 0.2\|g_k\|^2$$

and

$$k - t \geq n - 1$$

do not hold, go to Step 7; otherwise go to Step 6.

Step 6. *Set $t = k - 1$.*

Step 7. *Compute d_k by (4.2.37)-(4.2.39).*

Step 8. *If $k > t + 1$, go to Step 9; otherwise go to Step 2.*

Step 9 *If*

$$-1.2\|g_k\|^2 \leq d_k^T g_k \leq -0.8\|g_k\|^2,$$

go to Step 2; otherwise go to Step 6. □

4.2.3 Preconditioned Conjugate Gradient Method

In the discussion above we have known that if the conjugate gradient method is applied to minimize the quadratic function

$$f(x) = \frac{1}{2}x^T Gx + b^T x + c, \tag{4.2.42}$$

where G is symmetric and positive definite, it computes the solution of the system

$$Gx = -b. \tag{4.2.43}$$

In this case, the algorithm is called the linear conjugate gradient method, and the notation r is used for the gradient vector $Gx_k + b$, which, in fact, is the residual of the system (4.2.43).

The linear conjugate gradient method is as follows: given x_0 and $r_0 = Gx_0 + b, \beta_{-1} = 0, d_{-1} = 0$, and each iteration includes the following steps for $k = 0, 1, \cdots$:

$$\begin{aligned} d_k &= -r_k + \beta_{k-1} d_{k-1}, \\ \alpha_k &= \frac{r_k^T r_k}{d_k^T G d_k}, \\ x_{k+1} &= x_k + \alpha_k d_k, \\ r_{k+1} &= r_k + \alpha_k G d_k, \\ \beta_k &= \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}. \end{aligned} \tag{4.2.44}$$

If exact arithmetic is used, the convergence of the linear conjugate gradient method will be achieved in $m(\leq n)$ iterations, where m is the number of distinct eigenvalues of G. If the eigenvalues of G are clustered into groups of approximately equal value, the method may converge very quickly. However, for general eigenvalue structure, due to rounding errors, considerably more than n iterations may be required. Hence, the convergence rate depends on the structure of eigenvalues of G and the condition number of G. If the original system is replaced by an equivalent system in which the conditioning of G is improved, then the convergence rate can be improved. This technique is called preconditioning.

Consider the transformation

$$x = C^{-1} z,$$

where C is a nonsingular matrix. The solution of $Gx = -b$ is equivalent to solving the linear system

$$C^{-T}GC^{-1}z = -C^{-T}b.$$

If we adequately choose C such that the condition number of $C^{-T}GC^{-1}$ is as small as possible, the convergence rate of the algorithm will be improved. Since $C^{-T}GC^{-1}$ is similar to $W^{-1}G$, where $W = C^TC$, it means that we should choose W such that the condition number of $W^{-1}G$ is as small as possible.

The preconditioned conjugate gradient method is as follows: given x_0, set $g_0 = Gx_0 + b$, and let $v_0 = W^{-1}g_0$ and $d_0 = -v_0$. For $k = 0, 1, \cdots$,

$$\alpha_k = \frac{g_k^T v_k}{d_k^T G d_k}, \tag{4.2.45}$$

$$x_{k+1} = x_k + \alpha_k d_k, \tag{4.2.46}$$

$$g_{k+1} = g_k + \alpha_k G d_k, \tag{4.2.47}$$

$$v_{k+1} = W^{-1}g_{k+1}, \tag{4.2.48}$$

$$\beta_k = \frac{g_{k+1}^T v_{k+1}}{g_k^T v_k}, \tag{4.2.49}$$

$$d_{k+1} = -v_{k+1} + \beta_k d_k. \tag{4.2.50}$$

The preconditioning matrix W can be defined in several ways. The simplest strategy is to choose W as the diagonal of G. In this case, the condition number of $W^{-1}G$ is bounded by $(1+\delta)/(1-\delta)$, where $\delta \ll 1$. The popular strategy for preconditioning is use of incomplete Cholesky factorization. The basic idea is as follows. Instead of computing the exact Cholesky factor L which satisfies $G = LL^T$, we compute an approximate factor $\tilde{L}$ which is more sparse than L, such that $G \approx \tilde{L}\tilde{L}^T$, and then choose $C = \tilde{L}^T$, and hence $W = \tilde{L}\tilde{L}^T$,

$$C^{-T}GC^{-1} = \tilde{L}^{-1}G\tilde{L}^{-T} \approx I$$

and

$$W^{-1}G = (\tilde{L}\tilde{L}^T)^{-1}G \approx I.$$

In this procedure, any fill-in during the sparse Cholesky factorization is discarded.

The other preconditioning matrix can be obtained by performing a limited-memory quasi-Newton method. From the quasi-Newton method (see Chapter 5) the limited memory matrix M satisfies the quasi-Newton condition for

r ($r \ll n$) pairs of vectors $\{s_j, y_j\}$,

$$s_j = My_j,\ j = 1, \cdots, r,$$

where $s_j = x_{j+1} - x_j, y_j = g_{j+1} - g_j$. Since $Gs_j = y_j$, we have

$$s_j = MGs_j,$$

and the matrix MG has r unit eigenvalues with eigenvectors $\{s_j\}$. Therefore, M can be used as W^{-1}.

For the minimization of a non-quadratic function, the preconditioning matrix W is varied from iteration to iteration. In this case, we consider

$$x = C^{-1}z, \tag{4.2.51}$$

and the objective function is transformed as

$$f(x) = f(C^{-1}z) = \tilde{f}(z). \tag{4.2.52}$$

Set

$$z_k = Cx_k,\ \tilde{g}_k = \nabla\tilde{f}(z_k) = C^{-T}\nabla f(x_k) = C^{-T}g_k,$$

then

$$\tilde{d}_k = Cd_k,\ \tilde{s}_k = Cs_k,\ \tilde{y}_k = C^{-T}y_k.$$

So, application of conjugate gradient method, for example (4.2.18), to $\tilde{f}(z)$ yields the direction

$$\begin{aligned} \tilde{d}_{k+1} &= -\tilde{g}_{k+1} + \frac{\tilde{g}_{k+1}^T(\tilde{g}_{k+1} - \tilde{g}_k)}{\tilde{d}_k^T(\tilde{g}_{k+1} - \tilde{g}_k)}\tilde{d}_k \\ &= -\left(I - \frac{\tilde{d}_k\tilde{y}_k^T}{\tilde{d}_k^T\tilde{y}_k}\right)\tilde{g}_{k+1}, \end{aligned} \tag{4.2.53}$$

and hence

$$\begin{aligned} d_{k+1} &= -\left(I - \frac{d_k y_k^T}{d_k^T y_k}\right)W^{-1}g_{k+1}, \\ &\triangleq -P_{k+1}g_{k+1} \end{aligned} \tag{4.2.54}$$

which is the formula of the preconditioned conjugate gradient method, where $W = C^TC$. Similarly, we can obtain

$$d_{k+1} = -\left[I - \frac{1}{y_k^T s_k}(y_k s_k^T + s_k y_k^T) + \left(1 + \frac{y_k^T y_k}{y_k^T s_k}\right)\frac{s_k s_k^T}{y_k^T s_k}\right]W^{-1}g_{k+1} \tag{4.2.55}$$

which is the preconditioned conjugate gradient method in BFGS formula without memory.

In general, the preconditioning matrix is varied with different problems. There is not a general-purpose formula for preconditioners.

4.3 Convergence of Conjugate Gradient Methods

As for the convergence results of the conjugate gradient method for minimizing a general non-quadratic function, there have been various results. In this section, we introduce global convergence results of conjugate gradient methods due to Zoutendijk [385], Polyak [255] and Al-Baali [2] etc., and also give in brief the outline of local convergence rates obtained by Cohen [61], and McCormick and Ritter [205].

4.3.1 Global Convergence of Conjugate Gradient Methods

This subsection is divided into two parts. The first part discusses the global convergence of conjugate gradient methods with exact line search, and consists of three theorems which state respectively global convergence of Fletcher-Reeves (F-R) conjugate gradient method, Crowder-Wolfe (C-W) conjugate gradient method, and Polak-Ribière-Polyak (PRP) conjugate gradient method. The second part discusses the global convergence of F-R conjugate gradient method with inexact line search.

Now, we start the discussion by proving the global convergence result of F-R method in the case of exact line search.

Theorem 4.3.1 *(Global convergence of F-R conjugate gradient method)*

Suppose that $f : R^n \to R$ is continuously differentiable on a bounded level set $L = \{x \in R^n \mid f(x) \leq f(x_0)\}$, and that F-R conjugate gradient method is implemented with exact line search. Then the produced sequence $\{x_k\}$ has at least one accumulation point which is a stationary point, i.e.,

(1) when $\{x_k\}$ is a finite sequence, then the final point x^* is a stationary point of f;

(2) when $\{x_k\}$ is an infinite sequence, it has limit point, and any limit point is a stationary point.

Proof. (1) When $\{x_k\}$ is finite, from the termination condition, it follows that the final point x^* satisfies $\nabla f(x^*) = 0$, and hence x^* is a stationary point of f.

(2) When $\{x_k\}$ is infinite, we have $\nabla f(x_k) \neq 0, \forall k$. Noting that $d_k = -g_k + \beta_{k-1} d_{k-1}$ and $g_k^T d_{k-1} = 0$ by exact line search, we have

$$g_k^T d_k = -\|g_k\|^2 + \beta_{k-1} g_k^T d_{k-1} = -\|g_k\|^2 < 0, \tag{4.3.1}$$

which means that d_k is a descent direction, $\{f(x_k)\}$ is a monotone descent sequence, and thus $\{x_k\} \subset L$. Therefore $\{x_k\}$ is a bounded sequence and must have a limit point.

Let x^* be a limit point of $\{x_k\}$. Then there is a subsequence $\{x_k\}_{K_1}$ converging to x^*, where K_1 is an index set of a subsequence of $\{x_k\}$. Since $\{x_k\}_{K_1} \subset \{x_k\}$, $\{f(x_k)\}_{K_1} \subset \{f(x_k)\}$. It follows from the continuity of f that for $k \in K_1$,

$$f(x^*) = f(\lim_{k\to\infty} x_k) = \lim_{k\to\infty} f(x_k) = f^*. \tag{4.3.2}$$

Similarly, $\{x_{k+1}\}$ is also a bounded sequence. Hence there exists a subsequence $\{x_{k+1}\}_{K_2}$ converging to $\bar{x}^*$, where K_2 is an index set of a subsequence of $\{x_{k+1}\}$. In this case,

$$f(\bar{x}^*) = f(\lim_{k\to\infty} x_{k+1}) = \lim_{k\to\infty} f(x_{k+1}) = f^*. \tag{4.3.3}$$

Then

$$f(\bar{x}^*) = f(x^*) = f^*. \tag{4.3.4}$$

Now we prove $\nabla f(x^*) = 0$ by contradiction. Suppose that $\nabla f(x^*) \neq 0$, then, for α sufficiently small, we have

$$f(x^* + \alpha d^*) < f(x^*). \tag{4.3.5}$$

Since

$$f(x_{k+1}) = f(x_k + \alpha_k d_k) \leq f(x_k + \alpha d_k),\ \forall \alpha > 0,$$

then for $k \in K_2$, passing to limit $k \to \infty$ and using (4.3.5), we get

$$f(\bar{x}^*) \leq f(x^* + \alpha d^*) < f(x^*), \tag{4.3.6}$$

which contradicts (4.3.4). This proves $\nabla f(x^*) = 0$, i.e., x^* is a stationary point of f. □

Similarly, we can state the global convergence of Crowder-Wolfe (C-W) restart conjugate gradient method with exact line search as follows.

Theorem 4.3.2 *(Global convergence of Crowder-Wolfe conjugate gradient method) Suppose that the level set* $L = \{x \in R^n \mid f(x) \leq f(x_0)\}$ *is bounded, and that* $\nabla f(x)$ *is Lipschitz continuous. Assume that Crowder-Wolfe conjugate gradient method is implemented with exact line search and restart strategy. Then the produced sequence* $\{x_k\}$ *has at least one accumulation point which is a stationary point.*

Proof. See Polyak [255]. □

As mentioned before, PRP method is more efficient than F-R method. We naturally hope PRP method has also the above property for a general non-quadratic function. Unfortunately, the above Theorem 4.3.1 is not true for PRP method (see Powell [270]). However, with stronger condition that f is uniformly convex, the PRP method is globally convergent. The following theorem states this result.

Theorem 4.3.3 *Let* $f(x)$ *be twice continuously differentiable and the level set* $L = \{x \in R^n \mid f(x) \leq f(x_0)\}$ *be bounded. Suppose that there is a constant* $m > 0$ *such that for* $x \in L$,

$$m\|y\|^2 \leq y^T \nabla^2 f(x) y, \ \forall y \in R^n. \tag{4.3.7}$$

Then the sequence $\{x_k\}$ *generated by PRP method with exact line search converges to the unique minimizer* x^* *of* f.

Proof. From Theorem 2.2.4, we know that it is enough to prove that (2.2.13) holds, that is, there exists a constant $\rho > 0$ such that

$$-g_k^T d_k \geq \rho \|g_k\| \|d_k\|, \tag{4.3.8}$$

which means

$$\cos\theta_k \geq \rho > 0.$$

Then, from Theorem 2.2.4, we have $g_k \to 0$ and $g(x^*) = 0$. From (4.3.7), it follows that $\{x_k\} \to x^*$ which is a unique minimizer.

By using $g_k^T d_{k-1} = 0$ and (4.2.15), we have

$$g_k^T d_k = -\|g_k\|^2.$$

Then (4.3.8) is equivalent to

$$\frac{\|g_k\|}{\|d_k\|} \geq \rho. \tag{4.3.9}$$

From (4.2.17) and (4.2.15), it follows that

$$\alpha_{k-1} = -\frac{g_{k-1}^T d_{k-1}}{d_{k-1}^T G_{k-1} d_{k-1}} = \frac{\|g_{k-1}\|^2}{d_{k-1}^T G_{k-1} d_{k-1}}, \tag{4.3.10}$$

where

$$G_{k-1} = \int_0^1 G(x_{k-1} + t\alpha_{k-1} d_{k-1}) \mathrm{d}t. \tag{4.3.11}$$

By (4.3.11), the integral form of the mean-value theorem is

$$g_k - g_{k-1} = g(x_{k-1} + \alpha_{k-1} d_{k-1}) - g(x_{k-1}) = \alpha_{k-1} G_{k-1} d_{k-1}. \tag{4.3.12}$$

Then, by (4.3.11) and (4.3.10), (4.2.19) becomes

$$\begin{aligned} \beta_{k-1} &= \frac{g_k^T (g_k - g_{k-1})}{g_{k-1}^T g_{k-1}} = \alpha_{k-1} \frac{g_k^T G_{k-1} d_{k-1}}{\|g_{k-1}\|^2} \\ &= \frac{g_k^T G_{k-1} d_{k-1}}{d_{k-1}^T G_{k-1} d_{k-1}}. \end{aligned} \tag{4.3.13}$$

Since the level set L is bounded, there is a constant $M > 0$, such that

$$y^T G(x) y \le M\|y\|^2, \ x \in L, \forall y \in R^n. \tag{4.3.14}$$

Then, by (4.3.13), (4.3.14) and (4.3.7), we have

$$|\beta_{k-1}| \le \frac{\|g_k\| \|G_{k-1} d_{k-1}\|}{m\|d_{k-1}\|^2} \le \frac{M}{m} \frac{\|g_k\|}{\|d_{k-1}\|}. \tag{4.3.15}$$

Therefore

$$\begin{aligned} \|d_k\| &\le \|g_k\| + |\beta_{k-1}| \|d_{k-1}\| \\ &\le \|g_k\| + \frac{M}{m}\|g_k\| \\ &= \left(1 + \frac{M}{m}\right) \|g_k\|, \end{aligned} \tag{4.3.16}$$

which gives

$$\frac{\|g_k\|}{\|d_k\|} \ge \left(1 + \frac{M}{m}\right)^{-1}. \tag{4.3.17}$$

The above inequality shows that (4.3.9) holds. We complete the proof. □

Next, we discuss the case of inexact line search. Al-Baali [2] studied the F-R conjugate gradient method with strong Wolfe-Powell rule (2.5.3) and (2.5.9), and proved the global convergence. The following theorem indicates that, in the inexact case, the search direction d_k satisfies descent property: $g_k^T d_k < 0$.

Theorem 4.3.4 *If, for all k, α_k are determined by strong Wolfe-Powell rule (2.5.3) and (2.5.9), then for F-R-CG method, the inequality*

$$-\sum_{j=0}^{k} \sigma^j \leq \frac{g_k^T d_k}{\|g_k\|^2} \leq -2 + \sum_{j=0}^{k} \sigma^j \tag{4.3.18}$$

holds for all k, and hence the descent property

$$g_k^T d_k < 0, \ \forall k \tag{4.3.19}$$

holds, as long as $g_k \neq 0$.

Proof. The proof is by induction. For $k = 0$, $d_0 = -g_0, \sigma^0 = 1$, hence (4.3.18) and (4.3.19) hold for $k = 0$.

Now we suppose that (4.3.18) and (4.3.19) hold for any $k \geq 0$. By (4.2.15) and (4.2.16), we have

$$\frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} = -1 + \frac{g_{k+1}^T d_k}{\|g_k\|^2}. \tag{4.3.20}$$

Using (2.5.9) and induction assumption (4.3.19) yields

$$-1 + \sigma \frac{g_k^T d_k}{\|g_k\|^2} \leq \frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} \leq -1 - \sigma \frac{g_k^T d_k}{\|g_k\|^2}. \tag{4.3.21}$$

Also, by induction assumption (4.3.18), we have

$$\begin{aligned} -\sum_{j=0}^{k+1} \sigma^j &= -1 - \sigma \sum_{j=0}^{k} \sigma^j \leq \frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} \\ &\leq -1 + \sigma \sum_{j=0}^{k} \sigma^j = -2 + \sum_{j=0}^{k+1} \sigma^j. \end{aligned}$$

Then, (4.3.18) holds for $k + 1$.

Since

$$\frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} \leq -2 + \sum_{j=0}^{k+1} \sigma^j \tag{4.3.22}$$

and

$$\sum_{j=0}^{k+1} \sigma^j < \sum_{j=0}^{\infty} \sigma^j = \frac{1}{1-\sigma}, \tag{4.3.23}$$

where $\sigma \in (0,1)$, it follows from $1-\sigma > \frac{1}{2}$ that $-2 + \sum_{j=0}^{k+1} \sigma^j < 0$. Hence, from (4.3.22), we obtain $g_{k+1}^T d_{k+1} < 0$. We complete the proof by induction. □

Now, we are in a position to prove the global convergence of F-R-CG algorithm with inexact line search.

Theorem 4.3.5 *Let f be twice continuously differentiable, and the level set $L = \{x \in R^n \mid f(x) \leq f(x_0)\}$ be bounded. Suppose that the steplength α_k is determined by strong Wolfe-Powell rule (2.5.3) and (2.5.9), where $0 < \rho < \sigma < \frac{1}{2}$. Then the sequence $\{x_k\}$ generated by F-R-CG method is globally convergent, i.e.,*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{4.3.24}$$

Proof. By (2.5.9), (4.3.18) and (4.3.23), we have

$$|g_k^T d_{k-1}| \leq -\sigma g_{k-1}^T d_{k-1} \leq \sigma \sum_{j=0}^{k-1} \sigma^j \|g_{k-1}\|^2 \leq \frac{\sigma}{1-\sigma} \|g_{k-1}\|^2. \tag{4.3.25}$$

Also, by (4.2.15), (4.3.25) and (4.2.16), we obtain

$$\begin{aligned} \|d_k\|^2 &= \|g_k\|^2 - 2\beta_{k-1} g_k^T d_{k-1} + \beta_{k-1}^2 \|d_{k-1}\|^2 \\ &\leq \|g_k\|^2 + \frac{2\sigma}{1-\sigma} \|g_k\|^2 + \beta_{k-1}^2 \|d_{k-1}\|^2 \\ &= \left(\frac{1+\sigma}{1-\sigma}\right) \|g_k\|^2 + \beta_{k-1}^2 \|d_{k-1}\|^2. \end{aligned} \tag{4.3.26}$$

By applying this relation repeatedly, it follows that

$$\|d_k\|^2 \leq \left(\frac{1+\sigma}{1-\sigma}\right) \|g_k\|^4 \left(\sum_{j=0}^{k} \|g_j\|^{-2}\right), \tag{4.3.27}$$

where we used the facts that

$$\beta_k^2\beta_{k-1}^2\cdots\beta_{k-i}^2 = \frac{\|g_k\|^2}{\|g_{k-i-1}\|^2}.$$

Now we prove (4.3.24) by contradiction. It assumes that (4.3.24) does not hold, then there exists a constant $\epsilon > 0$ such that

$$\|g_k\| \geq \epsilon > 0 \tag{4.3.28}$$

holds for all k sufficiently large. Since g_k is bounded above on the level set L, it follows from (4.3.27) that

$$\|d_k\|^2 \leq c_1 k, \tag{4.3.29}$$

where c_1 is a positive constant. From (4.3.18) and (4.3.23), we have

$$\begin{aligned}\cos\theta_k &= -\frac{g_k^T d_k}{\|g_k\|\|d_k\|} \geq \left(2 - \sum_{j=0}^{k}\sigma^j\right)\frac{\|g_k\|}{\|d_k\|} \\ &\geq \left(\frac{1-2\sigma}{1-\sigma}\right)\frac{\|g_k\|}{\|d_k\|}.\end{aligned} \tag{4.3.30}$$

Since $\sigma < \frac{1}{2}$, substituting (4.3.29) and (4.3.28) into (4.3.30) gives

$$\sum_k \cos^2\theta_k \geq \left(\frac{1-2\sigma}{1-\sigma}\right)^2 \sum_k \frac{\|g_k\|^2}{\|d_k\|^2} \geq c_2 \sum_k \frac{1}{k}, \tag{4.3.31}$$

where c_2 is a positive constant. Therefore, the series $\sum_k \cos^2\theta_k$ is divergent.

Let M be an upper bound of $\|G(x)\|$ on the level set L, then

$$g_{k+1}^T d_k = (g_k + \alpha_k G(x_k)d_k)^T d_k \leq g_k^T d_k + \alpha_k M\|d_k\|^2.$$

By using (2.5.9), i.e., $\sigma g_k^T d_k \leq g_{k+1}^T d_k \leq -\sigma g_k^T d_k$, we obtain

$$\alpha_k \geq -\frac{1-\sigma}{M\|d_k\|^2} g_k^T d_k. \tag{4.3.32}$$

Substituting α_k of (4.3.32) into (2.5.3) gives

$$\begin{aligned} f_{k+1} &\leq f_k - \frac{(1-\sigma)\rho}{M}\left(\frac{g_k^T d_k}{\|d_k\|}\right)^2 \\ &= f_k - c_3\|g_k\|^2\cos^2\theta_k,\end{aligned}$$

where $c_3 = \frac{(1-\sigma)\rho}{M} > 0$. Since $f(x)$ is bounded below, $\sum_k \|g_k\|^2 \cos^2\theta_k$ converges, which indicates that $\sum_k \cos^2\theta_k$ converges by use of (4.3.28). This fact contradicts (4.3.31). We complete the proof. □

In the above theorem, the conclusion is also true if, instead of f being twice continuously differentiable, the assumptions on f are changed: let f be continuously differentiable and bounded below, and ∇f be Lipschitz continuous.

To conclude the subsection, we give the global convergence of D-Y conjugate gradient method with Wolfe-Powell rule.

Theorem 4.3.6 *Let x_1 be a starting point, $f(x)$ be continuously differentiable and bounded below on the level set L, $\nabla f(x)$ satisfy the Lipschitz condition on L. Let α_k satisfy Wolfe-Powell rule (2.5.3) and (2.5.7). Then, for all k,*

$$g_k^T d_k < 0,$$

and further

$$\liminf_{k\to\infty} \|g_k\| = 0.$$

Proof. See Dai and Yuan [75]. □

4.3.2 Convergence Rate of Conjugate Gradient Methods

We have already seen that the conjugate gradient method has quadratic termination, that is, for a convex quadratic function, the conjugate gradient method with exact line search terminates after n iterations.

In (4.2.31) and (4.2.32), we give two formulas for convergence rate of conjugate gradient method, from which we have seen that, for a quadratic function, the rate of convergence of conjugate gradient methods is not worse than that of the steepest descent method; that is, it is not worse than linear. Furthermore, we can also have the following demonstration. For convenience, we assume

$$f(x) = \frac{1}{2}x^T G x, \tag{4.3.33}$$

where G is an $n \times n$ positive definite matrix. Clearly, the explicit expression of steplength is

$$\alpha_k = -\frac{d_k^T G x_k}{d_k^T G d_k} = -\frac{d_k^T g_k}{d_k^T G d_k}. \tag{4.3.34}$$

So, we can obtain

$$\begin{aligned} f(x_{k+1}) &= \frac{1}{2}x_{k+1}^T G x_{k+1} \\ &= \frac{1}{2}(x_k + \alpha_k d_k)^T G(x_k + \alpha_k d_k) \\ &= \frac{1}{2}x_k^T G x_k - \frac{1}{2}\frac{(g_k^T d_k)^2}{d_k^T G d_k}. \end{aligned} \tag{4.3.35}$$

In the case of the steepest descent (SD) method we have $d_k = -g_k$ and thus

$$f(x_{SD}^{k+1}) = \frac{1}{2}x_k^T G x_k - \frac{1}{2}\frac{\|g_k\|^4}{g_k^T G g_k}. \tag{4.3.36}$$

Whereas in the case of the conjugate gradient (CG) method, we have $d_k = -g_k + \beta_{k-1}d_{k-1}$ and thus

$$f(x_{CG}^{k+1}) = \frac{1}{2}x_k^T G x_k - \frac{1}{2}\frac{\|g_k\|^4}{d_k^T G d_k} \tag{4.3.37}$$

$$= f(x_k) - \frac{1}{2}\frac{\|g_k\|^4}{d_k^T G d_k}. \tag{4.3.38}$$

Since

$$\begin{aligned} d_k^T G d_k &= (-g_k + \beta_{k-1}d_{k-1})^T G(-g_k + \beta_{k-1}d_{k-1}) \\ &= g_k^T G g_k + \beta_{k-1}^2 d_{k-1} G d_{k-1} \\ &\leq g_k^T G g_k, \end{aligned}$$

it follows that

$$f(x_{CG}^{k+1}) \leq f(x_{SD}^{k+1}). \tag{4.3.39}$$

The above discussion indicates again that the conjugate gradient method reduces the value of f at least as much as the steepest descent method. Since the steepest descent method has a linear convergence rate, we conclude that conjugate gradient methods have convergence rates that are no worse than the linear rate. From (4.3.38) we also know that, for conjugate gradient methods, the objective value is strictly decreasing. Similarly, the result is true for the preconditioned conjugate gradient method and we have

$$f(x_{CG}^{k+1}) = f(x_k) - \frac{1}{2}\frac{(g_k^T v_k)^2}{d_k^T G d_k}, \tag{4.3.40}$$

where $v_k = W^{-1}g_k$.

Note that the conjugate gradient method with exact line search can find the minimizer of a convex quadratic function in at most n iterations, which corresponds to one step of Newton method. Hence we can say that if n iterations of the conjugate gradient method are regarded as a big iteration, the conjugate gradient method should have a similar convergence rate as Newton method. Cohen [61], Burmeister [39], and McCormick and Ritter [205] studied the n-step quadratic convergence rate. We now state this result without proof in the following theorem.

Assume that

(A1) $f : R^n \to R$ is three times continuously differentiable;

(A2) there exist constants $M > m > 0$ such that

$$m\|y\|^2 \leq y^T \nabla^2 f(x) y \leq M\|y\|^2, \ \forall y \in R^n, \ x \in L, \tag{4.3.41}$$

where L is a bounded level set.

Theorem 4.3.7 *Assume that the conditions (A1) and (A2) are satisfied, then the sequence $\{x_k\}$ generated by PRP-CG and F-R-CG restart methods have n-step quadratic convergence rate, that is, there exists a constant $c > 0$, such that*

$$\limsup_{k\to\infty} \frac{\|x_{kr+n} - x^*\|}{\|x_{kr} - x^*\|^2} \leq c < \infty, \tag{4.3.42}$$

where r means that the methods restart per r iterations.

Further, Ritter [287] shows that the convergence rate is n-step super-quadratic, that is,

$$\|x_{k+n} - x^*\| = o(\|x_k - x^*\|^2). \tag{4.3.43}$$

The other results on convergence rate of conjugate gradient methods can consult Stoer [325].

Exercises

1. Let G be an $n \times n$ symmetric positive definite matrix, $p_1, p_2, \cdots, p_n$ be n linearly independent vectors. Define

$$d_1 = p_1,$$
$$d_{k+1} = p_{k+1} - \sum_{i=1}^{k} \frac{p_{k+1}^T G d_i}{d_i^T G d_i} d_i, \quad k = 1, 2, \cdots, n-1.$$

Prove that $\{d_k\}$ are G-conjugate.

2. Using F-R conjugate gradient method minimize the following functions:

(1) $f(x) = x_1^2 + 2x_2^2 - 2x_1x_2 + 2x_2 + 2$, the initial point $x^{(0)} = (0, 0)^T$.

(2) $f(x) = (x_1 - 1)^4 + (x_1 - x_2)^2$, the initial point $x^{(0)} = (0, 0)^T$.

3. Using respectively F-R conjugate gradient method and PRP conjugate gradient method minimize the Rosenbrock function in Appendix 1.1 and Extended Rosenbrock function in Appendix 1.2.

4. Try to prove respectively that $\{d_k\}$ generated by PRP-CG method and Dixon-CG method are conjugate.

5. Derive the Beale three-term conjugate gradient formula (4.2.38)–(4.2.39).

6. Let $f(x) = \frac{1}{2}x^TAx - b^Tx$, where A is an $n \times n$ symmetric positive definite matrix. Setting $x_{k+1} = x_k + \alpha_k d_k$ and $d_k = -r_k + \beta_k d_{k-1}$, prove

(1) the exact step size $\alpha_k = -\frac{r_k^T d_k}{d_k^T A d_k}$,

(2) $\beta_k = \frac{r_k^T A d_{k-1}}{d_{k-1}^T A d_{k-1}}$.

7. Using the linear conjugate gradient method minimize function $f(x) = \frac{1}{2}x^TAx - b^Tx$, where A is a Hilbert matrix $A = \left(\frac{1}{i+j-1}\right), b = (1, 1, \cdots, 1)^T$, the initial point $x^{(0)} = 0$. Try considering the cases of $n = 5, 10, 20$.

Chapter 5

Quasi-Newton Methods

5.1 Quasi-Newton Methods

We have seen that Newton's method $x_{k+1} = x_k - G_k^{-1} g_k$ is successful because it uses the Hessian which offers the useful curvature information. However, for various practical problems, the computing efforts of the Hessian matrices are very expensive, or the evaluation of the Hessian is difficult, even the Hessian is not available analytically. These lead to a class of methods that only uses the function values and the gradients of the objective function and that is closely related to Newton's method. Quasi-Newton method is such a class of methods which need not compute the Hessian, but generates a series of Hessian approximations, and at the same time maintains a fast rate of convergence.

Recall that in Chapter 3 the n-dimensional Newton's method $x_{k+1} = x_k - G_k^{-1} g_k$ comes from the one-dimensional Newton's method. Can we get any inspiration to the n-dimensional quasi-Newton method from the one-dimensional method? The answer is positive.

In Chapter 2, for quadratic interpolation with two points (2.4.6), we use interpolation condition (2.4.4) and obtain

$$\alpha_{k+1} = \alpha_k - \frac{\alpha_k - \alpha_{k-1}}{\phi'_k - \phi'_{k-1}} \phi'_k. \tag{5.1.1}$$

If we set

$$b_k = \frac{\phi'_k - \phi'_{k-1}}{\alpha_k - \alpha_{k-1}}, \tag{5.1.2}$$

then (5.1.1) can be written as

$$\alpha_{k+1} = \alpha_k - b_k^{-1}\phi_k' \tag{5.1.3}$$

which is also called the secant method. Comparing with Newton's form $\alpha_{k+1} = \alpha_k - [\phi_k'']^{-1}\phi_k'$ indicates that here b_k is used to approach ϕ_k'' without computing ϕ_k''. Also, the convergence rate of the secant method is $\frac{1+\sqrt{5}}{2} \approx 1.618$ (see Theorem 2.4.1) which is fast. Now we apply this idea to the n-dimensional quasi-Newton method.

5.1.1 Quasi-Newton Equation

Instead of computing the Hessian G_k, we would like to construct Hessian approximation, for example, B_k in the quasi-Newton method. We hope that the sequence $\{B_k\}$ possesses positive definiteness, has the direction $d_k = -B_k^{-1}g_k$ down, and behaves like Newton's method. In addition, it is also required that its computation is convenient. What conditions does such a sequence $\{B_k\}$ satisfy? How to form $\{B_k\}$? In this subsection, we first reply the first question, and in the subsequent subsections we shall discuss the formations of B_k.

Let $f : R^n \to R$ be twice continuously differentiable on an open set $D \subset R^n$. Let the quadratic approximation of f at x_{k+1} be

$$f(x) \approx f(x_{k+1}) + g_{k+1}^T(x - x_{k+1}) + \frac{1}{2}(x - x_{k+1})^T G_{k+1}(x - x_{k+1}), \tag{5.1.4}$$

where $g_{k+1} \triangleq \nabla f(x_{k+1})$ and $G_{k+1} \triangleq \nabla^2 f(x_{k+1})$. Finding the derivative yields

$$g(x) \approx g_{k+1} + G_{k+1}(x - x_{k+1}). \tag{5.1.5}$$

Setting $x = x_k, s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$, we get

$$G_{k+1}^{-1}y_k \approx s_k. \tag{5.1.6}$$

Clearly, it is true that (5.1.6) holds exactly with equality for quadratic function f with the Hessian G, i.e.,

$$s_k = G^{-1}y_k, \text{ or } y_k = Gs_k. \tag{5.1.7}$$

Now we ask the produced inverse Hessian approximations H_{k+1} in the quasi-Newton method to satisfy this relation, i.e.,

$$H_{k+1}y_k = s_k, \tag{5.1.8}$$

which is called the quasi-Newton equation or quasi-Newton condition, where

$$s_k = x_{k+1} - x_k,\ y_k = g_{k+1} - g_k. \tag{5.1.9}$$

In fact, if we consider the model function at x_{k+1},

$$m_{k+1}(x) = f(x_{k+1}) + g_{k+1}^T(x - x_{k+1}) + \frac{1}{2}(x - x_{k+1})^T B_{k+1}(x - x_{k+1}) \tag{5.1.10}$$

which satisfies the interpolation conditions

$$m_{k+1}(x_{k+1}) = f(x_{k+1}),\ \nabla m_{k+1}(x_{k+1}) = g_{k+1}, \tag{5.1.11}$$

where $B_{k+1} = H_{k+1}^{-1}$ is an approximation to the Hessian G_{k+1}. Instead of the interpolation condition $\nabla^2 m_{k+1}(x_{k+1}) = G_{k+1}$ in Newton's method, we ask the model (5.1.10) to satisfy

$$\nabla m_{k+1}(x_k) = g_k, \tag{5.1.12}$$

that is

$$g_k = g_{k+1} + B_{k+1}(x_k - x_{k+1}).$$

So we have

$$B_{k+1}(x_{k+1} - x_k) = g_{k+1} - g_k$$

or

$$B_{k+1}s_k = y_k \tag{5.1.13}$$

which is also the quasi-Newton equation expressed in Hessian approximation form.

Premultiplying (5.1.13) by s_k^T gives

$$s_k^T B_{k+1} s_k = s_k^T y_k.$$

It means that if

$$s_k^T y_k > 0, \tag{5.1.14}$$

the matrix B_{k+1} is positive definite. Usually, (5.1.14) is called the curvature condition.

The above discussion tells us that the key point of the quasi-Newton method is to produce H_{k+1} (or B_{k+1}) by use of some convenient methods such that the quasi-Newton equation (5.1.8) (or (5.1.13)) holds. In general, such an H_{k+1} will be produced by updating H_k into H_{k+1}, which is our topic in the subsequent subsections. Now we state a general quasi-Newton algorithm below.

Algorithm 5.1.1 *(A general quasi-Newton algorithm)*

Step 1. *Given* $x_0 \in R^n, H_0 \in R^{n\times n}, 0 \leq \epsilon < 1, k := 0.$

Step 2. *If* $\|g_k\| \leq \epsilon$, *stop.*

Step 3. *Compute*

$$d_k = -H_k g_k. \tag{5.1.15}$$

Step 4. *Find a step size* $\alpha_k > 0$ *by line search, and set* $x_{k+1} = x_k + \alpha_k d_k$.

Step 5. *Update* H_k *into* H_{k+1} *such that the quasi-Newton equation (5.1.8) holds.*

Step 6. $k := k + 1$ *and go to Step 2.* □

In the above algorithm, it is common to start the algorithm with $H_0 = I$, an identity matrix or set H_0 to be a finite-difference approximation to the inverse Hessian G_0^{-1}. If $H_0 = I$, the first iteration is just a steepest descent iteration. Sometimes, quasi-Newton method takes the form of Hessian approximation B_k. In this case, the Step 3 and Step 5 in Algorithm 5.1.1 have the following forms respectively.

Step 3*. Solve

$$B_k d = -g_k \text{ for } d_k. \tag{5.1.16}$$

Step 5*. Update B_k into B_{k+1} so that quasi-Newton equation (5.1.13) holds.

Next, we give some comparisons with Newton's method, which indicate that the quasi-Newton method is advantageous.

Comparison of quasi-Newton method vs Newton's method

quasi-Newton method	Newton's method
Only need the function values and gradients	Need the function values, gradients and Hessians
$\{H_k\}$ maintains positive definite for several updates	$\{G_k\}$ is not sure to be positive definite
Need $O(n^2)$ multiplications in each iteration	Need $O(n^3)$ multiplications in each iteration

As Newton's method is a steepest descent method under the norm $\|\cdot\|_{G_k}$, the quasi-Newton method is a steepest descent method under the norm $\|\cdot\|_{B_k}$, where B_k is the approximation of the Hessian G_k. In fact, d_k now is the solution of the minimization problem

$$\begin{aligned} \min \quad & g_k^T d \\ \text{s.t.} \quad & \|d\|_{B_k} \leq 1. \end{aligned} \tag{5.1.17}$$

From the inequality

$$(g_k^T d)^2 \leq (g_k^T B_k^{-1} g_k)(d^T B_k d),$$

it follows that when

$$d_k = -B_k^{-1} g_k = -H_k g_k,$$

$g_k^T d_k$ is the smallest.

By the way, since the metric matrices B_k are positive definite and always changed from iteration to iteration, the method is also called the variable metric method.

5.1.2 Symmetric Rank-One (SR1) Update

As we have seen, the key point of the quasi-Newton method is to generate H_{k+1} (or B_{k+1}) by means of the quasi-Newton equation. This subsection and the subsequent two subsections will discuss some typical and popular quasi-Newton updates. In this subsection we introduce a simple rank-one update that satisfies the quasi-Newton equation.

Let H_k be the inverse Hessian approximation of the k-th iteration. We try updating H_k into H_{k+1}, i.e.,

$$H_{k+1} = H_k + E_k, \tag{5.1.18}$$

where, usually, E_k is a matrix with lower rank. In the case of rank-one, we have

$$H_{k+1} = H_k + uv^T, \tag{5.1.19}$$

where $u, v \in R^n$. By quasi-Newton equation (5.1.8), we obtain

$$H_{k+1} y_k = (H_k + uv^T) y_k = s_k,$$

that is

$$(v^T y_k) u = s_k - H_k y_k. \tag{5.1.20}$$

This indicates that u must be in the direction of $s_k - H_k y_k$. Assume that $s_k - H_k y_k \neq 0$ (otherwise, H_k has satisfied the quasi-Newton equation) and that the vector v satisfies $v^T y_k \neq 0$, then it follows from (5.1.19) and (5.1.20) that

$$H_{k+1} = H_k + \frac{1}{v^T y_k}(s_k - H_k y_k)v^T. \tag{5.1.21}$$

Since the inverse Hessian approximation H_k is required to be symmetric, we can set simply $v = s_k - H_k y_k$ and get

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k} \tag{5.1.22}$$

which is called the symmetric rank-one update (SR1 update).

By the way, (5.1.21) is a general Broyden rank-one update in which, particularly, if $v = y_k$, (5.1.21) is called the Broyden rank-one update presented by Broyden (1965) for solving systems of nonlinear equations.

The distinct property of SR1 update is its natural quadratic termination, that is, for a quadratic function, it need not to do line search, but can be terminated within n steps, i.e., $H_n = G^{-1}$, where G is the Hessian of the quadratic function. This fact is proved by Theorem 5.1.2 below.

Theorem 5.1.2 *(Property Theorem of SR1 Update) Let $s_0, s_1, \cdots, s_{n-1}$ be linearly independent. Then, for a quadratic function with a positive definite Hessian, SR1 method terminates at $n+1$ steps, that is, $H_n = G^{-1}$.*

Proof. Let the Hessian G be positive definite. We can use

$$y_k = Gs_k, k = 0, 1, \cdots, n-1, \tag{5.1.23}$$

that is shared by all proofs on quadratic termination.

First, by induction, we prove the hereditary property

$$H_i y_j = s_j,\ j = 0, 1, \cdots, i-1. \tag{5.1.24}$$

For $i = 1$, it is trivial from (5.1.22). Now suppose it is true for $i \geq 1$; we will prove it holds for $i+1$.

From (5.1.22), we have

$$H_{i+1} y_j = H_i y_j + \frac{(s_i - H_i y_i)(s_i - H_i y_i)^T y_j}{(s_i - H_i y_i)^T y_i}. \tag{5.1.25}$$

When $j < i$, from the induction assumption and (5.1.23), we have

$$\begin{aligned}(s_i - H_i y_i)^T y_j &= s_i^T y_j - y_i^T H_i y_j \\ &= s_i^T y_j - y_i^T s_j \\ &= s_i^T G s_j - s_i^T G s_j \\ &= 0.\end{aligned}$$

Then

$$H_{i+1} y_j = H_i y_j = s_j,\ j < i.$$

When $j = i$, it is a direct consequence from (5.1.22) that

$$H_{i+1} y_i = s_i.$$

Therefore, (5.1.24) follows.

Furthermore, since

$$s_j = H_n y_j = H_n G s_j,\ j = 0, 1, \cdots, n-1$$

and $s_j\,(j = 0, 1, \cdots, n-1)$ are linearly independent, then $H_n G = I$, that is $H_n = G^{-1}$. □

It is not difficult to find that SR1 update has the following characteristics.

1. SR1 update possesses natural quadratic termination.

2. SR1 update satisfies the hereditary property: $H_i y_j = s_j, j < i$.

3. SR1 update does not retain the positive definiteness of H_k. If and only if $(s_k - H_k y_k)^T y_k > 0$, SR1 update retains positive definiteness. However, this condition is difficult to guarantee. The remedy is that SR1 update can be used in the trust region framework since the trust region method does not require positive definiteness of the Hessian approximations (see Chapter 6).

4. Sometimes, the denominator $(s_k - H_k y_k)^T y_k$ is very small or zero, which results in serious numerical difficulty or even the algorithm is broken. This disadvantage restricts its applications. So, it is a topic deserving research how to modify SR1 update such that it possesses not only natural quadratic termination but also positive definiteness. A special

skipping strategy to prevent the SR1 update from breaking down is as follows. We use (5.1.22) only if

$$|(s_i - H_i y_i)^T y_i| \geq r\|s_i - H_i y_i\|\|y_i\|, \tag{5.1.26}$$

where $r \in (0, 1)$; otherwise we set $H_{i+1} = H_i$.

5. The SR1 update has a good behavior that it continues to generate good Hessian approximations, which is stated in the following theorem.

Theorem 5.1.3 *Let f be twice continuously differentiable, and its Hessian be bounded and Lipschitz continuous in a neighborhood of a point x^*. Let $\{x_k\}$ be a sequence of iterates with $x_k \to x^*$. Suppose that the skipping rule (5.1.26) holds for all k, and the steps s_k are uniformly linearly independent. Then the matrix sequence $\{B_k\}$ generated by SR1 update satisfies*

$$\lim_{i\to\infty} \|H_i - [\nabla^2 f(x^*)]^{-1}\| = 0. \tag{5.1.27}$$

5.1.3 DFP Update

DFP update is another typical update which is a rank-two update, i.e., H_{k+1} is formed by adding to H_k two symmetric matrices, each of rank one. Let us consider the symmetric rank-two update

$$H_{k+1} = H_k + auu^T + bvv^T, \tag{5.1.28}$$

where $u, v \in R^n$, a and b are scalars to be determined. By the quasi-Newton equation (5.1.8),

$$H_k y_k + auu^T y_k + bvv^T y_k = s_k. \tag{5.1.29}$$

Clearly, u and v are not uniquely determined, but their obvious choices are

$$u = s_k,\ v = H_k y_k.$$

Then, from (5.1.29), we have

$$a = 1/u^T y_k = 1/s_k^T y_k,\ b = -1/v^T y_k = -1/y_k^T H_k y_k.$$

Therefore

$$H_{k+1} = H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k}. \tag{5.1.30}$$

The formula (5.1.30) is the first quasi-Newton update proposed originally by Davidon [79] and developed later by Fletcher and Powell [137]. Hence it is called DFP update.

Now we state the quasi-Newton algorithm with DFP update (in brief, DFP method) as follows.

Algorithm 5.1.4 *(DFP method)*

Initial Step: *Given $x_0 \in R^n$ an initial point, $H_0 \in R^{n\times n}$ a symmetric and positive definite matrix, $\epsilon > 0$ a termination scalar, $k := 0$.*

k-th Step: *For $k = 0, 1, \cdots$,*

1. *If $\|g_k\| \leq \epsilon$, stop.*
2. *Compute $d_k = -H_k g_k$.*
3. *Compute the step size α_k.*
4. *Set $s_k = \alpha_k d_k, x_{k+1} = x_k + s_k, y_k = g_{k+1} - g_k$, and*

$$H_{k+1} = H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k}.$$

5. *$k := k + 1$, go to Step 1.* □

DFP method has the following important properties:

1. For a quadratic function (under exact line search)
   - **(1)** DFP update has quadratic termination, i.e., $H_n = G^{-1}$.
   - **(2)** DFP update has hereditary property, i.e., $H_i y_j = s_j, j < i$.
   - **(3)** DFP method generates conjugate directions; when $H_0 = I$, the method generates conjugate gradients.
2. For a general function
   - **(1)** DFP update maintains positive definiteness.
   - **(2)** Each iteration requires $3n^2 + O(n)$ multiplications.
   - **(3)** DFP method is superlinearly convergent.
   - **(4)** For a strictly convex function, under exact line search, DFP method is globally convergent.

The convergence properties of DFP method will be established in §5.3 and §5.4. In the remainder of this subsection we shall discuss the other two important properties: positive definiteness of the update and quadratic termination of the method.

The fact that quasi-Newton update retains positive definiteness is of importance in efficiency, numerical stability and global convergence. If the Hessian $G(x^*)$ is positive definite, the stationary point x^* is a strong minimizer. Hence, we hope Hessian approximation $\{B_k\}$ (or inverse Hessian approximation $\{H_k\}$) is positive definite. In addition, if $\{B_k\}$ (or $\{H_k\}$) is positive definite, the local quadratic model of f has a unique local minimizer, and the direction d_k from (5.1.15) or (5.1.16) is a descent direction. Usually, the update retaining positive definiteness means that if H_k (or B_k) is positive definite, then H_{k+1} (or B_{k+1}) is also positive definite. Such an update is also called positive definite update. Next, we discuss the positive definiteness of DFP update.

Theorem 5.1.5 *(Positive Definiteness of DFP Update)*
DFP update (5.1.30) retains positive definiteness if and only if $s_k^T y_k > 0$.

Proof. For the proof, we give two methods.

Proof (I) Sufficiency. We will prove

$$z^T H_k z > 0, \ \forall z \neq 0 \tag{5.1.31}$$

by induction.

Obviously, H_0 is symmetric and positive definite. We now suppose that (5.1.31) holds for some $k \geq 0$ and set $H_k = LL^T$ as the Cholesky factorization of H_k. Let

$$a = L^T z, \ b = L^T y_k. \tag{5.1.32}$$

Then by DFP update (5.1.30) we have

$$\begin{aligned} z^T H_{k+1} z &= z^T \left(H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} \right) z + z^T \frac{s_k s_k^T}{s_k^T y_k} z \\ &= \left[a^T a - \frac{(a^T b)^2}{b^T b} \right] + \frac{(z^T s_k)^2}{s_k^T y_k}. \end{aligned} \tag{5.1.33}$$

It is obvious from Cauchy-Schwartz inequality that

$$a^T a - \frac{(a^T b)^2}{b^T b} \geq 0. \tag{5.1.34}$$

In addition, the second term in (5.1.33) is also nonnegative because of $s_k^T y_k > 0$. Therefore we obtain that

$$z^T H_{k+1} z \geq 0.$$

Below, we must prove that at least one term in (5.1.33) is strictly larger than zero. Since $z \neq 0$, the equality holds in (5.1.34) if and only if a is parallel to b, equivalently, if and only if z is parallel to y_k. If z is parallel to y_k, we have $z = \beta y_k$, where $\beta \neq 0$, and

$$\frac{(z^T s_k)^2}{s_k^T y_k} = \beta^2 s_k^T y_k > 0,$$

which indicates that if z is parallel to y_k, i.e., if the first term in (5.1.33) equals zero, the second term must be strictly larger than zero. Thus, for any $z \neq 0$, we always have $z^T H_{k+1} z > 0$. The sufficiency follows.

In analogy, the necessity can be shown. □

Proof (II). Let $H_k = LL^T, \bar{y} = L^T y_k, \bar{s} = L^{-1} s_k$. Then DFP update (5.1.30) can be written as

$$H_{k+1} = LWL^T, \tag{5.1.35}$$

where

$$W = I - \frac{\bar{y}\bar{y}^T}{\bar{y}^T \bar{y}} + \frac{\bar{s}\bar{s}^T}{\bar{s}^T \bar{y}}. \tag{5.1.36}$$

By the determinant relation (1.2.70) of update,

$$\det(W) = \frac{\bar{s}^T \bar{y}}{\bar{y}^T \bar{y}} = \frac{s_k^T y_k}{y_k^T H_k y_k},$$

which, together with (5.1.35), gives

$$\det(H_{k+1}) = \det(H_k) \frac{s_k^T y_k}{y_k^T H_k y_k}. \tag{5.1.37}$$

This implies that if H_k is positive definite, then $\det(H_{k+1}) > 0$ if and only if $s_k^T y_k > 0$.

Let

$$H_{k+1} = H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} = \bar{H} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k},$$

where $\bar{H} = H_k + \frac{s_k s_k^T}{s_k^T y_k}$. Since H_k is positive definite, we know by use of Theorem 1.2.17 that $s_k^T y_k > 0$ implies all eigenvalues of $\bar{H}$ are positive, i.e., $\bar{H}$ is positive definite. Using Theorem 1.2.17 again indicates that, at most, the smallest eigenvalue of H_{k+1} is not positive. Hence, $\det(H_{k+1})$ and the smallest eigenvalue of H_{k+1} have the same sign, which shows that H_{k+1} is positive definite if and only if $\det(H_{k+1}) > 0$. Therefore we have

$$s_k^T y_k > 0 \Leftrightarrow \det(H_{k+1}) > 0 \Leftrightarrow H_{k+1} \text{ is positive definite.} \quad \square$$

This theorem gives a sufficient and necessary condition of positive definite DFP update. By different definitions of positive definiteness and different algebraic tricks, we can establish this theorem. The interested readers may try different methods to give the proofs. The curvature condition $s_k^T y_k > 0$ for preserving positive definiteness is moderate, practical, and can be satisfied. For a quadratic positive definite function, obviously,

$$s_k^T y_k = s_k^T G s_k > 0.$$

For a strong convex function, the average Hessian

$$\bar{G}_k = \int_0^1 \nabla^2 f(x_k + \tau s_k) \mathrm{d}\tau \tag{5.1.38}$$

is positive definite. So, from Taylor's formula

$$y_k = \nabla f(x_k + s_k) - \nabla f(x_k) = \int_0^1 \nabla^2 f(x_k + \tau s_k) s_k \mathrm{d}\tau = \bar{G}_k s_k,$$

we have that

$$y_k^T s_k = s_k^T \bar{G}_k s_k > 0.$$

For a general function, we have

$$s_k^T y_k = g_{k+1}^T s_k - g_k^T s_k.$$

Note that $g_k^T s_k < 0$ is due to s_k being a descent direction. Using exact line search with $g_{k+1}^T s_k = 0$, we have $s_k^T y_k > 0$. When we use inexact line search, for example, if the rule (2.5.7) is satisfied, the condition $s_k^T y_k > 0$ can also be satisfied. In general, as long as we increase the precision of line search, we can make $g_{k+1}^T s_k$ small enough in magnitude to the desired degree.

From this theorem and the above discussion, it is obvious that, for Algorithm 5.1.4 with exact or inexact line search, the condition $s_k^T y_k > 0$ holds and therefore each update matrix H_k in DFP algorithm is positive definite. So, we have the following corollary.

Corollary 5.1.6 *Each matrix H_k generated by DFP Algorithm 5.1.4 is positive definite, and the directions $d_k = -H_k g_k$ are descent directions.*

Finally, we give a theorem on quadratic termination of DFP method. This theorem shows that, for a quadratic function with positive definite Hessian G, the directions generated from DFP method are conjugate, and the method terminates at n steps, that is $H_n = G^{-1}$.

Theorem 5.1.7 *(Quadratic Termination Theorem of DFP Method)*
Let $f(x)$ be a quadratic function with positive definite Hessian G. Then, if exact line search is used, the sequence $\{s_j\}$ generated from DFP method satisfies hereditary property, conjugate property and quadratic termination, that is, for $i = 0, 1, \cdots, m$, where $m \leq n-1$,

1. *$H_{i+1} y_j = s_j,\ j = 0, 1, \cdots, i$; (hereditary property)*
2. *$s_i^T G s_j = 0,\ j = 0, 1, \cdots, i-1$; (conjugate direction property)*
3. *The method terminates at $m+1 \leq n$ steps. If $m = n-1$, then $H_n = G^{-1}$.*

Proof. We prove part (1) and (2) by induction. Clearly, when $i = 0$, it is trivial. Now suppose that part (1) and (2) hold for some i. We show that they also hold for $i+1$. Since $g_{i+1} \neq 0$, by exact line search, the fact that $y_k = g_{k+1} - g_k = G(x_{k+1} - x_k) = G s_k, (1 \leq k \leq i)$ and the induction hypothesis, we have, for $j \leq i$,

$$\begin{aligned} g_{i+1}^T s_j &= g_{j+1}^T s_j + \sum_{k=j+1}^{i} (g_{k+1} - g_k)^T s_j \\ &= g_{j+1}^T s_j + \sum_{k=j+1}^{i} y_k^T s_j \\ &= 0 + \sum_{k=j+1}^{i} s_k^T G s_j \\ &= 0. \end{aligned} \tag{5.1.39}$$

Hence, by use of $s_{i+1} = -\alpha_{i+1} H_{i+1} g_{i+1}$, induction hypothesis in part (1) and (5.1.39), it follows that

$$s_{i+1}^T G s_j = -\alpha_{i+1} g_{i+1}^T H_{i+1} y_j$$

$$\begin{aligned} &= -\alpha_{i+1} g_{i+1}^T s_j \\ &= 0, \end{aligned} \tag{5.1.40}$$

which proves part (2) holds for $i+1$.

Next, we prove that part (1) holds for $i+1$, i.e.,

$$H_{i+2} y_j = s_j,\ j = 0, 1, \cdots, i+1. \tag{5.1.41}$$

When $j = i+1$, part (1) is immediate from DFP update (5.1.30), that is

$$H_{i+2} y_{i+1} = s_{i+1}. \tag{5.1.42}$$

When $j \leq i$, it follows from (5.1.40) and the induction hypothesis in part (1) that

$$s_{i+1}^T y_j = s_{i+1}^T G s_j = 0,$$
$$y_{i+1}^T H_{i+1} y_j = y_{i+1}^T s_j = s_{i+1}^T G s_j = 0.$$

Then

$$\begin{aligned} H_{i+2} y_j &= H_{i+1} y_j + \frac{s_{i+1} s_{i+1}^T y_j}{s_{i+1}^T y_{i+1}} - \frac{H_{i+1} y_{i+1} y_{i+1}^T H_{i+1} y_j}{y_{i+1}^T H_{i+1} y_{i+1}} \\ &= H_{i+1} y_j \\ &= s_j. \end{aligned} \tag{5.1.43}$$

This, together with (5.1.42), shows (5.1.41). Therefore part (1) follows.

Finally, since s_i $(i = 0, 1, \cdots, m)$ are conjugate, the method is a conjugate direction method. Based on Theorem 4.1.3 of the conjugate direction method, the method terminates after $m(\leq n)$ steps. When $m = n-1$, since s_i $(i = 0, 1, \cdots, n-1)$ are linearly independent, then part (1) means

$$H_n G s_j = H_n y_j = s_j,\ j = 0, 1, \cdots, n-1$$

which implies $H_n = G^{-1}$. $\square$

From this theorem we see that DFP method is a conjugate direction method. If the initial approximation $H_0 = I$, the method becomes a conjugate gradient method. By the hereditary property, we have $H_{i+1} G s_j = s_j, j = 0, 1, \cdots, i$, which also indicates that these s_j are eigenvectors of matrix $H_{i+1} G (j = 0, 1, \cdots, i)$ corresponding to the eigenvalue 1.

DFP method is a seminal quasi-Newton method and has been widely used in many computer codes. It has played an important role in theoretical

analysis and numerical computing. However, further studies indicate that DFP method is numerically unstable, and sometimes produces numerically singular Hessian approximations. The other famous quasi-Newton update — BFGS update introduced in the next subsection will overcome these drawbacks and perform better than DFP update.

5.1.4 BFGS Update and PSB Update

In §5.1.1 we have seen that

$$H_{k+1}y_k = s_k \text{ and } B_{k+1}s_k = y_k \tag{5.1.44}$$

are the quasi-Newton equations with respect to inverse Hessian approximation and Hessian approximation respectively. Note that any approximation in (5.1.44) can be obtained from the other by means of exchanging $H_{k+1} \leftrightarrow B_{k+1}$ and $s_k \leftrightarrow y_k$. In analogy to the derivation of DFP update (5.1.30) about H_k, we can get

$$B_{k+1}^{(BFGS)} = B_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}, \tag{5.1.45}$$

which is called BFGS update discovered independently by Broyden [27], Fletcher [125], Goldfarb [153] and Shanno [304]. In fact, if one makes directly simple exchanges $H_k \leftrightarrow B_k$ and $s_k \leftrightarrow y_k$, BFGS update (5.1.45) is just obtained from DFP update (5.1.30). Thus, BFGS update is also said to be a complement DFP update. Since $B_k s_k = -\alpha_k g_k$ and $B_k d_k = -g_k$, (5.1.45) can also be written as

$$B_{k+1}^{(BFGS)} = B_k + \frac{g_k g_k^T}{g_k^T d_k} + \frac{y_k y_k^T}{\alpha_k y_k^T d_k}. \tag{5.1.46}$$

By using twice the Sherman-Morrison formula (1.2.67), (5.1.45) will become as follows:

$$\begin{aligned} H_{k+1}^{(BFGS)} &= H_k + \left(1 + \frac{y_k^T H_k y_k}{s_k^T y_k}\right)\frac{s_k s_k^T}{s_k^T y_k} \\ &\quad - \frac{s_k y_k^T H_k + H_k y_k s_k^T}{s_k^T y_k} \\ &= H_k + \frac{(s_k - H_k y_k)s_k^T + s_k(s_k - H_k y_k)^T}{s_k^T y_k} \end{aligned} \tag{5.1.47}$$

$$-\frac{(s_k - H_k y_k)^T y_k}{(s_k^T y_k)^2} s_k s_k^T \tag{5.1.48}$$

$$= \left(I - \frac{s_k y_k^T}{s_k^T y_k}\right) H_k \left(I - \frac{y_k s_k^T}{s_k^T y_k}\right) + \frac{s_k s_k^T}{s_k^T y_k}. \tag{5.1.49}$$

(5.1.47)–(5.1.49) are the three forms of BFGS update about H_k. Furthermore, by making exchanges $H_k \leftrightarrow B_k$ and $s_k \leftrightarrow y_k$ in (5.1.47)–(5.1.49), we can get three corresponding forms of DFP update about B_k:

$$\begin{aligned} B_{k+1}^{(DFP)} &= B_k + \left(1 + \frac{s_k^T B_k s_k}{y_k^T s_k}\right) \frac{y_k y_k^T}{y_k^T s_k} \\ &\quad - \frac{y_k s_k^T B_k + B_k s_k y_k^T}{y_k^T s_k} \end{aligned} \tag{5.1.50}$$

$$\begin{aligned} &= B_k + \frac{(y_k - B_k s_k) y_k^T + y_k (y_k - B_k s_k)^T}{y_k^T s_k} \\ &\quad - \frac{(y_k - B_k s_k)^T s_k}{(y_k^T s_k)^2} y_k y_k^T \end{aligned} \tag{5.1.51}$$

$$= \left(I - \frac{y_k s_k^T}{y_k^T s_k}\right) B_k \left(I - \frac{s_k y_k^T}{y_k^T s_k}\right) + \frac{y_k y_k^T}{y_k^T s_k}. \tag{5.1.52}$$

The above discussions describe a method for finding its dual update from a given update. Given a quasi-Newton update H_{k+1} about H-form, by exchanging $H_k \leftrightarrow B_k$ and $s_k \leftrightarrow y_k$, we can get its dual update $B_{k+1}^{(D)}$ about B-form. Then, applying the Sherman-Morrison formula to $B_{k+1}^{(D)}$, we will produce the dual update $H_{k+1}^{(D)}$ of H_{k+1} about the H-form. Similarly, if we employ the same operations to the dual update $H_{k+1}^{(D)}$, the original update H_{k+1} will be restored. Notice that, for an H-form, the dual update of H_{k+1} is $H_{k+1}^{(D)}$. In addition, the dual operation maintains the quasi-Newton equation. The following figure represents the dual relation.

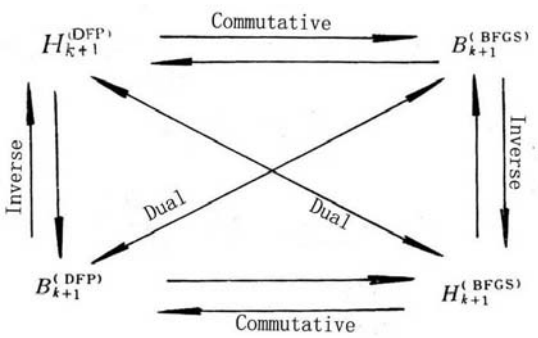


Figure 5.1.1 Duality of $H_{k+1}^{(DFP)}$ and $H_{k+1}^{(BFGS)}$

For SR1 update

$$H_{k+1}^{(SR1)} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}, \tag{5.1.53}$$

exchanging $H_k \leftrightarrow B_k$ and $s_k \leftrightarrow y_k$ gives

$$B_{k+1}^{(D)} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}. \tag{5.1.54}$$

Then applying the Sherman-Morrison formula to (5.1.54), we see that the resultant $H_{k+1}^{(D)}$ is still the $H_{k+1}^{(SR1)}$, i.e., $H_{k+1}^{(D)} = H_{k+1}^{(SR1)}$. Thus SR1 update is self-dual. As we pointed out in §5.1.2, SR1 update does not retain the positive definiteness of the update. A self-dual update retaining the positive definiteness is called Hoshino update which will be given in (5.2.6) of §5.2.

The BFGS update is presently considered to be the best one of all quasi-Newton updates, which has all good properties of DFP update. In addition, when inexact line search (2.5.3) and (2.5.7) are used, BFGS method is globally convergent. Note that it is still an open problem whether DFP update has this property. The numerical performance of BFGS update is superior to that of DFP update. In particular, BFGS update can often work well in conjunction with some line searches with lower accuracy.

The next topic in this subsection is PSB update which is formally known as the Powell-symmetric-Broyden update due to Powell [260].

Let $B \in R^{n\times n}$ be a symmetric matrix. Consider the general Broyden rank-one update

$$C_1 = B + \frac{(y - Bs)c^T}{c^T s},$$

where $c \in R^n, c^T s \neq 0$. In general, C_1 is not symmetric. So, we consider a symmetrization:

$$C_2 = (C_1 + C_1^T)/2.$$

Now C_2 is symmetric but, in general, does not obey the quasi-Newton equation. Then we might continue the above process and generate the sequence $\{C_k\}$:

$$\begin{aligned} C_{2k+1} &= C_{2k} + \frac{(y - C_{2k}s)c^T}{c^T s}, \\ C_{2k+2} &= (C_{2k+1} + C_{2k+1}^T)/2, \ k = 0, 1, \cdots \end{aligned} \tag{5.1.55}$$

where $C_0 = B$. Here each C_{2k+1} is the closest matrix in $Q(y, s)$ to C_{2k}, and each C_{2k+2} is the closest symmetric matrix to C_{2k+1}, where $Q(y, s) = \{C \in R^{n\times n} \mid Cs = y\}$ is a matrix set satisfying the quasi-Newton equation. The Figure 5.1.2 illustrates the symmetrization process, where S denotes the set of symmetric matrices.

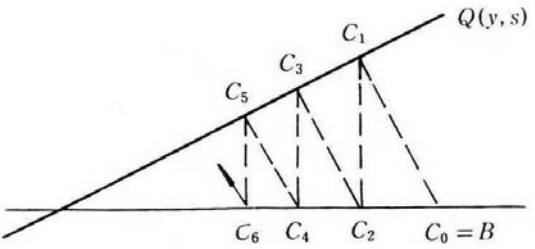


Figure 5.1.2 Production of the sequence C_k

Below, we show the limit of matrix sequence $\{C_k\}$ is

$$\bar{B} = B + \frac{(y - Bs)c^T + c(y - Bs)^T}{c^T s} - \frac{(y - Bs)^T s}{(c^T s)^2} cc^T \tag{5.1.56}$$

which satisfies symmetricity and the quasi-Newton equation.

Theorem 5.1.8 *Let $B \in R^{n\times n}$ be symmetric, $c, s, y \in R^n$ and $c^T s \neq 0$. Let the sequence $\{C_k\}$ be defined by (5.1.55), and $C_0 = B$. Then the sequence $\{C_k\}$ converges to $\bar{B}$ in (5.1.56).*

Proof. We only need to prove that the sequence $\{C_{2k}\}$ converges. Let $G_k = C_{2k}$. From (5.1.55), we have

$$G_{k+1} = G_k + \frac{1}{2}\frac{w_k c^T + c w_k^T}{c^T s}, \tag{5.1.57}$$

where $w_k = y - G_k s$. Note that

$$\begin{aligned} w_{k+1} &= y - G_{k+1}s \\ &= y - G_k s - \frac{1}{2}\frac{w_k c^T s + c w_k^T s}{c^T s} \\ &= \frac{1}{2}\left(I - \frac{cs^T}{c^T s}\right) w_k, \end{aligned}$$

that is

$$w_{k+1} = P w_k, \text{ where } P = \frac{1}{2}\left[I - \frac{cs^T}{c^T s}\right]. \tag{5.1.58}$$

Then it follows from Sherman-Morrison formula (1.2.67) that

$$\begin{aligned} \sum_{k=0}^{\infty} w_k &= \sum_{k=0}^{\infty} P^k (y - G_0 s) = \sum_{k=0}^{\infty} P^k (y - Bs) \\ &= (I - P)^{-1}(y - Bs) = 2\left[I - \frac{1}{2}\frac{cs^T}{c^T s}\right](y - Bs) \\ &= 2(y - Bs) - \frac{cs^T}{c^T s}(y - Bs). \end{aligned} \tag{5.1.59}$$

Since

$$\lim_{k\to\infty} G_k = B + \sum_{k=0}^{\infty}(G_{k+1} - G_k), \tag{5.1.60}$$

and by (5.1.57) and (5.1.59), we get that the sequence $\{G_k\}$ is convergent. Note that

$$\begin{aligned} &\sum_{k=0}^{\infty}(G_{k+1} - G_k) = \frac{1}{2}\sum_{k=0}^{\infty}\frac{w_k c^T + c w_k^T}{c^T s} \\ = \;& \frac{1}{c^T s}\left[(y - Bs)c^T - \frac{1}{2}\frac{s^T(y - Bs)}{c^T s} cc^T + c(y - Bs)^T - \frac{1}{2}\frac{(y - Bs)^T s}{c^T s} cc^T\right] \\ = \;& \frac{1}{c^T s}[(y - Bs)c^T + c(y - Bs)^T] - \frac{(y - Bs)^T s}{(c^T s)^2} cc^T, \end{aligned} \tag{5.1.61}$$

hence the conclusion (5.1.56) follows by (5.1.60) and (5.1.61). □

(5.1.56) gives a class of rank-two update which is derived by a symmetrization process. If we add the subscripts, it can be written as

$$\begin{aligned} B_{k+1} &= B_k + \frac{(y_k - B_k s_k)c_k^T + c_k(y_k - B_k s_k)^T}{c_k^T s_k} \\ &\quad - \frac{(y_k - B_k s_k)^T s_k}{(c_k^T s_k)^2} c_k c_k^T, \end{aligned} \tag{5.1.62}$$

which is called the general PSB update. In particular,

If $c_k = y_k - B_k s_k$, (5.1.62) is SR1 update (5.1.54).

If $c_k = y_k$, (5.1.62) is DFP update (5.1.51).

If $c_k = \frac{1}{w_k+1} y_k + \frac{w_k}{w_k+1} B_k s_k$, where $w_k = (y_k^T s_k / s_k^T B_k s_k)^{\frac{1}{2}}$, (5.1.62) is BFGS update (5.1.46).

If $c_k = s_k$, (5.1.62) is PSB update:

$$\begin{aligned} B_{k+1} &= B_k + \frac{(y_k - B_k s_k)s_k^T + s_k(y_k - B_k s_k)^T}{s_k^T s_k} \\ &\quad - \frac{(y_k - B_k s_k)^T s_k}{(s_k^T s_k)^2} s_k s_k^T. \end{aligned} \tag{5.1.63}$$

Its dual update in H-form is

$$\begin{aligned} H_{k+1} &= H_k + \frac{(s_k - H_k y_k)y_k^T + y_k(s_k - H_k y_k)^T}{y_k^T y_k} \\ &\quad - \frac{(s_k - H_k y_k)^T y_k}{(y_k^T y_k)^2} y_k y_k^T \end{aligned} \tag{5.1.64}$$

which is called Greenstadt update (see Greenstadt [163]).

PSB update (5.1.63) is important in theoretical research and practical computing. However, the drawback that PSB update does not retain the positive definiteness of updates hurts its performance in computing. Fortunately, the drawback can be avoided if we employ the trust region framework with PSB update.

5.1.5 The Least Change Secant Update

Various quasi-Newton updates obey the least change property which refers to the H_{k+1} (or B_{k+1}) being the minimum change to H_k (or B_k) consistent with the quasi-Newton equation if the change $H_{k+1} - H_k$ (or $B_{k+1} - B_k$) is measured under some norm. This property is helpful to maintain some information of the last iteration. By the way, by use of the property, we also can derive quasi-Newton update.

Theorem 5.1.9 *Let $B \in R^{n\times n}$, $s, y \in R^n$ and $s \neq 0$. Then Broyden rank-one update*

$$\bar{B} = B + \frac{(y - Bs)s^T}{s^T s} \tag{5.1.65}$$

is a unique solution of the minimization problem

$$\min\{\|\hat{B} - B\|_F \ : \ \hat{B}s = y\}. \tag{5.1.66}$$

Proof. [proof I] Since $y = \hat{B}s$, then

$$\begin{aligned} \|\bar{B} - B\| &= \left\|\frac{(y - Bs)s^T}{s^T s}\right\|_F = \left\|(\hat{B} - B)\frac{ss^T}{s^T s}\right\|_F \\ &\leq \|\hat{B} - B\|_F. \end{aligned} \tag{5.1.67}$$

Also, since the Frobenius norm is strictly convex and the set of matrix $\hat{B}$ satisfying the quasi-Newton equation is convex, then the solution of (5.1.66) is unique.

[proof II] Define $C = \hat{B} - B$ and let c_i^T be the i-th row of C. Then (5.1.66) can be represented as

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} \|c_i^T\|_2^2 \\ \text{s.t.} \quad & c_i^T s = (y - Bs)_i, \ i = 1, \cdots, n \end{aligned} \tag{5.1.68}$$

where $(y - Bs)_i$ denotes the i-th component of $y - Bs$. Obviously, (5.1.68) can be divided into n subproblems

$$\begin{aligned} \min \quad & \|c_i^T\|_2^2 \\ \text{s.t.} \quad & c_i^T s = (y - Bs)_i. \end{aligned} \tag{5.1.69}$$

Solving (5.1.69) is equivalent to finding the Moore-Penrose inverse s^+ of s. Therefore

$$c_i^T = (y - Bs)_i s^+ = \frac{(y - Bs)_i s^T}{s^T s}$$

which indicates that (5.1.65) is the unique solution of (5.1.66). □

This theorem shows that Broyden's rank-one update

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k) s_k^T}{s_k^T s_k} \tag{5.1.70}$$

is the unique solution of the minimization problem

$$\min\{\|\hat{B} - B_k\|_F \ : \ \hat{B}s_k = y_k\}. \tag{5.1.71}$$

Similarly,

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k) y_k^T}{y_k^T y_k} \tag{5.1.72}$$

is the unique solution of the minimization problem

$$\min\{\|\hat{H} - H_k\|_F \ : \ \hat{H}y_k = s_k\}. \tag{5.1.73}$$

Next, we discuss the least change property of general symmetric rank-two update.

Theorem 5.1.10 *Let $B \in R^{n\times n}$ be symmetric, $c, s, y \in R^n$, and $c^T s > 0$. Assume that $M \in R^{n\times n}$ is a symmetric and nonsingular matrix satisfying*

$$Mc = M^{-1}s. \tag{5.1.74}$$

Then the general PSB update

$$\bar{B} = B + \frac{(y - Bs)c^T + c(y - Bs)^T}{c^T s} - \frac{(y - Bs)^T s}{(c^T s)^2} cc^T \tag{5.1.75}$$

is the unique solution of the minimization problem

$$\min\{\|\hat{B} - B\|_{M,F} \ : \ \hat{B}s = y, \hat{B}^T = \hat{B}\}, \tag{5.1.76}$$

where $\|B\|_{M,F} = \|MBM\|_F$.

Proof. Let $\hat{B}$ be a symmetric matrix obeying $y = \hat{B}s$. Let also $Mc = M^{-1}s = z, E = M(\hat{B} - B)M, \bar{E} = M(\bar{B} - B)M$. Left- and right-multiplying (5.1.75) by M yields

$$\bar{E} = \frac{Ezz^T + zz^TE}{z^Tz} - \frac{z^TEz}{(z^Tz)^2}zz^T.$$

Clearly, $\|\bar{E}z\|_2 = \|Ez\|_2$, and if $v \perp z$, then $\|\bar{E}v\|_2 \leq \|Ev\|_2$. Therefore $\|\bar{E}\|_F \leq \|E\|_F$. Also, note that the weighted Frobenius norm $\|\cdot\|_{M,F}$ is strictly convex and the matrix set $\{\hat{B} \mid \hat{B}s = y, \hat{B}^T = \hat{B}\}$ is convex, thus the general PSB update (5.1.75) is the unique solution of the problem (5.1.76). □

In particular, some different choices of c in (5.1.75) give different conclusions.

Choosing $c = s$ (in this case, $M = I$), we get PSB update (5.1.63). Hence, Theorem 5.1.10 implies that $\bar{B}^{PSB}$ is the unique solution to the problem

$$\min_{\hat{B} \in R^{n \times n}} \{\|\hat{B} - B\|_F \mid \hat{B}s = y, \hat{B}^T = \hat{B}\}. \tag{5.1.77}$$

Choosing $c = y$ (in this case, M satisfies $M^{-2}s = y$), we get DFP update (5.1.50). Hence Theorem 5.1.10 implies that $\bar{B}^{DFP}$ is the unique solution to the problem

$$\min\{\|\hat{B} - B\|_{M,F} \mid \hat{B}s = y, \hat{B}^T = \hat{B}\}. \tag{5.1.78}$$

Similarly, by the dual technique, $\bar{H}^{BFGS}$ in (5.1.47) is the unique solution to the problem

$$\min\{\|\hat{H} - H\|_{M^{-1},F} \mid \hat{H}y = s, \hat{H}^T = \hat{H}\}. \tag{5.1.79}$$

As an exercise, it is not difficult to discuss the least change property of dual general PSB update.

5.2 The Broyden Class

From the last section we have seen that both DFP and BFGS updates are symmetric and positive definite rank-two updates consisting of H_ky_k and s_k. It is natural to discuss their weighted (or convex) combinations which have the same type, and consider their behaviors.

Consider the update class

$$H_{k+1}^{\phi} = (1-\phi)H_{k+1}^{DFP} + \phi H_{k+1}^{BFGS}, \tag{5.2.1}$$

where ϕ is a parameter. (5.2.1) is called the Broyden class of update. If $\phi \in [0,1]$, (5.2.1) is called the Broyden convex class of update. Obviously, Broyden class (5.2.1) satisfies quasi-Newton equation (5.1.8). We can also write (5.2.1) in the following forms:

$$\begin{aligned} H_{k+1}^{\phi} &= H_{k+1}^{DFP} + \phi v_k v_k^T && (5.2.2)\\ &= H_{k+1}^{BFGS} + (\phi-1) v_k v_k^T && (5.2.3)\\ &= H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \phi v_k v_k^T, && (5.2.4) \end{aligned}$$

where

$$v_k = (y_k^T H_k y_k)^{\frac{1}{2}} \left[\frac{s_k}{s_k^T y_k} - \frac{H_k y_k}{y_k^T H_k y_k} \right]. \tag{5.2.5}$$

In particular, in (5.2.4),

set $\phi = 0$, we get DFP update (5.1.30);

set $\phi = 1$, we get BFGS update (5.1.47);

set $\phi = \frac{s_k^T y_k}{(s_k - H_k y_k)^T y_k}$, we get SR1 update (5.1.22);

set

$$\phi = \frac{1}{1 \mp (y_k^T H_k y_k / s_k^T y_k)}, \tag{5.2.6}$$

we get Hoshino update.

Broyden class (5.2.2)–(5.2.4) can be derived directly by the quasi-Newton equation. Consider a general rank-two update consisting of s_k and $H_k y_k$:

$$H_{k+1} = H_k + a s_k s_k^T + b(H_k y_k s_k^T + s_k y_k^T H_k) + c H_k y_k y_k^T H_k, \tag{5.2.7}$$

where a, b, c are scalars to be determined. Using the quasi-Newton equation yields

$$\begin{aligned} 1 &= a s_k^T y_k + b y_k^T H_k y_k, \\ 0 &= 1 + b s_k^T y_k + c y_k^T H_k y_k. \end{aligned} \tag{5.2.8}$$

Here are two equations with three unknowns and one free degree. Set

$$b = -\phi / s_k^T y_k, \tag{5.2.9}$$

where ϕ is a parameter. Solving (5.2.8) and substituting the result into (5.2.7), we have

$$H_{k+1}^{\phi} = H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \phi v_k v_k^T = H_{k+1}^{DFP} + \phi v_k v_k^T,$$

where v_k is defined by (5.2.5). The above expression is just (5.2.2) and (5.2.4). By a slight arrangement, Broyden class has the following matrix form:

$$H_{k+1}^{\phi} = H_k + [s_k, H_k y_k] \begin{bmatrix} \frac{1+\phi y_k^T H_k y_k / s_k^T y_k}{s_k^T y_k} & -\frac{\phi}{s_k^T y_k} \\ -\frac{\phi}{s_k^T y_k} & \frac{\phi - 1}{y_k^T H_k y_k} \end{bmatrix} [s_k, H_k y_k]^T. \tag{5.2.10}$$

Correspondingly, it is easy to produce Broyden class in B-form:

$$\begin{aligned} B_{k+1}^{\theta} &= \theta B_{k+1}^{DFP} + (1-\theta) B_{k+1}^{BFGS} && (5.2.11)\\ &= B_{k+1}^{BFGS} + \theta w_k w_k^T && (5.2.12)\\ &= B_{k+1}^{DFP} + (\theta - 1) w_k w_k^T && (5.2.13)\\ &= B_k + \frac{y_k y_k^T}{s_k^T y_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \theta w_k w_k^T, && (5.2.14) \end{aligned}$$

where

$$w_k = (s_k^T B_k s_k)^{1/2} \left[\frac{y_k}{s_k^T y_k} - \frac{B_k s_k}{s_k^T B_k s_k} \right]. \tag{5.2.15}$$

Note that the relation between θ and ϕ is

$$\theta = (\phi - 1)/(\phi - 1 - \phi\mu), \tag{5.2.16}$$

where

$$\mu = \frac{y_k^T H_k y_k s_k^T B_k s_k}{(s_k^T y_k)^2}. \tag{5.2.17}$$

Since $v_k^T y_k = 0$ and $w_k^T s_k = 0$, then (5.2.1)-(5.2.2) and (5.2.11)-(5.2.14) satisfy respectively the quasi-Newton equation (5.1.8) and (5.1.13) for any parameter ϕ and θ. In analogous to Theorem 5.1.2 and Theorem 5.1.5, we can show the quadratic termination property and positive definite property of Broyden class.

Theorem 5.2.1 *(Quadratic Termination Theorem of Broyden Class) Let $f(x)$ be a quadratic function with positive definite Hessian G. Then, when exact line search is used, the Broyden class of update has hereditary property and conjugate direction property, that is, for $i = 0, 1, \cdots, m, (m \leq n-1)$,*

$$\text{Hereditary property:} \quad H_{i+1} y_j = s_j, \ j = 0, 1, \cdots, i. \tag{5.2.18}$$

$$\text{Conjugate direction:} \quad s_i^T G s_j = 0, \ j = 0, 1, \cdots, i-1. \tag{5.2.19}$$

The method terminates at m steps. If $m = n-1$, then $H_n = G^{-1}$.

Proof. It is similar to the proof of Theorem 5.1.5. □

Theorem 5.2.2 *(Positive Definiteness of Broyden Class of Update) Let $\phi \geq 0$. If and only if $s_k^T y_k > 0$, Broyden class of update (5.2.2) retains the positive definiteness.*

Proof. From Theorem 5.1.2, if and only if $s_k^T y_k > 0$, DFP update retains positive definiteness. Since $\phi \geq 0$, it follows from (5.2.3) and Theorem 1.2.17 that the smallest eigenvalue of H_{k+1}^{ϕ} is not less than the smallest one of H_{k+1}^{DFP}. Hence H_{k+1}^{ϕ} is positive definite. □

This theorem shows that not all members of Broyden class retain the positive definiteness. Clearly, when $\phi \geq 0$, H_{k+1}^{ϕ} maintains its positive definiteness; when $\phi < 0$, it is possible that the update becomes singular. The following Theorem 5.2.3 gives a value $\bar{\phi}$, and says that as long as $\phi > \bar{\phi}$, H_{k+1}^{ϕ} will maintain positive definiteness. Such a value $\bar{\phi}$ is called the degenerate value of Broyden class, which makes $H_{k+1}^{\bar{\phi}}$ singular.

Theorem 5.2.3 *The degenerate value of Broyden class of update is*

$$\bar{\phi} = \frac{1}{1-\mu} = \frac{1}{1 - y_k^T H_k y_k s_k^T B_k s_k / (s_k^T y_k)^2}. \tag{5.2.20}$$

Proof. Let $d_k = -H_k g_k, s_k = \alpha_k d_k$. When we use exact line search, $g_{k+1}^T d_k = 0 = g_{k+1}^T s_k$. Notice also that $g_{k+1} = y_k + g_k, v_k^T g_k = 0$, and using (5.2.5), we have

$$\begin{aligned} d_{k+1}^{\phi} &= -H_{k+1}^{\phi} g_{k+1} \\ &= -\left(H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \phi v_k v_k^T \right) g_{k+1} \end{aligned}$$

$$
\begin{aligned}
&= -H_k g_k - H_k y_k + \frac{y_k^T H_k (g_k + y_k)}{y_k^T H_k y_k} H_k y_k - \phi v_k^T g_k v_k \\
&= -H_k g_k + \frac{y_k^T H_k g_k}{y_k^T H_k y_k} H_k y_k - \phi v_k^T g_k v_k \\
&= d_k - \frac{y_k^T d_k}{y_k^T H_k y_k} H_k y_k - \phi v_k^T g_k v_k \\
&= \frac{d_k^T y_k}{(y_k^T H_k y_k)^{1/2}} \left[(y_k^T H_k y_k)^{1/2} \left(\frac{d_k}{d_k^T y_k} - \frac{H_k y_k}{y_k^T H_k y_k} \right) \right] - \phi v_k^T g_k v_k \\
&= \left(\frac{d_k^T y_k}{(y_k^T H_k y_k)^{1/2}} - \phi v_k^T g_k \right) v_k. \qquad (5.2.21)
\end{aligned}
$$

This shows that when exact line search is used, (5.2.21) holds. When $g_{k+1} \neq 0$, if $d_{k+1}^{\phi} = -H_{k+1}^{\phi} g_{k+1} = 0$, then ϕ is called the degenerate value of H_{k+1}^{ϕ}. By using $d_{k+1}^{\phi} = 0$ and (5.2.5), we obtain

$$
\begin{aligned}
\phi &= \frac{y_k^T d_k}{(y_k^T H_k y_k)^{1/2} v_k^T g_k} \\
&= \frac{y_k^T d_k}{-g_k^T H_k y_k + (s_k^T g_k)(y_k^T H_k y_k)/s_k^T y_k} \\
&= \frac{1}{1 - \frac{(s_k^T B_k s_k)(y_k^T H_k y_k)}{(s_k^T y_k)^2}} \\
&= \frac{1}{1-\mu}. \quad \square
\end{aligned}
$$

(5.2.21) indicates that the parameter ϕ of Broyden class does not change the search direction, but only the length. Hence, we could expect that: any method of Broyden class is, in some degree, independent from the parameter ϕ. Dixon [107] proves: under exact line search, all updates of Broyden class ($\phi_k > \bar{\phi}$) generate the identical points, although for non-quadratic functions.

Theorem 5.2.4 *Let $f : R^n \to R$ be continuously differentiable, the level set $L(x_0) = \{x \mid f(x) \leq f(x_0)\}$ be bounded, and $H_0 \in R^{n \times n}$ be symmetric and positive definite. Let $\{H_k^{\phi}\}$ be a sequence generated by Broyden class, where $\phi_k > \bar{\phi}$ and $\bar{\phi}$ is the degenerate value of Broyden class. Assume that H_{k+1}^{BFGS} is an update obtained by applying BFGS update to H_k^{ϕ}. Then, under exact*

line search, Broyden class of update has the property: for all $k \geq 0$, x_{k+1} and H_{k+1}^{BFGS} are independent from parameters $\phi_0, \phi_1, \cdots, \phi_{k-1}$.

Proof. We show this result by induction. For $k = 0$, it is trivially true. Now suppose it is true for $k \geq 0$, i.e., x_{k+1} and H_{k+1}^{BFGS} are independent from $\phi_0, \phi_1, \cdots, \phi_{k-1}$. We shall show it is also true for $k + 1$.

From (5.2.21), the direction d_{k+1} generated by Broyden class does not depend on ϕ_k. Since $d_{k+1} \propto -H_{k+1}^{BFGS} g_{k+1}$, by the induction hypothesis, the direction d_{k+1} does not also depend on $\phi_0, \phi_1, \cdots, \phi_{k-1}$. Then, by exact line search, $x_{k+2} = x_{k+1} + \alpha_{k+1} d_{k+1}$ does not depend on $\phi_0, \phi_1, \cdots, \phi_{k-1}, \phi_k$. Now, from the assumption,

$$H_{k+2}^{BFGS} = \left(I - \frac{s_{k+1} y_{k+1}^T}{s_{k+1}^T y_{k+1}} \right) H_{k+1}^{\phi} \left(I - \frac{y_{k+1} s_{k+1}^T}{s_{k+1}^T y_{k+1}} \right) + \frac{s_{k+1} s_{k+1}^T}{s_{k+1}^T y_{k+1}}. \quad (5.2.22)$$

Note

$$H_{k+1}^{\phi} = H_{k+1}^{BFGS} + (\phi_k - 1) v_k v_k^T. \quad (5.2.23)$$

Since

$$\left[I - \frac{s_{k+1} y_{k+1}^T}{s_{k+1}^T y_{k+1}} \right] s_{k+1} = 0,$$

it follows from (5.2.21) that

$$\left[I - \frac{s_{k+1} y_{k+1}^T}{s_{k+1}^T y_{k+1}} \right] v_k = 0. \quad (5.2.24)$$

Then, substituting (5.2.23) into (5.2.22) and using (5.2.24) yield that H_{k+2}^{BFGS} can be defined by use of H_{k+1}^{BFGS}, s_{k+1}, and y_{k+1}. So, by induction hypothesis, H_{k+2}^{BFGS} is independent from $\phi_0, \phi_1, \cdots, \phi_k$. We complete the proof. □

To conclude this section, we give a brief introduction to Huang class of updates. Huang [180] presented a wider class of updates than Broyden class. In Broyden class, the update matrix sequence $\{H_k\}$ satisfies symmetricity and quasi-Newton equation, i.e.,

$$H_k^T = H_k \text{ and } H_{k+1} y_k = s_k. \quad (5.2.25)$$

However, in Huang class, the symmetricity condition is removed, and the update matrix $\{H_k\}$ is required to obey

$$H_{k+1} y_k = \rho s_k, \quad (5.2.26)$$

which is said to be a generalized quasi-Newton equation or a generalized quasi-Newton condition, where ρ is a parameter.

Huang class of updates can be described as follows:

$$H_{k+1} = H_k + s_k u_k^T + H_k y_k v_k^T, \tag{5.2.27}$$

where u_k and v_k satisfy

$$u_k = a_{11} s_k + a_{12} H_k^T y_k, \tag{5.2.28}$$
$$v_k = a_{21} s_k + a_{22} H_k^T y_k, \tag{5.2.29}$$
$$u_k^T y_k = \rho, \tag{5.2.30}$$
$$v_k^T y_k = -1. \tag{5.2.31}$$

There are five parameters $a_{11}, a_{12}, a_{21}, a_{22}$ and ρ, in which three parameters are free. Hence, in fact, Huang class of update depends on three parameters. In particular, if requiring $\{H_k\}$ symmetric and setting $\rho = 1$, then Huang class is just Broyden class. This means that Broyden class is a subclass of Huang class.

The main properties of Huang class of update are as follows:

- For positive definite and quadratic functions, Huang class generates conjugate directions and has quadratic termination property. All methods of Huang class generate the identical points.
- For general functions, the sequence generated by Huang class only depends on the parameter ρ.

Based on our experience, the generalized quasi-Newton equation (5.2.26) is important to present a good quasi-Newton method. The parameter ρ will play a big role on the iterative sequence and the properties of algorithms.

5.3 Global Convergence of Quasi-Newton Methods

In this section we discuss the global convergence for quasi-Newton methods. The global properties of quasi-Newton methods were established by Powell [262] and Powell [265]. These results have been extended to restricted Broyden's class by Byrd, Nocedal and Yuan [47]. We will study the global convergence of quasi-Newton methods under exact line search and inexact line search respectively in §5.3.1 and §5.3.2.

In the discussion of this section, we need the following assumptions:

Assumption 5.3.1 **(a)** *$f : R^n \to R$ is twice continuously differentiable on convex set D.*

(b) *$f(x)$ is uniformly convex, i.e., there exist positive constants m and M such that for all $x \in L(x) = \{x | f(x) \leq f(x_0)\}$, which is convex, we have*

$$m\|u\|^2 \leq u^T \nabla^2 f(x) u \leq M\|u\|^2, \ \forall u \in R^n. \tag{5.3.1}$$

The assumption (b) implies that $\nabla^2 f(x)$ is positive definite on $L(x)$, and that f has a unique minimizer x^* in $L(x)$.

5.3.1 Global Convergence under Exact Line Search

We begin the discussion in case of exact line search.

Let

$$\bar{G} = \int_0^1 \nabla^2 f(x_k + \tau s_k) \mathrm{d}\tau, \tag{5.3.2}$$

then we have from Taylor's theorem that

$$y_k = \bar{G} s_k. \tag{5.3.3}$$

Immediately, we have

$$m \leq \frac{y_k^T s_k}{\|s_k\|^2} = \frac{s_k^T \bar{G}_k s_k}{\|s_k\|^2} \leq M \tag{5.3.4}$$

and

$$\frac{1}{M} \leq \frac{\|s_k\|^2}{y_k^T s_k} \leq \frac{1}{m}. \tag{5.3.5}$$

Since also

$$\frac{\|y_k\|^2}{s_k^T y_k} = \frac{s_k^T \bar{G}_k^2 s_k}{s_k^T \bar{G}_k s_k},$$

if we let $z_k = \bar{G}_k^{\frac{1}{2}} s_k$, then

$$\frac{\|y_k\|^2}{s_k^T y_k} = \frac{z_k^T \bar{G}_k z_k}{z_k^T z_k} \leq M. \tag{5.3.6}$$

In addition, we have

$$\|y_k\| \leq \|\bar{G}\| \|s_k\|, \quad \|s_k\| \leq \|\bar{G}_k^{-1}\| \|y_k\|$$

which give

$$\frac{\|y_k\|}{\|s_k\|} \leq M \tag{5.3.7}$$

and

$$\frac{\|s_k\|}{\|y_k\|} \leq \frac{1}{m}. \tag{5.3.8}$$

Therefore, from the above discussion, we have

Lemma 5.3.2 *Let $f : R^n \to R$ satisfy Assumption 5.3.1. Then*

$$\frac{\|s_k\|}{\|y_k\|}, \frac{\|y_k\|}{\|s_k\|}, \frac{s_k^T y_k}{\|s_k\|^2}, \frac{s_k^T y_k}{\|y_k\|^2}, \frac{\|y_k\|^2}{s_k^T y_k}$$

are bounded.

Lemma 5.3.3 *Under exact line search, $\sum \|s_k\|^2$ and $\sum \|y_k\|^2$ are convergent.*

Proof. Let $\psi(\tau) = f(x_{k+1} - \tau s_k)$. From (5.3.1), it follows that $\psi''(\tau) \geq m\|s_k\|^2$. Note that the exact line search gives $\psi'(0) = 0$. Then we have

$$\psi(\tau) \geq \psi(0) + \frac{1}{2} m \|s_k\|^2 \tau^2.$$

Taking $\tau = 1$, we deduces

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2} m \|s_k\|^2.$$

By summing this expression we have

$$\sum_{k=0}^{\infty} \|s_k\|^2 \leq 2\{f(x_0) - f(x^*)\}/m,$$

which implies $\sum \|s_k\|^2$ is convergent, where $f(x^*)$ is the minimum of $f(x)$.

By Lemma 5.3.2, we also obtain that $\sum \|y_k\|^2$ is convergent. □

Lemma 5.3.4 *For all vectors x, the inequality*

$$\|g(x)\|^2 \geq m[f(x) - f(x^*)] \tag{5.3.9}$$

holds, where $f(x^*)$ is the minimum of $f(x)$.

Proof. Since the function

$$\psi(\tau) = f(x + \tau(x^* - x)),\ (0 \le \tau \le 1)$$

is a convex function, then

$$f(x + \tau(x^* - x)) \ge f(x) + \tau(x^* - x)^T g(x).$$

In particular, set $\tau = 1$, then we have

$$f(x) - f(x^*) \le -(x^* - x)^T g(x) \le \|g(x)\|\|x^* - x\|. \tag{5.3.10}$$

By (5.3.5) and Cauchy-Schwartz inequality, we deduce

$$\begin{aligned} \|x^* - x\|^2 &\le (x^* - x)^T (g(x^*) - g(x))/m \\ &\le \|x^* - x\|\|g(x^*) - g(x)\|/m, \end{aligned} \tag{5.3.11}$$

which gives

$$\|x^* - x\| \le \|g(x^*) - g(x)\|/m = \|g(x)\|/m. \tag{5.3.12}$$

Substituting (5.3.12) into (5.3.10) establishes (5.3.9). □

Theorem 5.3.5 *Suppose that $f(x)$ satisfies Assumption 5.3.1. Then, under exact line search, the sequence $\{x_k\}$ generated by DFP method converges to the minimizer x^* of f.*

Proof. Consider DFP formula of inverse Hessian approximation

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{s_k^T y_k} \tag{5.3.13}$$

and DFP formula of Hessian approximation

$$B_{k+1} = \left(I - \frac{y_k s_k^T}{s_k^T y_k}\right) B_k \left(I - \frac{s_k y_k^T}{s_k^T y_k}\right) + \frac{y_k y_k^T}{s_k^T y_k}. \tag{5.3.14}$$

Obviously, $B_{k+1}H_{k+1} = I$. By computing the trace of (5.3.14), we have

$$\mathrm{Tr}(B_{k+1}) = \mathrm{Tr}(B_k) - 2\frac{s_k^T B_k y_k}{s_k^T y_k} + \frac{(s_k^T B_k s_k)(y_k^T y_k)}{(s_k^T y_k)^2} + \frac{y_k^T y_k}{s_k^T y_k}. \tag{5.3.15}$$

The middle two terms can be written as

$$\begin{aligned} & -2\frac{s_k^T B_k y_k}{s_k^T y_k} + \frac{(s_k^T B_k s_k)(y_k^T y_k)}{(s_k^T y_k)^2} \\ = \; & \alpha_k \left[\frac{2g_k^T y_k}{s_k^T y_k} + \frac{(-g_k^T s_k)(y_k^T y_k)}{(s_k^T y_k)^2} \right] \\ = \; & \alpha_k \frac{2g_k^T y_k + y_k^T y_k}{s_k^T y_k} \\ = \; & \frac{\|g_{k+1}\|^2 - \|g_k\|^2}{g_k^T H_k g_k}. \end{aligned} \tag{5.3.16}$$

Since $g_{k+1}^T s_k = 0$, then

$$\begin{aligned} g_{k+1}^T H_{k+1} g_{k+1} &= g_{k+1}^T \left[H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} \right] g_{k+1} \\ &= g_k^T \left[H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} \right] g_k \\ &= g_k^T \left[H_k - \frac{H_k g_k g_k^T H_k}{y_k^T H_k y_k} \right] g_k \\ &= \frac{(g_k^T H_k g_k)(g_{k+1}^T H_k g_{k+1})}{g_k^T H_k g_k + g_{k+1}^T H_k g_{k+1}}. \end{aligned}$$

By finding the inverse number of the above expression, we get

$$\frac{1}{g_{k+1}^T H_{k+1} g_{k+1}} = \frac{1}{g_{k+1}^T H_k g_{k+1}} + \frac{1}{g_k^T H_k g_k}. \tag{5.3.17}$$

Using (5.3.16) and (5.3.17), then (5.3.15) becomes

$$\begin{aligned} \mathrm{Tr}(B_{k+1}) &= \mathrm{Tr}(B_k) + \frac{\|g_{k+1}\|^2}{g_{k+1}^T H_{k+1} g_{k+1}} - \frac{\|g_k\|^2}{g_k^T H_k g_k} \\ & \quad - \frac{\|g_{k+1}\|^2}{g_{k+1}^T H_k g_{k+1}} + \frac{\|y_k\|^2}{s_k^T y_k}. \end{aligned} \tag{5.3.18}$$

By recurrence, we obtain

$$\begin{aligned} \mathrm{Tr}(B_{k+1}) &= \mathrm{Tr}(B_0) + \frac{\|g_{k+1}\|^2}{g_{k+1}^T H_{k+1} g_{k+1}} - \frac{\|g_0\|^2}{g_0^T H_0 g_0} \\ & \quad - \sum_{j=0}^{k} \frac{\|g_{j+1}\|^2}{g_{j+1}^T H_j g_{j+1}} + \sum_{j=0}^{k} \frac{\|y_j\|^2}{s_j^T y_j}. \end{aligned} \tag{5.3.19}$$

Therefore, by Lemma 5.3.2, there exists a positive number M which is independent of k, such that

$$\mathrm{Tr}(B_{k+1}) \leq \frac{\|g_{k+1}\|^2}{g_{k+1}^T H_{k+1} g_{k+1}} - \sum_{j=0}^{k} \frac{\|g_{j+1}\|^2}{g_{j+1}^T H_j g_{j+1}} + Mk. \tag{5.3.20}$$

In the left part, we will prove that if the theorem does not hold, then the sum of the last two terms in (5.3.20) is negative.

Now consider the trace of H_{k+1}. From (5.3.13), we have

$$\mathrm{Tr}(H_{k+1}) = \mathrm{Tr}(H_0) - \sum_{j=0}^{k} \frac{\|H_j y_j\|^2}{y_j^T H_j y_j} + \sum_{j=0}^{k} \frac{\|s_j\|^2}{s_j^T y_j}. \tag{5.3.21}$$

Since H_{k+1} is positive definite, the right-hand side of (5.3.21) is positive. By Lemma 5.3.2, there exists $m > 0$ which is independent of k, such that

$$\sum_{j=0}^{k} \frac{\|H_j y_j\|^2}{y_j^T H_j y_j} < \frac{k}{m}. \tag{5.3.22}$$

Note that

$$(y_j^T H_j y_j)^2 \leq \|H_j y_j\|^2 \|y_j\|^2 \tag{5.3.23}$$

and

$$\begin{aligned} y_j^T H_j y_j &= g_{j+1}^T H_j g_{j+1} + g_j^T H_j g_j + 2 g_{j+1}^T d_j \\ &> g_{j+1}^T H_j g_{j+1} \end{aligned} \tag{5.3.24}$$

by the positive definiteness of H_j and exact line search, then by using (5.3.24), (5.3.23) and (5.3.22) in turn, we obtain

$$\sum_{j=0}^{k} \frac{g_{j+1}^T H_j g_{j+1}}{\|y_j\|^2} \leq \sum_{j=0}^{k} \frac{y_j^T H_j y_j}{\|y_j\|^2} \leq \sum_{j=0}^{k} \frac{\|H_j y_j\|^2}{y_j^T H_j y_j} \leq \frac{k}{m}. \tag{5.3.25}$$

By using Cauchy-Schwartz inequality and (5.3.25)

$$\begin{aligned} \sum_{j=0}^{k} \frac{\|g_{j+1}\|^2}{g_{j+1}^T H_j g_{j+1}} &\geq \left(\sum_{j=0}^{k} \frac{\|g_{j+1}\|}{\|y_j\|} \right)^2 \Big/ \sum_{j=0}^{k} \frac{g_{j+1}^T H_j g_{j+1}}{\|y_j\|^2} \\ &\geq \frac{m}{k} \left(\sum_{j=0}^{k} \frac{\|g_{j+1}\|}{\|y_j\|} \right)^2. \end{aligned} \tag{5.3.26}$$

Now suppose that the theorem is not true, that is, there exists $\delta > 0$ such that for all sufficiently large k,

$$\|g_k\| \geq \delta. \tag{5.3.27}$$

Also, by (5.3.11) and Theorem 2.2.9, there exists a constant $\eta > 0$ such that

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2}\eta\|s_k\|^2,$$

which gives $\|s_k\| \to 0$ and further $\|y_k\| \to 0$. Then, by (5.3.26) and (5.3.27), we deduce, for k sufficiently large, that

$$\sum_{j=0}^{k} \frac{\|g_{j+1}\|^2}{g_{j+1}^T H_j g_{j+1}} > Mk. \tag{5.3.28}$$

The above inequality implies that the sum of the last two terms in (5.3.20) is negative.

By (5.3.28) and (5.3.20), we immediately obtain

$$\mathrm{Tr}(B_{k+1}) < \frac{\|g_{k+1}\|^2}{g_{k+1}^T H_{k+1} g_{k+1}}. \tag{5.3.29}$$

Note that, for a symmetric and positive definite matrix, the inverse of trace is the lower bound of the least eigenvalue of inverse of the matrix. Then, it follows from (5.3.29) that

$$\frac{g_{k+1}^T H_{k+1} g_{k+1}}{\|g_{k+1}\|^2} < \mu, \tag{5.3.30}$$

where μ is the lower bound of the least eigenvalue of H_{k+1}. However, from Theorem 1.2.10 on the property of Rayleigh quotient, we have

$$\frac{g_{k+1}^T H_{k+1} g_{k+1}}{\|g_{k+1}\|^2} > \mu, \tag{5.3.31}$$

which contradicts (5.3.30). This contradiction proves that $\{x_k\}$ converges to x^* and that our theorem holds. □

5.3.2 Global Convergence under Inexact Line Search

Now, we turn to study the global convergence of BFGS method under inexact line search.

Let us rewrite BFGS method as follows:

$$x_{k+1} = x_k + s_k = x_k + \alpha_k d_k = x_k - \alpha_k B_k^{-1} g_k, \tag{5.3.32}$$

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}. \tag{5.3.33}$$

Theorem 5.3.6 *Let x_0 and B_0 be a starting point and a symmetric positive definite initial matrix, respectively. Suppose that $f(x)$ satisfies Assumption 5.3.1. Then, under Wolfe-Powell inexact line search (2.5.3) and (2.5.7), the sequence $\{x_k\}$ generated by BFGS method converges to the minimizer x^* of f.*

Proof. By computing the trace and determinant of BFGS formula (5.3.33), we obtain that

$$\mathrm{Tr}(B_{k+1}) = \mathrm{Tr}(B_k) - \frac{\|B_k s_k\|}{s_k^T B_k s_k} + \frac{\|y_k\|^2}{y_k^T s_k} \tag{5.3.34}$$

and

$$\det(B_{k+1}) = \det(B_k) \frac{y_k^T s_k}{s_k^T B_k s_k}. \tag{5.3.35}$$

Let us define

$$m_k = \frac{y_k^T s_k}{s_k^T s_k}, \ M_k = \frac{y_k^T y_k}{y_k^T s_k}. \tag{5.3.36}$$

It follows from (5.3.4) and (5.3.6) that

$$m \le m_k \le M, \ m \le M_k \le M. \tag{5.3.37}$$

Let us also define

$$\cos\theta_k = \frac{s_k^T B_k s_k}{\|s_k\| \|B_k s_k\|}, \ q_k = \frac{s_k^T B_k s_k}{s_k^T s_k}. \tag{5.3.38}$$

We then obtain that

$$\frac{\|B_k s_k\|^2}{s_k^T B_k s_k} = \frac{\|B_k s_k\|^2 \|s_k\|^2}{(s_k^T B_k s_k)^2} \frac{s_k^T B_k s_k}{\|s_k\|^2} = \frac{q_k}{\cos^2\theta_k}. \tag{5.3.39}$$

In addition, we have from (5.3.36) that

$$\det(B_{k+1}) = \det(B_k)\frac{y_k^T s_k}{s_k^T s_k}\frac{s_k^T s_k}{s_k^T B_k s_k} = \det(B_k)\frac{m_k}{q_k}. \tag{5.3.40}$$

Now we introduce the following function of a positive definite matrix B_k:

$$\psi(B_k) = \mathrm{Tr}(B_k) - \ln(\det(B_k)), \tag{5.3.41}$$

where $\ln(\cdot)$ denotes the natural logarithm. It is not difficult to show that $\psi(B_k) > 0$. By using (5.3.34)–(5.3.41), we have that

$$\begin{aligned}\psi(B_{k+1}) &= \mathrm{Tr}(B_k) + M_k - \frac{q_k}{\cos^2\theta_k} - \ln(\det(B_k)) - \ln m_k + \ln q_k \\ &= \psi(B_k) + (M_k - \ln m_k - 1) \\ &\quad + \left[1 - \frac{q_k}{\cos^2\theta_k} + \ln\frac{q_k}{\cos^2\theta_k}\right] + \ln\cos^2\theta_k. \end{aligned} \tag{5.3.42}$$

Note that the function $h(t) = 1 - t + \ln t \le 0$ for all $t > 0$. Hence the term inside the square brackets is nonpositive, and thus by summing both sides of (5.3.42), we have

$$0 < \psi(B_{k+1}) \le \psi(B_1) + ck + \sum_{j=1}^{k} \ln\cos^2\theta_j, \tag{5.3.43}$$

where the constant $c = M - \ln m - 1$ is assumed to be positive without loss of generality.

From Theorem 2.5.5, we have

$$\lim_{k\to\infty} \|g_k\|\cos\theta_k = 0. \tag{5.3.44}$$

If θ_k is bounded away from 90°, there is a positive constant δ such that

$$\cos\theta_k > \delta > 0, \quad \text{for } k \text{ sufficiently large,}$$

and thus we have our result.

Now assume, by contradiction, that $\cos\theta_k \to 0$. Then there exists $k_1 > 0$ such that for all $j > k_1$, we have

$$\ln\cos^2\theta_j < -2c,$$

where c is the constant defined above.

By using (5.3.43), we deduce, for all $k > k_1$, that

$$\begin{aligned} 0 &< \psi(B_1) + ck + \sum_{j=1}^{k_1} \ln \cos^2 \theta_j + \sum_{j=k_1+1}^{k} (-2c) \\ &= \psi(B_1) + \sum_{j=1}^{k_1} \ln \cos^2 \theta_j + 2ck_1 - ck \\ &< 0, \end{aligned}$$

which gives a contradiction. Therefore the assumption $\cos \theta_j \to 0$ is not true, and there exists a subsequence $\{j_k\}$ such that

$$\{\cos \theta_{j_k}\} \geq \delta > 0,$$

which means

$$\liminf \|\nabla f(x_k)\| = 0. \tag{5.3.45}$$

Since the problem is strong convex, then (5.3.45) implies $x_k \to x^*$. □

5.4 Local Convergence of Quasi-Newton Methods

In this section, we discuss local convergence of quasi-Newton methods. The convergence analysis in this section mainly makes use of Broyden, Dennis, and Moré [29], Dennis and Moré [91], Dennis and Moré [92], Nocedal and Wright [233] and others. In §5.4.1 we first consider solving $F(x) = 0$. The necessary and sufficient condition of superlinear convergence for solving $F(x) = 0$ is given in Theorem 5.4.3, which is basic and the most important in convergence analysis for quasi-Newton methods. Theorem 5.4.4 is a corollary of Theorem 5.4.3, and Lemma 5.4.5 gives the geometry of superlinear convergence for quasi-Newton methods. Then, we generalize the above results to minimization problems. We give superlinear convergence results in the case of basic iteration, exact line search and inexact line search respectively in Theorem 5.4.6, Theorem 5.4.7 and Theorem 5.4.8. In §5.4.2, we give linear convergence of general quasi-Newton methods by means of the bounded deterioration principle. In §5.4.3 the linear and superlinear convergence of SR1 method is established. In §5.5.4, we discuss the linear convergence of DFP method. In §5.4.5 and 5.4.6. we give the superlinear convergence results of BFGS and DFP methods respectively by different techniques. Finally, in §5.4.7, the local convergence of Broyden's class methods is discussed.

5.4.1 Superlinear Convergence of General Quasi-Newton Methods

First, we consider

$$F(x) = 0, \tag{5.4.1}$$

where $F : R^n \to R^n$ is a mapping. In convergence analysis, we often need the following assumption.

Assumption 5.4.1 *(a) $F : R^n \to R^n$ is continuously differentiable on an open convex set $D \subset R^n$.*

(b) There is $x^* \in D$ with $F(x^*) = 0$ and $F'(x^*)$ nonsingular.

(c) F' is Lipschitzian at x^*, i.e., there is a constant γ such that

$$\|F'(x) - F'(x^*)\| \leq \gamma \|x - x^*\|, \ x \in D.$$

Second, we consider the minimization problem

$$\min_{x \in R^n} f(x). \tag{5.4.2}$$

If in Assumption 5.4.1, we replace $F(x)$ and $F'(x)$ by $g(x)$ and $\nabla^2 f(x)$ respectively, we get the following assumption for optimization problem (5.4.2):

Assumption 5.4.2 *(a) $f : R^n \to R$ is twice continuously differentiable on an open convex set $D \subset R^n$.*

(b) There is a strong local minimizer $x^* \in D$ with $\nabla^2 f(x^*)$ symmetric and positive definite.

(c) There is a neighborhood $N(x^*, \varepsilon)$ of x^* such that

$$\|\nabla^2 f(\bar{x}) - \nabla^2 f(x)\| \leq \gamma \|\bar{x} - x\|, \ \forall x, \bar{x} \in N(x^*, \varepsilon).$$

Superlinear Convergence: Nonlinear System

We begin our discussion on a basic necessary and sufficient condition of superlinear convergence for a nonlinear system.

Theorem 5.4.3 *Let $F : R^n \to R^n$ satisfy (a) and (b) in Assumption 5.4.1. Let $\{B_k\}$ be a sequence of nonsingular matrices. Suppose, for $x_0 \in D$, that the iterates generated by*

$$x_{k+1} = x_k - B_k^{-1} F(x_k) \tag{5.4.3}$$

remain in D. $x_k \neq x^*$ $(\forall k \geq 0)$. Suppose also that $\{x_k\}$ converges to x^*. Then $\{x_k\}$ converges to x^* at a superlinear rate if and only if

$$\lim_{k \to +\infty} \frac{\|[B_k - F'(x^*)](x_{k+1} - x_k)\|}{\|x_{k+1} - x_k\|} = 0. \tag{5.4.4}$$

Proof. Our idea is to prove the following equivalence:

$$\begin{aligned} \lim_{k\to\infty} \frac{\|[B_k - F'(x^*)]s_k\|}{\|s_k\|} = 0 \quad &\Leftrightarrow \quad \lim_{k\to\infty} \frac{\|F(x_{k+1})\|}{\|s_k\|} = 0 \\ &\Leftrightarrow \quad \lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0, \end{aligned} \tag{5.4.5}$$

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

First, suppose (5.4.4) holds. By (5.4.3), we have

$$\begin{aligned} & [B_k - F'(x^*)](x_{k+1} - x_k) \\ = \; & -F(x_k) - F'(x^*)(x_{k+1} - x_k) \\ = \; & [F(x_{k+1}) - F(x_k) - F'(x^*)(x_{k+1} - x_k)] - F(x_{k+1}). \end{aligned} \tag{5.4.6}$$

By taking the norm, dividing by $\|s_k\|$, and using Theorem 1.2.24, we obtain

$$\begin{aligned} \tfrac{\|F(x_{k+1})\|}{\|s_k\|} \; &\leq \frac{\|(B_k - F'(x^*))s_k\|}{\|s_k\|} + \frac{\|F(x_{k+1}) - F(x_k) - F'(x^*)s_k\|}{\|s_k\|} \\ &\leq \frac{\|(B_k - F'(x^*))s_k\|}{\|s_k\|} + \frac{\gamma}{2}(\|x_k - x^*\| + \|x_{k+1} - x^*\|). \end{aligned} \tag{5.4.7}$$

Since $\lim_{k\to\infty} x_k = x^*$, it follows from (5.4.4) that

$$\lim_{k\to\infty} \frac{\|F(x_{k+1})\|}{\|s_k\|} = 0. \tag{5.4.8}$$

Since also $\lim_{k\to\infty} \|s_k\| = 0$, we have

$$F(x^*) = \lim_{k\to\infty} F(x_k) = 0.$$

Noting that $F'(x^*)$ is nonsingular, it follows from Theorem 1.2.25 that there is a $\beta > 0$ and $k_0 \geq 0$ such that $\forall k \geq k_0$, we have

$$\|F(x_{k+1})\| = \|F(x_{k+1}) - F(x^*)\| \geq \beta \|x_{k+1} - x^*\|.$$

Thus

$$\frac{\|F(x_{k+1})\|}{\|x_{k+1}-x_k\|} \geq \frac{\beta\|x_{k+1}-x^*\|}{\|x_{k+1}-x^*\|+\|x_k-x^*\|} = \beta\frac{r_k}{1+r_k}, \tag{5.4.9}$$

where

$$r_k = \frac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|}.$$

Combining (5.4.8) and (5.4.9) implies that

$$\frac{r_k}{1+r_k} \to 0$$

which gives

$$\lim_{k\to\infty} r_k = 0, \tag{5.4.10}$$

i.e., the sequence $\{x_k\}$ is convergent to x^* superlinearly.

Conversely, assume that $\{x_k\}$ converges superlinearly to x^* and $F(x^*) = 0$. By Theorem 1.2.25, there exist $\bar{\beta} > 0$ and $k_0 \geq 0$, such that $\forall k \geq k_0$, we have

$$\|F(x_{k+1})\| \leq \bar{\beta}\|x_{k+1}-x^*\|.$$

Since $\{x_k\}$ is convergent superlinearly, we have

$$\begin{aligned} 0 &= \lim_{k\to\infty}\frac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|} \geq \lim_{k\to\infty}\frac{\|F(x_{k+1})\|}{\bar{\beta}\|x_k-x^*\|} \\ &= \lim_{k\to\infty}\frac{1}{\bar{\beta}}\frac{\|F(x_{k+1})\|}{\|x_{k+1}-x_k\|}\frac{\|x_{k+1}-x_k\|}{\|x_k-x^*\|}. \end{aligned}$$

By use of Theorem 1.5.2 giving $\lim_{k\to\infty}\|x_{k+1}-x_k\|/\|x_k-x^*\| = 1$, we obtain

$$\lim_{k\to\infty}\frac{\|F(x_{k+1})\|}{\|x_{k+1}-x_k\|} = 0,$$

which gives (5.4.4) by means of (5.4.6). □

Theorem 5.4.3 indicates that if B_k converges to $F'(x^*)$ along the direction s_k, then quasi-Newton methods converge superlinearly. This theorem is very important in analysis of quasi-Newton methods. Equation (5.4.4) is called the Dennis-Moré characterization of superlinear convergence. The following theorem shows, for the iteration (5.4.11), that the method is convergent superlinearly if and only if the sequence of steplength factors $\{\alpha_k\}$ converges to 1. The proof of Theorem 5.4.4 is completed by use of Theorem 5.4.3.

Theorem 5.4.4 *Let $F : R^n \to R^n$ satisfy the assumptions of Theorem 5.4.3. Let $\{B_k\}$ be a sequence of nonsingular matrices. Suppose, for $x_0 \in D$, that the iteration*

$$x_{k+1} = x_k - \alpha_k B_k^{-1} F(x_k) \tag{5.4.11}$$

remains in D and $\{x_k\}$ converges to x^*. If (5.4.4) holds, then $\{x_k\}$ converges to x^* superlinearly and $F(x^*) = 0$ if and only if $\{\alpha_k\}$ converges to 1.

Proof. Necessity. Suppose that $\{x_k\}$ converges to x^* superlinearly and $F(x^*) = 0$. By Theorem 5.4.3, we have

$$\lim_{k\to\infty} \frac{\|[\alpha_k^{-1} B_k - F'(x^*)](x_{k+1} - x_k)\|}{\|x_{k+1} - x_k\|} = 0. \tag{5.4.12}$$

So, (5.4.4) implies that

$$\lim_{k\to\infty} \|(\alpha_k^{-1} - 1)B_k(x_{k+1} - x_k)\|/\|x_{k+1} - x_k\| = 0.$$

Since $B_k(x_{k+1} - x_k) = -\alpha_k F(x_k)$, the above equality can be written as

$$\lim_{k\to\infty} \|(\alpha_k - 1)F(x_k)\|/\|x_{k+1} - x_k\| = 0. \tag{5.4.13}$$

Noting that $F'(x^*)$ is nonsingular, it follows from Theorem 1.2.25 that there exists $\beta > 0$ such that $\|F(x_k)\| \geq \beta\|x_k - x^*\|$. Then, from (5.4.13), we obtain

$$\lim_{k\to\infty} |\alpha_k - 1| \frac{\beta\|x_k - x^*\|}{\|x_{k+1} - x_k\|} = 0. \tag{5.4.14}$$

Since also $\{x_k\}$ is convergent superlinearly, i.e., $\lim_{k\to\infty} \|x_{k+1} - x_k\|/\|x_k - x^*\| = 1$, we obtain immediately from (5.4.14) that $\{\alpha_k\} \to 1$.

Sufficiency. Suppose that $\{\alpha_k\} \to 1$. It follows from (5.4.4) that (5.4.12) holds. Therefore, from Theorem 5.4.3, we obtain that $\{x_k\}$ converges to x^* superlinearly and $F(x^*) = 0$. □

This theorem suggests that when a method is required to be superlinearly convergent, we should ask for $\alpha_k \to 1$ as $k \to \infty$.

Next, we interpret the geometry of superlinear convergence of quasi-Newton methods, which is an equivalent and geometric representation of (5.4.4).

Let $s_k = x_{k+1} - x_k$. Let also Newton's iteration be $s_k^N = -F'(x_k)^{-1}F(x_k)$. Since $F(x_k) = -B_k s_k$, then

$$s_k - s_k^N = s_k + F'(x_k)^{-1}F(x_k) = F'(x_k)^{-1}[F'(x_k) - B_k]s_k. \tag{5.4.15}$$

By use of Assumption 5.4.1, we have that $\|F'(x_k)^{-1}\|$ is bounded above for x_k sufficiently close to x^*. Thus,

$$F'(x_k)^{-1}[F'(x_k) - B_k]s_k = O(\|[F'(x_k) - B_k]s_k\|) = o(\|s_k\|),$$

where we have used (5.4.4). Therefore (5.4.15) is equivalent to

$$\lim_{k\to\infty} \frac{\|s_k - s_k^N\|}{\|s_k\|} = 0. \tag{5.4.16}$$

The above (5.4.16) indicates that when $\{x_k\}$ converges superlinearly, the relative error of s_k should tend to zero. It is not difficult to prove that (5.4.16) is equivalent to the fact that s_k tends to s_k^N in both direction and length. For this, we introduce the following lemma.

Lemma 5.4.5 *Let $u, v \in R^n, u, v \neq 0$, and $\alpha \in (0, 1)$. If $\|u - v\| \leq \alpha\|u\|$, then $\langle u, v\rangle$ is positive and*

$$\left|1 - \frac{\|v\|}{\|u\|}\right| \leq \alpha,\ 1 - \left(\frac{\langle u, v\rangle}{\|u\|\|v\|}\right)^2 \leq \alpha^2. \tag{5.4.17}$$

Conversely, if $\langle u, v\rangle$ is positive and (5.4.17) holds, then

$$\|u - v\| \leq 3\alpha\|u\|. \tag{5.4.18}$$

Proof. First, assume that $\|u - v\| \leq \alpha\|u\|$. Then

$$\left|\frac{\|u\| - \|v\|}{\|u\|}\right| \leq \frac{\|u - v\|}{\|u\|} \leq \alpha,$$

which implies that the first inequality in (5.4.17) holds.

Let $\omega = \langle u, v\rangle/(\|u\|\|v\|)$. Since

$$\|v\|^2 - 2\langle u, v\rangle + \frac{\langle u, v\rangle^2}{\|v\|^2} = \left[\|v\| - \frac{\langle u, v\rangle}{\|v\|}\right]^2 \geq 0,$$

then

$$\|v\|^2 - 2\langle u, v\rangle \geq -\frac{\langle u, v\rangle^2}{\|v\|^2}.$$

So,

$$\begin{aligned} \|u - v\|^2 &= \|u\|^2 - 2\|u\|\|v\|\omega + \|v\|^2 \qquad (5.4.19) \\ &\geq \|u\|^2 - \frac{\langle u, v\rangle^2}{\|v\|^2} \\ &= \|u\|^2(1 - \omega^2). \qquad (5.4.20) \end{aligned}$$

Therefore,

$$1 - \omega^2 \leq \frac{\|u - v\|^2}{\|u\|^2} \leq \alpha^2$$

giving the second inequality of (5.4.17). In addition, if $\omega \leq 0$, it follows from (5.4.19) that $\|u - v\| \geq \|u\|$, and therefore $\alpha \geq 1$. Hence, if $\alpha < 1$, we have that $\langle u, v\rangle$ is positive.

Conversely, if $\langle u, v\rangle$ is positive and (5.4.17) holds, then by using (5.4.17) and some manipulations, we obtain

$$\begin{aligned} \|u - v\|^2 &= (\|u\| - \|v\|)^2 + 2(1 - \omega)\|u\|\|v\| \\ &\leq \alpha^2\|u\|^2[1 + 2(1 + \alpha)], \end{aligned}$$

which gives (5.4.18) since $\alpha < 1$. □

If (5.4.16) holds, we have, for given $\varepsilon \in (0, 1)$, that

$$\|s_k - s_k^N\| \leq \varepsilon\|s_k\|$$

when $k \geq k_0$. So, by Lemma 5.4.5, it follows that if $\langle s_k, s_k^N\rangle > 0$ and $k \geq k_0$, we have

$$\left|1 - \frac{\|s_k^N\|}{\|s_k\|}\right| \leq \varepsilon$$

and

$$1 - \left(\frac{\langle s_k, s_k^N\rangle}{\|s_k\|\|s_k^N\|}\right)^2 \leq \varepsilon^2.$$

They show that (5.4.16) is equivalent to

$$\lim_{k\to\infty} \frac{\|s_k^N\|}{\|s_k\|} = \lim_{k\to\infty} \left\langle \frac{s_k}{\|s_k\|}, \frac{s_k^N}{\|s_k^N\|} \right\rangle = 1. \qquad (5.4.21)$$

Therefore we have a conclusion: the necessary and sufficient condition of superlinear convergence of quasi-Newton method is that s_k approaches s_k^N in both length and direction.

Superlinear Convergence: Minimization Problem

Next, we consider minimization problem (5.4.2) and discuss the superlinear convergence in the case of basic iteration, exact line search, and inexact line search.

Completely similar to Theorem 5.4.3, for minimization problem (5.4.2), we have

Theorem 5.4.6 *Let $f : R^n \to R$ satisfy the assumptions (a) and (b) in Assumption 5.4.2. Consider iteration sequence*

$$x_{k+1} = x_k - B_k^{-1} g_k, \tag{5.4.22}$$

where $\{B_k\}$ is a sequence of symmetric and positive definite matrices. Assume that $\{x_k\}$ converges to x^*. Then $\{x_k\}$ converges superlinearly to x^* if and only if

$$\lim_{k\to\infty} \frac{\|[B_k - \nabla^2 f(x^*)]s_k\|}{\|s_k\|} = 0. \tag{5.4.23}$$

Proof. The proof is the same as for Theorem 5.4.3. □

The following Theorem 5.4.7 shows the superlinear convergence of quasi-Newton method in the case of exact line search.

Theorem 5.4.7 *Let $f : R^n \to R$ satisfy conditions (a) and (b) in Assumption 5.4.2. Suppose $\{B_k\}$ is a sequence of symmetric and positive definite matrices. Consider, for a given $x_0 \in D$, the iteration*

$$x_{k+1} = x_k - \alpha_k B_k^{-1} g_k, \tag{5.4.24}$$

where α_k is determined by exact line search. If the sequence $\{x_k\}$ provided by (5.4.24) remains in D and $x_k \neq x^*$ $(\forall k \geq 0)$, and if $x_k \to x^*$, then when

$$\lim_{k\to\infty} \frac{\|[B_k - \nabla^2 f(x^*)]s_k\|}{\|s_k\|} = 0, \tag{5.4.25}$$

we have $\alpha_k \to 1$ and $g(x^*) = 0$, hence $\{x_k\}$ converges to x^* superlinearly.

Proof. It is enough to prove $\alpha_k \to 1$ when (5.4.25) holds. Other conclusions can be obtained direct from Theorem 5.4.4.

Since $\nabla^2 f(x^*)$ is positive definite, there exists $m > 0$ such that

$$s_k^T \nabla^2 f(x^*) s_k \geq m\|s_k\|^2.$$

Therefore we only need to prove

$$(\alpha_k - 1)s_k^T \nabla^2 f(x^*) s_k = o(\|s_k\|^2). \tag{5.4.26}$$

From (1.2.111), we have

$$\|g_{k+1} - g_k - \nabla^2 f(x^*) s_k\| \leq \max_{0 \leq t \leq 1} \|\nabla^2 f(x_k + t s_k) - \nabla^2 f(x^*)\| \|s_k\|.$$

Then from $x_k \to x^*$ and the continuity of $\nabla^2 f(x)$, we obtain

$$\|g_{k+1} - g_k - \nabla^2 f(x^*) s_k\| = o(\|s_k\|)$$

which implies

$$g_{k+1}^T s_k - g_k^T s_k - s_k^T \nabla^2 f(x^*) s_k = o(\|s_k\|^2). \tag{5.4.27}$$

Since α_k is a steplength from exact line search, $g_{k+1}^T s_k = 0$. Also, noting that $B_k s_k = \alpha_k B_k d_k = -\alpha_k g_k$, we may write (5.4.27) as

$$\begin{aligned} s_k^T \nabla^2 f(x^*) s_k &= -g_k^T s_k + o(\|s_k\|^2) \\ &= \frac{1}{\alpha_k} s_k^T B_k s_k + o(\|s_k\|^2). \end{aligned} \tag{5.4.28}$$

From (5.4.25), we have

$$s_k^T [B_k - \nabla^2 f(x^*)] s_k = o(\|s_k\|^2). \tag{5.4.29}$$

So, combining (5.4.28) and (5.4.29) gives

$$\begin{aligned} (\alpha_k - 1) s_k^T \nabla^2 f(x^*) s_k &= s_k^T [B_k - \nabla^2 f(x^*)] s_k + o(\|s_k\|^2) \\ &= o(\|s_k\|^2) \end{aligned}$$

which proves (5.4.26). □

About inexact line search, we consider Wolfe-Powell rule (2.5.3) and (2.5.7). By use of $d_k = -B_k g_k$, we employ the following rule: if

$$f(x_k - B_k^{-1} g_k) \leq f(x_k) - \rho g_k^T B_k^{-1} g_k, \tag{5.4.30}$$

$$g(x_k - B_k^{-1} g_k)^T B_k^{-1} g_k \leq \sigma g_k^T B_k^{-1} g_k \tag{5.4.31}$$

hold, take $\alpha_k = 1$; otherwise, take $\alpha_k > 0$ such that

$$f(x_k - \alpha_k B_k^{-1} g_k) \leq f(x_k) - \rho \alpha_k g_k^T B_k^{-1} g_k, \tag{5.4.32}$$

$$g(x_k - \alpha_k B_k^{-1} g_k)^T B_k^{-1} g_k \leq \sigma g_k^T B_k^{-1} g_k, \tag{5.4.33}$$

where $g(\cdot) = \nabla f(\cdot)$.

Theorem 5.4.8 *Let $f : R^n \to R$ satisfy conditions (a) and (b) in Assumption 5.4.2. Suppose that $\{B_k\}$ is a sequence of symmetric and positive definite matrices. For given $x_0 \in D$, consider the iteration (5.4.24), where α_k is determined by Wolfe-Powell rule (5.4.30)–(5.4.33). If the sequence $\{x_k\}$ produced by (5.4.24) remains in D and $x_k \neq x^*$ $(\forall k \geq 0)$ and if $x_k \to x^*$, then when (5.4.25) holds, $\alpha_k \to 1$ and hence $\{x_k\}$ converges to x^* superlinearly.*

Proof. Now we only need to prove that for sufficiently large k, (5.4.30)–(5.4.33) hold, and thus $\alpha_k = 1$. The remainder is obtained from Theorem 5.4.4.

Since $B_k s_k = -\alpha_k g_k$, it follows from (5.4.25) that

$$\begin{aligned} 0 &= \lim_{k\to\infty} \frac{\|[B_k - \nabla^2 f(x^*)]s_k\|}{\|s_k\|} \\ &= \lim_{k\to\infty} \frac{\|g_k - \nabla^2 f(x^*)B_k^{-1}g_k\|}{\|B_k^{-1}g_k\|}. \end{aligned}$$

Then

$$\begin{aligned} & g_k^T B_k^{-1} g_k - (B_k^{-1}g_k)^T \nabla^2 f(x^*)(B_k^{-1}g_k) \\ = \;& (g_k - \nabla^2 f(x^*)B_k^{-1}g_k)^T (B_k^{-1}g_k) \\ = \;& o(\|B_k^{-1}g_k\|^2), \end{aligned}$$

that is

$$g_k^T B_k^{-1} g_k = (B_k^{-1}g_k)^T \nabla^2 f(x^*)(B_k^{-1}g_k) + o(\|B_k^{-1}g_k\|^2). \tag{5.4.34}$$

Since $\nabla^2 f(x^*)$ is positive definite, there exists $\eta > 0$ such that for sufficiently large k,

$$g_k^T B_k^{-1} g_k \geq \eta \|B_k^{-1} g_k\|^2. \tag{5.4.35}$$

Then, from Taylor's expansion (1.2.103) and (5.4.34), we have

$$\begin{aligned} f(x_k - B_k^{-1}g_k) - f(x_k) &= -g_k^T B_k^{-1} g_k + \frac{1}{2} g_k^T B_k^{-1} g_k + o(\|B_k^{-1}g_k\|^2) \\ &= -\frac{1}{2} g_k^T B_k^{-1} g_k + o(\|B_k^{-1}g_k\|^2) \\ &\leq -\rho g_k^T B_k^{-1} g_k. \end{aligned} \tag{5.4.36}$$

Also, by (1.2.111) and a proof similar to (5.4.27), we get

$$\begin{aligned} & g(x_k - B_k^{-1}g_k)^T B_k^{-1}g_k - g_k^T B_k^{-1}g_k + (B_k^{-1}g_k)^T \nabla^2 f(x^*)(B_k^{-1}g_k) \\ = \; & o(\|B_k^{-1}g_k\|^2), \end{aligned}$$

which, together with (5.4.34), gives

$$g(x_k - B_k^{-1}g_k)^T B_k^{-1}g_k = o(\|B_k^{-1}g_k\|^2) \leq \sigma g_k^T B_k^{-1}g_k. \tag{5.4.37}$$

It follows from (5.4.36) and (5.4.37) that (5.4.30)-(5.4.31) hold, and thus $\alpha_k = 1$ for k sufficiently large. □

5.4.2 Linear Convergence of General Quasi-Newton Methods

In this subsection, our goal is to discuss the local and linear convergence results of general quasi-Newton methods. Let the iterative scheme of general quasi-Newton methods be

$$x_{k+1} = x_k - B_k^{-1}F(x_k), \tag{5.4.38}$$
$$B_{k+1} \in U(x_k, B_k), \tag{5.4.39}$$

where $U(x_k, B_k)$ denotes a nonempty set of updates, $(x_k, B_k) \in \text{dom}U$, dom$U$ denotes the domain of U.

Theorem 5.4.9 *Let $F : R^n \to R^n$ satisfy the assumptions (a), (b) and (c) in Assumption 5.4.1, U an update function, such that for all $(x_k, B_k) \in domU$ and $B_{k+1} \in U(x_k, B_k)$, we have that*

$$\|B_{k+1} - F'(x^*)\| \leq \|B_k - F'(x^*)\| + \frac{\gamma}{2}(\|x_{k+1} - x^*\| + \|x_k - x^*\|), \tag{5.4.40}$$

where γ is some constant, or that

$$\|B_{k+1} - F'(x^*)\| \leq [1 + \alpha_1\sigma(x_k, x_{k+1})]\|B_k - F'(x^*)\| + \alpha_2\sigma(x_k, x_{k+1}), \tag{5.4.41}$$

where α_1 and α_2 are some constants, and

$$\sigma(x_k, x_{k+1}) = \max\{\|x_k - x^*\|, \|x_{k+1} - x^*\|\}. \tag{5.4.42}$$

Then, there exist constants ε and δ, such that, for $\|x_0 - x^*\| < \varepsilon$ and $\|B_0 - F'(x^*)\| < \delta$, the iteration (5.4.38)–(5.4.39) is well-defined, and $\{x_k\}$ converges to x^* linearly.

Proof. First, we prove the conclusion for the given condition (5.4.40). Assume $\|F'(x^*)^{-1}\| \leq \beta$ and choose ε and δ such that

$$6\beta\delta < 1, \tag{5.4.43}$$

$$3\gamma\varepsilon \leq 2\delta. \tag{5.4.44}$$

To prove the local and linear convergence, we prove, by induction, that

$$\|B_k - F'(x^*)\| \leq (2 - 2^{-k})\delta, \tag{5.4.45}$$

$$\|x_{k+1} - x^*\| \leq \frac{1}{2}\|x_k - x^*\|. \tag{5.4.46}$$

For $k = 0$, (5.4.45) is obvious. Since the proof of (5.4.46) for $k = 0$ is the same as that in the following general case, we omit it here.

Now, suppose that (5.4.45) and (5.4.46) hold for $k = 0, 1, \cdots, i-1$. For $k = i$, by assumption of induction and (5.4.40), we have

$$\begin{aligned} \|B_i - F'(x^*)\| &\leq \|B_{i-1} - F'(x^*)\| + \frac{\gamma}{2}(\|x_i - x^*\| + \|x_{i-1} - x^*\|) \\ &\leq (2 - 2^{-(i-1)})\delta + \frac{3}{4}\gamma\|x_{i-1} - x^*\|. \end{aligned} \tag{5.4.47}$$

From (5.4.46) and $\|x_0 - x^*\| < \varepsilon$, we have

$$\|x_{i-1} - x^*\| \leq 2^{-(i-1)}\|x_0 - x^*\| \leq 2^{-(i-1)}\varepsilon. \tag{5.4.48}$$

Substituting (5.4.48) into (5.4.47) and using (5.4.44) yield

$$\begin{aligned} \|B_i - F'(x^*)\| &\leq (2 - 2^{-(i-1)})\delta + \frac{3}{4}\gamma \cdot 2^{-(i-1)}\varepsilon \\ &\leq (2 - 2^{-(i-1)} + 2^{-i})\delta = (2 - 2^{-i})\delta, \end{aligned} \tag{5.4.49}$$

which proves (5.4.45).

To prove (5.4.46), we first show that B_i is invertible. In fact, since $\|F'(x^*)^{-1}\| \leq \beta$, it follows from (5.4.45) and (5.4.43) that

$$\begin{aligned} & \|F'(x^*)^{-1}[B_i - F'(x^*)]\| \\ \leq\ & \|F'(x^*)^{-1}\|\|B_i - F'(x^*)\| \\ \leq\ & \beta(2 - 2^{-i})\delta \leq 2\beta\delta \leq \frac{1}{3}. \end{aligned}$$

Then, by Von-Neumann Theorem 1.2.5, we know that B_i is invertible, and

$$\begin{aligned} \|B_i^{-1}\| &\leq \frac{\|F'(x^*)^{-1}\|}{1-\|F'(x^*)^{-1}(B_i-F'(x^*))\|} \\ &\leq \frac{\beta}{1-\frac{1}{3}} = \frac{3\beta}{2}. \end{aligned} \tag{5.4.50}$$

Thus, x_{i+1} is well-defined. Also,

$$\begin{aligned} B_i(x_{i+1}-x^*) &= B_i(x_i-x^*)-F(x_i)+F(x^*) \\ &= [-F(x_i)+F(x^*)+F'(x^*)(x_i-x^*)] \\ &\quad +[B_i-F'(x^*)](x_i-x^*), \end{aligned} \tag{5.4.51}$$

which gives

$$\begin{aligned} \|x_{i+1}-x^*\| \leq\ & \|B_i^{-1}\|[\|-F(x_i)+F(x^*)+F'(x^*)(x_i-x^*)\| \\ & +\|B_i-F'(x^*)\|\|x_i-x^*\|]. \end{aligned} \tag{5.4.52}$$

By use of Theorem 1.2.22,

$$\|-F(x_i)+F(x^*)+F'(x^*)(x_i-x^*)\| \leq \frac{\gamma}{2}\|x_i-x^*\|^2. \tag{5.4.53}$$

So, (5.4.52), (5.4.50), (5.4.53) and (5.4.49) give

$$\|x_{i+1}-x^*\| \leq \frac{3}{2}\beta\left[\frac{\gamma}{2}\|x_i-x^*\|+(2-2^{-i})\delta\right]\|x_i-x^*\|. \tag{5.4.54}$$

Also, using (5.4.48) and (5.4.44), we have

$$\frac{\gamma}{2}\|x_i-x^*\| \leq 2^{-(i+1)}\gamma\varepsilon \leq \frac{2^{-i}}{3}\delta.$$

Substituting the above inequality into (5.4.54), we obtain

$$\begin{aligned} \|x_{i+1}-x^*\| &\leq \frac{3}{2}\beta\left[\frac{1}{3}2^{-i}+2-2^{-i}\right]\delta\|x_i-x^*\| \\ &\leq 3\beta\delta\|x_i-x^*\| \\ &\leq \frac{1}{2}\|x_i-x^*\|. \end{aligned}$$

Therefore, the desired result (5.4.46) is proved.

Similarly, for the given condition (5.4.41), we also can prove our conclusion.

In fact, let $\|F'(x^*)\| \leq \beta$ and $r \in (0,1)$. Choose $\varepsilon(r) = \varepsilon$ and $\delta(r) = \delta$, such that

$$(2\alpha_1\delta + \alpha_2)\frac{\varepsilon}{1-r} \leq \delta, \tag{5.4.55}$$

$$\beta(1+r)(\gamma\varepsilon + 2\delta) \leq r. \tag{5.4.56}$$

To prove the local and linear convergence, we still prove

$$\|B_k - F'(x^*)\| \leq 2\delta, \tag{5.4.57}$$

$$\|x_{k+1} - x^*\| \leq r\|x_k - x^*\| \tag{5.4.58}$$

by induction.

Obviously, for $k = 0$, the conclusion holds. Suppose that the conclusion holds for $k = 0, 1, \cdots, i-1$. By (5.4.41), we have

$$\|B_{k+1} - F'(x^*)\| - \|B_k - F'(x^*)\| \leq 2\alpha_1\delta\varepsilon r^k + \alpha_2\varepsilon r^k.$$

Summing for $k = 0$ to $i-1$ yields

$$\|B_i - F'(x^*)\| \leq \|B_0 - F'(x^*)\| + (2\alpha_1\delta + \alpha_2)\frac{\varepsilon}{1-r}.$$

So, using (5.4.55) and $\|B_0 - F'(x^*)\| \leq \delta$, we obtain

$$\|B_i - F'(x^*)\| \leq 2\delta, \tag{5.4.59}$$

which proves (5.4.57).

To prove (5.4.58), first note that $\|B_i^{-1}\| \leq (1+r)\beta$ from (5.4.59) and Theorem 1.2.5. Then, by Theorem 1.2.24, we have

$$\begin{aligned}\|x_{i+1} - x^*\| &\leq \|B_i^{-1}\|[\|F(x_i) - F(x^*) - F'(x^*)(x_i - x^*)\| \\ &\quad + \|B_i - F'(x^*)\|\|x_i - x^*\|] \\ &\leq \beta(1+r)(\gamma\varepsilon + 2\delta)\|x_i - x^*\|.\end{aligned}$$

By using (5.4.56) we immediately obtain

$$\|x_{i+1} - x^*\| \leq r\|x_i - x^*\|.$$

So, (5.4.58) is proved. We complete the proof by induction. □

Similarly, we have the following local and linear convergence theorem for update of the inverse Hessian approximation.

Theorem 5.4.10 *Let $F : R^n \to R^n$ satisfy the conditions (a), (b), and (c) in Assumption 5.4.1. Let U be an update function, such that for all $(x_k, H_k) \in domU$ and $H_{k+1} \in U(x_k, H_k)$, we have that*

$$\|H_{k+1} - F'(x^*)^{-1}\| \leq \|H_k - F'(x^*)^{-1}\| + \frac{\gamma}{2}(\|x_{k+1} - x^*\| + \|x_k - x^*\|), \quad (5.4.60)$$

where γ is some constant, or that

$$\|H_{k+1} - F'(x^*)^{-1}\| \leq [1 + \alpha_1\sigma(x_k, x_{k+1})]\|H_k - F'(x^*)^{-1}\| + \alpha_2\sigma(x_k, x_{k+1}), \quad (5.4.61)$$

where α_1 and α_2 are some constants, and

$$\sigma(x_k, x_{k+1}) = \max\{\|x_k - x^*\|, \|x_{k+1} - x^*\|\}.$$

Then, there exist constants ε and δ, such that, for $\|x_0 - x^*\| < \varepsilon$ and $\|H_0 - F'(x^*)^{-1}\| < \delta$, the iteration

$$x_{k+1} = x_k - H_k F(x_k), \ H_{k+1} \in U(x_k, H_k) \quad (5.4.62)$$

is well-defined and $\{x_k\}$ converges to x^* linearly.

As a consequence of the above two theorems, we give the following corollaries on superlinear convergence for general iterations.

Corollary 5.4.11 *Suppose that the assumptions of Theorem 5.4.9 hold. If some subsequence of $\{\|B_k - F'(x^*)\|\}$ converges to zero, then $\{x_k\}$ converges to x^* superlinearly.*

Proof. We hope to prove

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

Let $r \in (0, 1)$. It follows from Theorem 5.4.9 that there exist $\varepsilon(r)$ and $\delta(r)$ such that $\|B_0 - F'(x^*)\| < \delta(r)$ and $\|x_0 - x^*\| < \varepsilon(r)$ imply that $\|x_{k+1} - x^*\| \leq r\|x_k - x^*\|, \forall k \geq 0$. From the assumption, we can choose $m > 0$ such that $\|B_m - F'(x^*)\| < \delta(r)$ and $\|x_m - x^*\| < \varepsilon(r)$. Hence $\|x_{k+1} - x^*\| \leq r\|x_k - x^*\|, \forall k \geq m$. Since $r \in (0, 1)$ is arbitrary, the conclusion is shown. □

Similarly, we have

Corollary 5.4.12 *Suppose that the conditions of Theorem 5.4.10 hold. If some subsequence of $\{\|H_k - F'(x^*)^{-1}\|\}$ converges to zero, then $\{x_k\}$ converges to x^* superlinearly.*

5.4.3 Local Convergence of Broyden's Rank-One Update

In this section, we prove the linear convergence and superlinear convergence of Broyden's rank-one update

$$x_{k+1} = x_k - B_k^{-1} F(x_k), \tag{5.4.63}$$

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k) s_k^T}{s_k^T s_k}. \tag{5.4.64}$$

Theorem 5.4.13 *Let $F : R^n \to R^n$ satisfy the conditions (a), (b), and (c) in Assumption 5.4.1. Assume that there exist positive constants ε and δ such that $\|x_0 - x^*\| < \varepsilon$ and $\|B_0 - F'(x^*)\| < \delta$. Then the sequence $\{x_k\}$ generated by Broyden's rank-one update (5.4.63)–(5.4.64) is well-defined and convergent to x^* superlinearly.*

Proof. It is enough to prove, under the conditions of the theorem, that (5.4.40) and (5.4.4) are satisfied respectively.

First, we prove that B_{k+1} generated by Broyden's rank-one update satisfies (5.4.40).

By (5.4.63)–(5.4.64), we have

$$\begin{aligned} B_{k+1} - F'(x^*) &= B_k - F'(x^*) + \frac{(y_k - B_k s_k) s_k^T}{s_k^T s_k} \\ &= B_k - F'(x^*) + \frac{(F'(x^*) s_k - B_k s_k) s_k^T}{s_k^T s_k} + \frac{(y_k - F'(x^*) s_k) s_k^T}{s_k^T s_k} \\ &= (B_k - F'(x^*)) \left[I - \frac{s_k s_k^T}{s_k^T s_k} \right] + \frac{(y_k - F'(x^*) s_k) s_k^T}{s_k^T s_k}. \end{aligned} \tag{5.4.65}$$

Taking norms gives

$$\|B_{k+1} - F'(x^*)\| \le \|B_k - F'(x^*)\| \left\| I - \frac{s_k s_k^T}{s_k^T s_k} \right\| + \frac{\|y_k - F'(x^*) s_k\|}{\|s_k\|}. \tag{5.4.66}$$

Note that

$$\left\| I - \frac{s_k s_k^T}{s_k^T s_k} \right\| = 1 \tag{5.4.67}$$

and

$$\begin{aligned} \|y_k - F'(x^*) s_k\| &= \|F(x_{k+1}) - F(x_k) - F'(x^*) s_k\| \\ &\le \frac{\gamma}{2} (\|x_{k+1} - x^*\| + \|x_k - x^*\|) \|s_k\| \end{aligned} \tag{5.4.68}$$

by Theorem 1.2.24, we obtain immediately that

$$\|B_{k+1} - F'(x^*)\| \leq \|B_k - F'(x^*)\| + \frac{\gamma}{2}(\|x_{k+1} - x^*\| + \|x_k - x^*\|),$$

which is (5.4.40). The linear convergence is proved.

Next, we prove the superlinear convergence of Broyden's rank-one update by use of Theorem 5.4.3, that is, we want to prove that (5.4.4) holds.

Let $E_k = B_k - F'(x^*)$. From (5.4.65),

$$\|E_{k+1}\|_F \leq \left\| E_k \left(I - \frac{s_k s_k^T}{s_k^T s_k} \right) \right\|_F + \frac{\|(y_k - F'(x^*)s_k)s_k^T\|_F}{s_k^T s_k}. \tag{5.4.69}$$

Since

$$\begin{aligned} \left\| E_k \frac{s_k s_k^T}{s_k^T s_k} \right\|_F^2 &= \operatorname{tr}\left(\frac{(E_k s_k)^T (E_k s_k)}{(s_k^T s_k)^2} s_k s_k^T \right) \\ &= \frac{\|E_k s_k\|^2}{\|s_k\|^4} \|s_k\|^2 = \frac{\|E_k s_k\|^2}{\|s_k\|^2}, \end{aligned}$$

we get

$$\begin{aligned} \|E_k\|_F^2 &= \left\| E_k \frac{s_k s_k^T}{s_k^T s_k} \right\|_F^2 + \left\| E_k \left(I - \frac{s_k s_k^T}{s_k^T s_k} \right) \right\|_F^2 \\ &= \frac{\|E_k s_k\|^2}{\|s_k\|^2} + \left\| E_k \left(I - \frac{s_k s_k^T}{s_k^T s_k} \right) \right\|_F^2. \end{aligned}$$

Hence

$$\left\| E_k \left(I - \frac{s_k s_k^T}{s_k^T s_k} \right) \right\|_F = \left(\|E_k\|_F^2 - \frac{\|E_k s_k\|^2}{\|s_k\|^2} \right)^{\frac{1}{2}}. \tag{5.4.70}$$

Since $(\alpha^2 - \beta^2)^{\frac{1}{2}} \leq \alpha - \beta^2/(2\alpha)$ for any $\alpha \geq |\beta| \geq 0$, (5.4.70) implies that

$$\left\| E_k \left(I - \frac{s_k s_k^T}{s_k^T s_k} \right) \right\|_F \leq \|E_k\|_F - \frac{1}{2\|E_k\|_F} \left(\frac{\|E_k s_k\|}{\|s_k\|} \right)^2. \tag{5.4.71}$$

Also, by means of Theorem 1.2.24,

$$\|y_k - F'(x^*)s_k\|_F \leq \frac{\gamma}{2}(\|x_{k+1} - x^*\| + \|x_k - x^*\|)\|s_k\|. \tag{5.4.72}$$

So, by using (5.4.71), (5.4.72), and (5.4.46), we can write (5.4.69) as

$$\|E_{k+1}\|_F \leq \|E_k\|_F - \frac{\|E_k s_k\|^2}{2\|E_k\|_F \|s_k\|^2} + \frac{3}{4}\gamma\|x_k - x^*\|,$$

which is

$$\frac{\|E_k s_k\|^2}{\|s_k\|^2} \leq 2\|E_k\|_F \left[\|E_k\|_F - \|E_{k+1}\|_F + \frac{3}{4}\gamma\|x_k - x^*\|\right]. \tag{5.4.73}$$

Recalling (5.4.45) and (5.4.46), we have that

$$\|E_k\|_F \leq 2\delta, \; \forall k \geq 0$$

and

$$\sum_{k=0}^{\infty} \|x_k - x^*\| \leq 2\varepsilon.$$

Thus, (5.4.73) can be written as

$$\frac{\|E_k s_k\|^2}{\|s_k\|^2} \leq 4\delta \left[\|E_k\|_F - \|E_{k+1}\|_F + \frac{3}{4}\gamma\|x_k - x^*\|\right]. \tag{5.4.74}$$

By summing both sides, we obtain

$$\begin{aligned} \sum_{k=0}^{i} \frac{\|E_k s_k\|^2}{\|s_k\|^2} &\leq 4\delta \left[\|E_0\|_F - \|E_{i+1}\|_F + \frac{3}{4}\gamma \sum_{k=0}^{i} \|x_k - x^*\|\right] \\ &\leq 4\delta \left[\|E_0\|_F + \frac{3}{2}\gamma\varepsilon\right] \\ &\leq 4\delta \left[\delta + \frac{3}{2}\gamma\varepsilon\right], \end{aligned} \tag{5.4.75}$$

which holds for any $i \geq 0$. Therefore

$$\sum_{k=0}^{\infty} \frac{\|E_k s_k\|^2}{\|s_k\|^2}$$

is finite and further

$$\lim_{k\to\infty} \frac{\|E_k s_k\|}{\|s_k\|} = 0, \tag{5.4.76}$$

which is (5.4.4). Then we have proved the superlinear convergence of Broyden's rank-one update by use of Theorem 5.4.3. □

Similarly, for the following form of Broyden's rank-one update in inverse Hessian approximation:

$$x_{k+1} = x_k - H_k F(x_k), \tag{5.4.77}$$

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k) y_k^T}{y_k^T y_k}, \tag{5.4.78}$$

we have the following theorem.

Theorem 5.4.14 *Let $F : R^n \to R^n$ satisfy the conditions (a), (b), and (c) in Assumption 5.4.1. Assume that there exist ε and δ such that $\|x_0 - x^*\| < \varepsilon$ and $\|H_0 - F'(x^*)^{-1}\| < \delta$. Then the sequence $\{x_k\}$ generated by Broyden's rank-one update (5.4.77)–(5.4.78) is well-defined and convergent to x^* superlinearly.*

5.4.4 Local and Linear Convergence of DFP Method

In this subsection and subsequent subsections, we discuss the local convergence of rank-two methods, which includes the linear and superlinear convergence, and local convergence under line search. Note that we introduce two different techniques to prove the superlinear convergence of BFGS and DFP methods respectively.

The DFP iteration we consider is

$$x_{k+1} = x_k - B_k^{-1} \nabla f(x_k), \tag{5.4.79}$$

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k) y_k^T + y_k (y_k - B_k s_k)^T}{y_k^T s_k} - \frac{(y_k - B_k s_k)^T s_k}{(y_k^T s_k)^2} y_k y_k^T. \tag{5.4.80}$$

To study the local convergence of DFP method, it is required to estimate $\|B_{k+1} - \nabla^2 f(x^*)\|$. As shown in the following theorem, there is a matrix $P = I - \frac{s_k y_k^T}{s_k^T y_k}$ in $B_{k+1} - \nabla^2 f(x^*)$. Since

$$\|P\|_2 = \frac{\|s_k\| \|y_k\|}{s_k^T y_k}, \tag{5.4.81}$$

it is a secant of the angle between y_k and s_k. In general, y_k and s_k is not parallel, so $\|P\|_2$ may be quite big, and it is not suitable to estimate

$\|B_{k+1} - \nabla^2 f(x^*)\|$ by means of l_2 norm. However, near x^*, $f(x)$ closes a quadratic function, and hence $A^{-\frac{1}{2}} y_k$ and $A^{\frac{1}{2}} s_k$ are approximately parallel, where $A = \nabla^2 f(x^*)$. It motivates us to use some weighted norm to estimate $\|B_{k+1} - \nabla^2 f(x^*)\|$. Then we define

$$\|E\|_{DFP} = \|E\|_{A^{-\frac{1}{2}},F} = \|A^{-\frac{1}{2}} E A^{-\frac{1}{2}}\|_F. \tag{5.4.82}$$

Below, we first develop the linear convergence of DFP method.

Theorem 5.4.15 *Let $f : R^n \to R$ satisfy Assumption 5.4.2. Also let*

$$\mu\gamma\sigma(x_k, x_{k+1}) \leq \frac{1}{3} \tag{5.4.83}$$

in a neighborhood of x^*, where $\mu = \|\nabla^2 f(x^*)^{-1}\|, \sigma(x_k, x_{k+1}) = \max\{\|x_k - x^*\|, \|x_{k+1} - x^*\|\}$. Then, there exist $\varepsilon > 0$ and $\delta > 0$ such that for $\|x_0 - x^*\| < \varepsilon$ and $\|B_0 - \nabla^2 f(x^*)\|_{DFP} < \delta$, the iteration (5.4.79)–(5.4.80) of DFP method is well-defined, and the produced sequence $\{x_k\}$ converges to x^* linearly.

Proof. Based on Theorem 5.4.9, to prove the linear convergence of DFP method, it is enough to prove

$$\begin{aligned} \|B_{k+1} - \nabla^2 f(x^*)\|_{DFP} \quad < \quad & [1 + \alpha_1 \sigma(x_k, x_{k+1})] \|B_k - \nabla^2 f(x^*)\|_{DFP} \\ & + \alpha_2 \sigma(x_k, x_{k+1}), \end{aligned} \tag{5.4.84}$$

where α_1 and α_2 are positive constants independent of x_k and x_{k+1}, $\sigma(x_k, x_{k+1}) = \max\{\|x_k - x^*\|, \|x_{k+1} - x^*\|\}$.

Let $A = \nabla^2 f(x^*)$. From (5.4.79)–(5.4.80), it follows that

$$B_{k+1} - A = P^T (B_k - A) P + \frac{(y_k - A s_k) y_k^T + y_k (y_k - A s_k)^T P}{y_k^T s_k}, \tag{5.4.85}$$

where

$$P = I - \frac{s_k y_k^T}{s_k^T y_k}. \tag{5.4.86}$$

Note that $\|P\|_2 = \|s_k\| \|y_k\| / s_k^T y_k$, hence

$$\begin{aligned} \|P^T (B_k - A) P\|_{DFP} \quad & \leq \quad \|A^{\frac{1}{2}} P A^{-\frac{1}{2}}\|_2^2 \|B_k - A\|_{DFP} \\ & \leq \quad \frac{1}{\omega^2} \|B_k - A\|_{DFP}, \end{aligned} \tag{5.4.87}$$

$$\left\| \frac{y_k(y_k - As_k)^T P}{y_k^T s_k} \right\|_{DFP} \leq \frac{1}{\omega^2} \frac{\|A^{-\frac{1}{2}} y_k - A^{\frac{1}{2}} s_k\|}{\|A^{\frac{1}{2}} s_k\|}, \tag{5.4.88}$$

$$\left\| \frac{(y_k - As_k) y_k^T}{y_k^T s_k} \right\|_{DFP} \leq \frac{1}{\omega} \frac{\|A^{-\frac{1}{2}} y_k - A^{\frac{1}{2}} s_k\|}{\|A^{\frac{1}{2}} s_k\|}, \tag{5.4.89}$$

where

$$\omega = \frac{y_k^T s_k}{\|A^{-\frac{1}{2}} y_k\| \|A^{\frac{1}{2}} s_k\|} = \frac{\langle A^{-\frac{1}{2}} y_k, A^{\frac{1}{2}} s_k \rangle}{\|A^{-\frac{1}{2}} y_k\| \|A^{\frac{1}{2}} s_k\|}. \tag{5.4.90}$$

Now, we estimate $\|B_{k+1} - A\|_{DFP}$ by using (5.4.87), (5.4.88) and (5.4.89), and have

$$\begin{aligned} \|B_{k+1} - A\|_{DFP} &\leq \frac{1}{\omega^2} \|B_k - A\|_{DFP} \\ &\quad + \frac{2}{\omega^2} \frac{\|A^{-\frac{1}{2}} y_k - A^{\frac{1}{2}} s_k\|}{\|A^{\frac{1}{2}} s_k\|}. \end{aligned} \tag{5.4.91}$$

Note from Theorem 1.2.24 that

$$\begin{aligned} \frac{\|A^{-\frac{1}{2}} y_k - A^{-\frac{1}{2}} s_k\|}{\|A^{\frac{1}{2}} s_k\|} &\leq \frac{\|A^{-\frac{1}{2}}\| \|y_k - As_k\|}{\|s_k\| / \|A^{-\frac{1}{2}}\|} \\ &= \mu \frac{\|y_k - As_k\|}{\|s_k\|} \\ &\leq \mu\gamma\sigma(x_k, x_{k+1}) \leq \frac{1}{3}. \end{aligned} \tag{5.4.92}$$

Also, by Lemma 5.4.5, we have

$$1 - \omega^2 \leq \left[\mu \frac{\|y_k - As_k\|}{\|s_k\|} \right]^2 \leq [\mu\gamma\sigma(x_k, x_{k+1})]^2.$$

Then, if x_k and x_{k+1} are in the neighborhood of x^*, then

$$1 - \omega^2 \leq [\mu\gamma\sigma(x_k, x_{k+1})]^2 < \frac{1}{2},$$

which is

$$\omega^2 > \frac{1}{2} > \mu\gamma\sigma(x_k, x_{k+1}).$$

Hence

$$\begin{aligned}\frac{1}{\omega^2} &= 1+\frac{1-\omega^2}{\omega^2} < 1+\frac{[\mu\gamma\sigma(x_k,x_{k+1})]^2}{\mu\gamma\sigma(x_k,x_{k+1})} \\ &= 1+\mu\gamma\sigma(x_k,x_{k+1}).\end{aligned}$$

So, the two terms in (5.4.91) satisfy respectively

$$\frac{1}{\omega^2}\|B_k-A\|_{DFP} < (1+\mu\gamma\sigma(x_k,x_{k+1}))\|B_k-A\|_{DFP} \tag{5.4.93}$$

and

$$\begin{aligned}\frac{2}{\omega^2}\frac{\|A^{-\frac{1}{2}}y_k-A^{\frac{1}{2}}s_k\|}{\|A^{\frac{1}{2}}s_k\|} &< 2[1+\mu\gamma\sigma(x_k,x_{k+1})]\mu\gamma\sigma(x_k,x_{k+1})]\mu\gamma\sigma(x_k,x_{k+1}) \\ &< 3\mu\gamma\sigma(x_k,x_{k+1}).\end{aligned} \tag{5.4.94}$$

Substituting (5.4.93) and (5.4.94) into (5.4.112) yields (5.4.113), where $\alpha_1 = \mu\gamma$, $\alpha_2 = 3\mu\gamma$. So, we complete the proof. □

5.4.5 Superlinear Convergence of BFGS Method

In this subsection, we discuss the superlinear convergence of BFGS method. Let

$$\tilde{s}_k = G_*^{\frac{1}{2}}s_k,\ \tilde{y}_k = G_*^{-\frac{1}{2}}y_k,\ \tilde{B}_k = G_*^{-\frac{1}{2}}B_kG_*^{-\frac{1}{2}}, \tag{5.4.95}$$

where $G_* = G(x^*) = \nabla^2 f(x^*)$. Define

$$\cos\tilde{\theta}_k = \frac{\tilde{s}_k^T\tilde{B}_k\tilde{s}_k}{\|\tilde{s}_k\|\|\tilde{B}_k\tilde{s}_k\|},\ \ \tilde{q}_k = \frac{\tilde{s}_k^T\tilde{B}_k\tilde{s}_k}{\|\tilde{s}_k\|^2}, \tag{5.4.96}$$

and define

$$\tilde{M}_k = \frac{\|\tilde{y}_k\|^2}{\tilde{y}_k^T\tilde{s}_k},\ \ \tilde{m}_k = \frac{\tilde{y}_k^T\tilde{s}_k}{\tilde{s}_k^T\tilde{s}_k}. \tag{5.4.97}$$

By pre- and postmultiplying the BFGS update (5.1.45) by $G_*^{-\frac{1}{2}}$, we obtain

$$\tilde{B}_{k+1} = \tilde{B}_k - \frac{\tilde{B}_k\tilde{s}_k\tilde{s}_k^T\tilde{B}_k}{\tilde{s}_k^T\tilde{B}_k\tilde{s}_k} + \frac{\tilde{y}_k\tilde{y}_k^T}{\tilde{y}_k^T\tilde{s}_k}. \tag{5.4.98}$$

Since this expression has precisely the same form as the BFGS formula, it follows from the argument leading to (5.3.42) that

$$\begin{aligned}\psi(\tilde{B}_{k+1}) &= \psi(\tilde{B}_k) + (\tilde{M}_k - \ln \tilde{m}_k - 1) \\ &= \left[1 - \frac{\tilde{q}_k}{\cos^2 \tilde{\theta}_k} + \ln \frac{\tilde{q}_k}{\cos^2 \tilde{\theta}_k}\right] + \ln \cos^2 \tilde{\theta}_k. \end{aligned} \tag{5.4.99}$$

Noting that

$$y_k - G_* s_k = (\bar{G}_k - G_*) s_k,$$

where

$$\bar{G}_k = \int_0^1 \nabla^2 f(x_k + \tau s_k) \mathrm{d}\tau,$$

we obtain

$$\tilde{y}_k - \tilde{s}_k = G_*^{-\frac{1}{2}} (\bar{G}_* - G_k) G_*^{-\frac{1}{2}} \tilde{s}_k.$$

Assuming that the Hessian matrix G is Lipschitz at x^*, then we have

$$\|\tilde{y}_k - \tilde{s}_k\| \leq \|G_*^{-\frac{1}{2}}\|^2 \|\tilde{s}_k\| \|\bar{G}_k - G_*\| \leq \|G_*^{-\frac{1}{2}}\|^2 \|\tilde{s}_k\| L \epsilon_k,$$

which gives

$$\frac{\|\tilde{y}_k - \tilde{s}_k\|}{\|\tilde{s}_k\|} \leq \bar{c} \epsilon_k \tag{5.4.100}$$

for some positive constant $\bar{c}$, where

$$\epsilon_k = \max\{\|x_{k+1} - x^*\|, \|x_k - x^*\|\}. \tag{5.4.101}$$

Now we are in a position to prove the superlinear convergence theorem.

Theorem 5.4.16 *Let f be twice continuously differentiable and the Hessian matrix G be Lipschitz continuous at x^*. Suppose that the sequence generated by the BFGS algorithm converges to a minimizer x^* and that the condition*

$$\sum_{k=1}^{\infty} \|x_k - x^*\| < \infty \tag{5.4.102}$$

holds. Then $\{x_k\}$ converges to x^* at a superlinear rate.

Proof. By (5.4.100), we have

$$\|\tilde{y}_k\| - \|\tilde{s}_k\| \le \bar{c}\epsilon_k\|\tilde{s}_k\|, \ \|\tilde{s}_k\| - \|\tilde{y}_k\| \le \bar{c}\epsilon_k\|\tilde{s}_k\|,$$

which give

$$(1 - \bar{c}\epsilon_k)\|\tilde{s}_k\| \le \|\tilde{y}_k\| \le (1 + \bar{c}\epsilon_k)\|\tilde{s}_k\|. \tag{5.4.103}$$

By squaring (5.4.100) and using (5.4.103), we obtain

$$(1 - \bar{c}\epsilon_k)^2\|\tilde{s}_k\|^2 - 2\tilde{y}_k^T\tilde{s}_k + \|\tilde{s}_k\|^2 \le \|\tilde{y}_k\|^2 - 2\tilde{y}_k^T\tilde{s}_k + \|\tilde{s}_k\|^2 \le \bar{c}^2\epsilon_k^2\|\tilde{s}_k\|^2,$$

and therefore

$$2\tilde{y}_k^T\tilde{s}_k \ge (1 - 2\bar{c}\epsilon_k + \bar{c}^2\epsilon_k^2 + 1 - \bar{c}^2\epsilon_k^2)\|\tilde{s}_k\|^2 = 2(1 - \bar{c}\epsilon_k)\|\tilde{s}_k\|^2.$$

It follows from the definition of $\tilde{m}_k$ that

$$\tilde{m}_k = \frac{\tilde{y}_k^T\tilde{s}_k}{\|\tilde{s}_k\|^2} \ge 1 - \bar{c}\epsilon_k. \tag{5.4.104}$$

Combining (5.4.103) and (5.4.104) gives also that

$$\tilde{M}_k = \frac{\|\tilde{y}_k\|^2}{\tilde{y}_k^T\tilde{s}_k} \le \frac{1 + \bar{c}\epsilon_k}{1 - \bar{c}\epsilon_k}. \tag{5.4.105}$$

Since $x_k \to x^*$, we have that $\epsilon_k \to 0$. Thus by (5.4.105) there exists a positive constant $c > \bar{c}$ such that the following inequalities hold for all sufficient large k:

$$\tilde{M}_k \le 1 + \frac{2\bar{c}}{1 - \bar{c}\epsilon_k}\epsilon_k \le 1 + c\epsilon_k. \tag{5.4.106}$$

Making use of the nonpositiveness of the function $h(t) = 1 - t + \ln t$ gives

$$\frac{-x}{1-x} - \ln(1-x) = h\left(\frac{1}{1-x}\right) \le 0.$$

Now for k large enough we can assume that $\bar{c}\epsilon_k < \frac{1}{2}$, and by using the above inequality we have

$$\ln(1 - \bar{c}\epsilon_k) \ge \frac{-\bar{c}\epsilon_k}{1 - \bar{c}\epsilon_k} \ge -2\bar{c}\epsilon_k.$$

This relation and (5.4.104) imply that for sufficiently large k, we have

$$\ln \tilde{m}_k \ge \ln(1 - \bar{c}\epsilon_k) \ge -2\bar{c}\epsilon_k > -2c\epsilon_k. \tag{5.4.107}$$

We can now deduce from (5.4.99), (5.4.106), and (5.4.107) that

$$0 < \psi(\tilde{B}_{k+1}) \leq \psi(\tilde{B}_k) + 3c\epsilon_k + \ln\cos^2\tilde{\theta}_k + \left[1 - \frac{\tilde{q}_k}{\cos^2\tilde{\theta}_k} + \ln\frac{\tilde{q}_k}{\cos^2\tilde{\theta}_k}\right]. \tag{5.4.108}$$

By summing this expression and making use of (5.4.102) we have that

$$\begin{aligned} & \sum_{j=0}^{\infty}\left(\ln\frac{1}{\cos^2\tilde{\theta}_j} - \left[1 - \frac{\tilde{q}_j}{\cos^2\tilde{\theta}_j} + \ln\frac{\tilde{q}_j}{\cos^2\tilde{\theta}_j}\right]\right) \\ \leq \quad & \psi(\tilde{B}_0) + 3c\sum_{j=0}^{\infty}\epsilon_j < +\infty. \end{aligned}$$

Since the term in the square brackets is nonpositive, and since $\ln(1/\cos^2\tilde{\theta}_j) \geq 0$ for all j, we obtain

$$\lim_{j\to\infty}\ln\frac{1}{\cos^2\tilde{\theta}_j} = 0, \ \lim_{j\to\infty}\left(1 - \frac{\tilde{q}_j}{\cos^2\tilde{\theta}_j} + \ln\frac{\tilde{q}_j}{\cos^2\tilde{\theta}_j}\right) = 0,$$

which imply that

$$\lim_{j\to\infty}\cos\tilde{\theta}_j = 1, \ \lim_{j\to\infty}\tilde{q}_j = 1. \tag{5.4.109}$$

By use of these limits we can obtain that

$$\begin{aligned} \frac{\|G_*^{-\frac{1}{2}}(B_k - G_*)s_k\|^2}{\|G_*^{\frac{1}{2}}s_k\|^2} &= \frac{\|(\tilde{B}_k - I)\tilde{s}_k\|^2}{\|\tilde{s}_k\|^2} \\ &= \frac{\|\tilde{B}_k\tilde{s}_k\|^2 - 2\tilde{s}_k^T\tilde{B}_k\tilde{s}_k + \tilde{s}_k^T\tilde{s}_k}{\tilde{s}_k^T\tilde{s}_k} \\ &= \frac{\tilde{q}_k^2}{\cos^2\tilde{\theta}_k} - 2\tilde{q}_k + 1 \\ &\to 0. \end{aligned} \tag{5.4.110}$$

Then we conclude that

$$\lim_{k\to\infty}\frac{\|(B_k - G_*)s_k\|}{\|s_k\|} = 0 \tag{5.4.111}$$

which shows that the rate of convergence is superlinear. □

5.4.6 Superlinear Convergence of DFP Method

We first give the following three lemmas.

Lemma 5.4.17 *Let $M \in R^{n\times n}$ be a nonsingular symmetric matrix. If, for $\beta \in [0, 1/3]$, the inequality*

$$\|My_k - M^{-1}s_k\| \le \beta\|M^{-1}s_k\| \tag{5.4.112}$$

holds, then for any nonzero matrix $E \in R^{n\times n}$, we have
(a)

$$(1-\beta)\|M^{-1}s_k\|^2 \le y_k^T s_k \le (1+\beta)\|M^{-1}s_k\|^2, \tag{5.4.113}$$

(b)

$$\left\|E\left[I - \frac{(M^{-1}s_k)(M^{-1}s_k)^T}{y_k^T s_k}\right]\right\|_F \le \sqrt{1-\alpha\theta^2}\|E\|_F, \tag{5.4.114}$$

(c)

$$\begin{aligned}&\left\|E\left[I - \frac{M^{-1}s_k(My_k)^T}{y_k^T s_k}\right]\right\|_F \\ \le\ &\left[\sqrt{1-\alpha\theta^2} + (1-\beta)^{-1}\frac{\|My_k - M^{-1}s_k\|}{\|M^{-1}s_k\|}\right]\|E\|_F,\end{aligned} \tag{5.4.115}$$

where

$$\alpha = \frac{1-2\beta}{1-\beta^2} \in \left[\frac{3}{8}, 1\right),\ \theta = \frac{\|EM^{-1}s_k\|}{\|E\|_F\|M^{-1}s_k\|} \in [0,1]. \tag{5.4.116}$$

Proof. Note that

$$y_k^T s_k = (My_k)^T(M^{-1}s_k) = (My_k - M^{-1}s_k)^T M^{-1}s_k + \|M^{-1}s_k\|^2. \tag{5.4.117}$$

Also, it follows from Cauchy-Schwartz inequality and (5.4.112) that

$$|(My_k - M^{-1}s_k)^T M^{-1}s_k| \le \beta\|M^{-1}s_k\|^2. \tag{5.4.118}$$

Then, combining (5.4.117) and (5.4.118) gives the first conclusion (a).

Now, we prove (b). By using the property (1.2.71) of Frobenius norm of a rank-one update, we have

$$\|E(I - uv^T)\|_F^2 = \|E\|_F^2 - 2v^T E^T Eu + \|Eu\|^2\|v\|^2.$$

In particular,

$$\begin{aligned} & \left\| E\left[I - \frac{(M^{-1}s_k)(M^{-1}s_k)^T}{y_k^T s_k}\right]\right\|_F^2 \\ = \quad & \|E\|_F^2 + (-2y_k^T s_k + \|M^{-1}s_k\|^2)\frac{\|EM^{-1}s_k\|}{(y_k^T s_k)^2}. \end{aligned}$$

Using (a) and (5.4.116) yields

$$\begin{aligned} & \left\| E\left[I - \frac{(M^{-1}s_k)(M^{-1}s_k)^T}{y_k^T s_k}\right]\right\|_F^2 \\ \leq \quad & \|E\|_F^2 - \left(\frac{1-2\beta}{1-\beta}\right)\frac{\|EM^{-1}s_k\|^2}{y_k^T s_k} \\ \leq \quad & \|E\|_F^2 - \alpha\left(\frac{\|EM^{-1}s_k\|}{\|M^{-1}s_k\|}\right)^2 \\ = \quad & \|E\|_F^2(1-\alpha\theta^2), \end{aligned}$$

which shows (b).

Finally, we prove (c) by means of (b). It is enough to prove that

$$\begin{aligned} & \left\| E\frac{M^{-1}s_k(M^{-1}s_k - My_k)^T}{y_k^T s_k}\right\|_F \\ \leq \quad & (1-\beta)^{-1}\left(\frac{\|My_k - M^{-1}s_k\|}{\|M^{-1}s_k\|}\right)\|E\|_F. \end{aligned} \tag{5.4.119}$$

Since

$$\left\|\frac{M^{-1}s_k(M^{-1}s_k - My_k)^T}{y_k^T s_k}\right\|_F \leq \frac{\|M^{-1}s_k\|\|M^{-1}s_k - My_k\|}{y_k^T s_k},$$

then we obtain (5.4.119) by using (a). □

Lemma 5.4.18 *Let $\{\phi_k\}$ and $\{\delta_k\}$ be sequences of nonnegative numbers satisfying*

$$\phi_{k+1} \leq (1+\delta_k)\phi_k + \delta_k \tag{5.4.120}$$

and

$$\sum_{k=1}^{\infty}\delta_k < +\infty, \tag{5.4.121}$$

then $\{\phi_k\}$ converges.

Proof. We first prove that $\{\phi_k\}$ is bounded above. Let

$$\mu_k = \prod_{j=1}^{k-1}(1+\delta_j).$$

Obviously, $\mu_k \geq 1$. Inequality (5.4.121) indicates that there exists a constant μ such that $\mu_k \leq \mu$. By using (5.4.120), we have

$$\frac{\phi_{k+1}}{\mu_{k+1}} \leq \frac{\phi_k}{\mu_k} + \frac{\delta_k}{\mu_{k+1}} \leq \frac{\phi_k}{\mu_k} + \delta_k.$$

Hence

$$\frac{\phi_{m+1}}{\mu_{m+1}} \leq \frac{\phi_1}{\mu_1} + \sum_{k=1}^{m}\delta_k.$$

From (5.4.121) and the boundedness of $\{\mu_k\}$, we obtain that $\{\phi_k\}$ is bounded.

Since $\{\phi_k\}$ is bounded, then there is at least a limit point. Suppose that there are two subsequences $\{\phi_{k_n}\}$ and $\{\phi_{k_m}\}$, which converge to ϕ' and ϕ'' respectively. We can show that $\phi' \leq \phi''$, and that $\phi'' \leq \phi'$ by symmetry. Thus $\phi' = \phi''$ and $\{\phi_k\}$ is convergent.

In fact, let ϕ be a bound of $\{\phi_k\}$. Let also, for example, $k_n \geq k_m$. From (5.4.120), we have

$$\phi_{k_n} - \phi_{k_m} \leq (1+\phi)\sum_{j=k_m}^{k_n-1}\delta_j.$$

By the selection of k_n, we have

$$\phi' - \phi_{k_m} \leq (1+\phi)\sum_{j=k_m}^{\infty}\delta_j.$$

By the selection of k_m, we have

$$\phi' - \phi'' \leq 0.$$

Therefore $\phi' \leq \phi''$. Similarly, by symmetry, we obtain $\phi'' \leq \phi'$. We complete the proof. □

We have known that if $f : R^n \to R$ satisfies Assumption 5.4.2, then (5.4.84) holds. Let $\|B_k - A\|_{DFP} = \phi_k$ and $\max\{\alpha_1\sigma(x_k, x_{k+1}),\ \alpha_2\sigma(x_k, x_{k+1})\} = \delta_k$. Then (5.4.121) holds. Thus, it follows from Lemma 5.4.18 that the limit

$$\lim_{k\to+\infty}\|B_k - A\|_{DFP} \tag{5.4.122}$$

exists.

Lemma 5.4.19 *Under the assumptions of Theorem 5.4.15, there exist positive constants β_1, β_2, and β_3, such that $\forall x_k, x_{k+1} \in N(x^*, \varepsilon)$, we have*

$$\begin{aligned}\|B_{k+1} - \nabla^2 f(x^*)\|_{DFP} &\leq \left[\sqrt{1-\beta_1\theta_k^2} + \beta_2\sigma(x_k, x_{k+1})\right]\|B_k - \nabla^2 f(x^*)\|_{DFP} \\ &\quad +\beta_3\sigma(x_k, x_{k+1}),\end{aligned} \tag{5.4.123}$$

where

$$\sigma(x_k, x_{k+1}) = \max\{\|x_k - x^*\|, \|x_{k+1} - x^*\|\}, \tag{5.4.124}$$

$$\theta_k = \frac{\|\nabla^2 f(x^*)^{-\frac{1}{2}}[B_k - \nabla^2 f(x^*)^{\frac{1}{2}}]s_k\|}{\|B_k - \nabla^2 f(x^*)\|_{DFP}\|\nabla^2 f(x^*)^{\frac{1}{2}}s_k\|}. \tag{5.4.125}$$

Proof. Write $A = \nabla^2 f(x^*)$. From (5.4.85), we have

$$\begin{aligned}\|B_{k+1} - A\|_{DFP} &\leq \|P^T(B_k - A)P\|_{DFP} + \left\|\frac{(y_k - As_k)y_k^T}{y_k^T s_k}\right\|_{DFP} \\ &\quad + \left\|\frac{y_k(y_k - As_k)^T P}{y_k^T s_k}\right\|_{DFP}.\end{aligned} \tag{5.4.126}$$

Let

$$Q = I - \frac{A^{\frac{1}{2}}s_k y_k^T A^{-\frac{1}{2}}}{y_k^T s_k},\ E_k = A^{-\frac{1}{2}}(B_k - A)A^{-\frac{1}{2}}. \tag{5.4.127}$$

Then

$$\begin{aligned}\|P^T(B_k - A)P\|_{DFP} &= \|(A^{-\frac{1}{2}}P^T A^{\frac{1}{2}})(A^{-\frac{1}{2}}(B_k - A)A^{-\frac{1}{2}})(A^{\frac{1}{2}}PA^{-\frac{1}{2}})\|_F \\ &= \|Q^T E Q\|_F.\end{aligned}$$

Similar to the proof of Theorem 5.4.15, we know that there exist α_3 and $\alpha_4 > 0$ such that

$$\left\|\frac{(y_k - As_k)y_k^T}{y_k^T s_k}\right\|_{DFP} \leq \frac{1}{\omega}\frac{\|A^{-\frac{1}{2}}y_k - A^{\frac{1}{2}}s_k\|}{\|A^{\frac{1}{2}}s_k\|} \leq \alpha_3\sigma(x_k, x_{k+1}),$$

$$\left\|\frac{y_k(y_k - As_k)^T P}{y_k^T s_k}\right\|_{DFP} \leq \frac{1}{\omega^2}\frac{\|A^{-\frac{1}{2}}y_k - A^{\frac{1}{2}}s_k\|}{\|A^{\frac{1}{2}}s_k\|} \leq \alpha_4\sigma(x_k, x_{k+1}).$$

If we let $\beta_3 = \alpha_3 + \alpha_4$, then (5.4.126) becomes

$$\|B_{k+1} - A\|_{DFP} \leq \|Q^T E Q\|_F + \beta_3\sigma(x_k, x_{k+1}). \tag{5.4.128}$$

Since

$$\frac{\|A^{-\frac{1}{2}}y_k - A^{\frac{1}{2}}s_k\|}{\|A^{\frac{1}{2}}s_k\|} \leq \mu\gamma\sigma(x_k, x_{k+1}) \leq \frac{1}{3},$$

then, by use of Lemma 5.4.17, we obtain

$$\|Q^T EQ\|_F \leq \left[1 + (1-\beta)^{-1}\frac{\|A^{-\frac{1}{2}}y_k - A^{\frac{1}{2}}s_k\|}{\|A^{\frac{1}{2}}s_k\|}\right]\|Q^T E\|_F.$$

Note that $\|Q^T E\|_F = \|E^T Q\|_F = \|EQ\|_F$, thus, by using Lemma 5.4.17 once more, we obtain

$$\|EQ\|_F \leq \left[\sqrt{1-\alpha\theta_k^2} + (1-\beta)^{-1}\frac{\|A^{-\frac{1}{2}}y_k - A^{\frac{1}{2}}s_k\|}{\|A^{\frac{1}{2}}s_k\|}\right]\|E\|_F,$$

where θ_k is defined by (5.4.125). Then

$$\begin{aligned}\|Q^T EQ\|_F &\leq \left[\sqrt{1-\alpha\theta_k^2} + \frac{5}{2}(1-\beta)^{-1}\frac{\|A^{-\frac{1}{2}}y_k - A^{\frac{1}{2}}s_k\|}{\|A^{\frac{1}{2}}s_k\|}\right]\|E\|_F \\ &\leq [\sqrt{1-\beta_1\theta_k^2} + \beta_2\sigma(x_k, x_{k+1})]\|E\|_F, \qquad (5.4.129)\end{aligned}$$

where $\beta_1 = \alpha$, $\beta_2 = \frac{5}{2}(1-\beta)^{-1}\mu\gamma$. Substituting (5.4.129) into (5.4.128), we deduce the desired result (5.4.123). The proof is complete. □

Using the above three lemmas, we can establish the following superlinear convergence theorem of DFP method.

Theorem 5.4.20 *Under the assumptions of Theorem 5.4.15, DFP method defined by (5.4.79)–(5.4.80) is convergent superlinearly.*

Proof. Since $(1-\beta_1\theta_k^2)^{\frac{1}{2}} \leq 1 - (\beta_1/2)\theta_k^2$, then (5.4.123) can be written as

$$\begin{aligned}(\beta_1\theta_k^2/2)\|B_k - A\|_{DFP} &\leq \|B_k - A\|_{DFP} - \|B_{k+1} - A\|_{DFP} \\ &\quad + [\beta_2\|B_k - A\|_{DFP} + \beta_3]\sigma(x_k, x_{k+1}).\end{aligned}$$

Summing both sides yields

$$\begin{aligned}\frac{1}{2}\beta_1\sum_{k=1}^{\infty}\theta_k^2\|B_k - A\|_{DFP} &\leq \|B_1 - A\|_{DFP} + \beta_2\sum_{k=1}^{\infty}\sigma(x_k, x_{k+1})\|B_k - A\|_{DFP} \\ &\quad + \beta_3\sum_{k=1}^{\infty}\sigma(x_k, x_{k+1}).\end{aligned}$$

Since, from Theorem 5.4.15, $\{x_k\}$ is linearly convergent, then $\sum_{k=1}^{\infty} \sigma(x_k, x_{k+1}) < \infty$. Also, since $\{\|B_k - A\|_{DFP}\}$ is bounded, then

$$\frac{\beta_1}{2} \sum_{k=1}^{\infty} \theta_k^2 \|B_k - A\|_{DFP} < \infty.$$

By (5.4.56), the limit $\lim_{k\to\infty} \|B_k - A\|_{DFP}$ exists. Hence, if some subsequence of $\{\|B_k - A\|_{DFP}\}$ converges to zero, the whole sequence converges to zero. Therefore

$$\lim_{k\to\infty} \frac{\|(B_k - A)s_k\|}{\|s_k\|} = 0,$$

and the conclusion holds. Otherwise, if $\|B_k - A\|_{DFP} \geq \omega > 0, \forall k \geq k_0$, then $\theta_k \to 0$. Note that

$$\begin{aligned} \frac{\|(B_k - A)s_k\|}{\|s_k\|} &\leq \frac{\|A^{\frac{1}{2}}\| \|A^{-\frac{1}{2}}(B_k - A)s_k\|}{\|A^{\frac{1}{2}}\|^{-1} \|A^{\frac{1}{2}} s_k\|} \\ &= \|A\| \|B_k - A\|_{DFP} \frac{\|A^{-\frac{1}{2}}(B_k - A)s_k\|}{\|B_k - A\|_{DFP} \|A^{\frac{1}{2}} s_k\|} \\ &= \|A\| \|B_k - A\|_{DFP} \theta_k, \end{aligned}$$

then, by using $\theta_k \to 0$, we immediately obtain

$$\lim_{k\to\infty} \frac{\|(B_k - A)s_k\|}{\|s_k\|} = 0.$$

Hence $\{x_k\}$ is convergent superlinearly. We complete the proof. □

Similarly, we can state the superlinear convergence theorem for BFGS method.

Theorem 5.4.21 *Under the assumptions of Theorem 5.4.15, the sequence $\{x_k\}$ generated by BFGS method (5.4.79) and (5.1.45) is convergent to x^* superlinearly.*

It is not difficult to describe the above theorems in inverse Hessian approximations, which proofs are left to interested readers as an exercise.

We consider BFGS update in inverse Hessian approximation (5.1.48), now written again as

$$x_{k+1} = x_k - H_k g_k, \tag{5.4.130}$$

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)s_k^T + s_k(s_k - H_k y_k)^T}{s_k^T y_k} - \frac{(s_k - H_k y_k)^T y_k}{(s_k^T y_k)^2} s_k s_k^T. \quad (5.4.131)$$

We employ the weighted norm

$$\|E\|_{BFGS} = \|E\|_{A^{1/2},F} = \|A^{1/2} E A^{1/2}\|_F, \quad (5.4.132)$$

where $A = \nabla^2 f(x^*)$.

Theorem 5.4.22 *Let $f : R^n \to R$ satisfy Assumption 5.4.2. Also let*

$$\mu\gamma\sigma(x_k, x_{k+1}) \leq \frac{1}{3} \quad (5.4.133)$$

in a neighborhood of x^*, where $\mu = \|\nabla^2 f(x^*)^{-1}\|$ and $\sigma(x_k, x_{k+1}) = \max\{\|x_k - x^*\|, \|x_{k+1} - x^*\|\}$. Then, there exist $\varepsilon > 0$ and $\delta > 0$ such that for $\|x_0 - x^*\| < \varepsilon$ and $\|H_0 - \nabla^2 f(x^*)^{-1}\|_{BFGS} < \delta$, BFGS method (5.4.130)–(5.4.131) is well-defined, and the produced sequence $\{x_k\}$ converges to x^* linearly. Further, if $\sum_{k=0}^{\infty} \|x_k - x^*\| < +\infty$, then the sequence $\{x_k\}$ converges to x^* superlinearly.

5.4.7 Local Convergence of Broyden's Class Methods

Finally, in this section, we discuss local convergence of Broyden's class methods.

Byrd, Nocedal and Yuan [47] proved the superlinear convergence of Broyden's class method. We state the theorem without proof.

Theorem 5.4.23 *Suppose that $f : R^n \to R$ is twice continuously differentiable on a convex set D and that $f(x)$ is uniformly convex, i.e., there exists $m > 0$ such that for any $x \in R^n$ and $u \in R^n$,*

$$u^T \nabla^2 f(x) u \geq m\|u\|^2.$$

Suppose also that there is a neighborhood $N(x^*, \varepsilon)$ of x^*, such that

$$\|\nabla^2 f(\bar{x}) - \nabla^2 f(x)\| \leq \gamma\|\bar{x} - x\|, \ \forall x, \bar{x} \in N(x^*, \varepsilon).$$

Then, for any positive definite matrix B_0, when line search satisfies Wolfe-Powell rule (5.4.30)–(5.4.33), the sequence $\{x_k\}$ generated by the restricted Broyden's class ($\theta \in (0, 1)$) converges to x^* superlinearly.

For Broyden's class with exact line search, we have

Theorem 5.4.24 *Suppose that the assumptions of Theorem 5.4.23 hold. When the exact line search is employed, the sequence $\{x_k\}$ generated by Broyden's class method converges to x^* superlinearly.*

Byrd, Liu, and Nocedal [43] established the following superlinear characterization in which the superlinear characterization (5.4.25) is replaced by (5.4.135) and (5.4.136).

Theorem 5.4.25 *Let iterates generated by*

$$x_{k+1} = x_k - \alpha_k B_k^{-1} g_k$$

converge to x^* with $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ positive definite. Then

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0 \tag{5.4.134}$$

if and only if

$$\lim_{k\to\infty} \cos^2\langle B_k^{-1} g_k, -\nabla^2 f(x^*)^{-1} g_k\rangle = 1 \tag{5.4.135}$$

and

$$\lim_{k\to\infty} \frac{s_k^T B_k s_k}{\alpha_k s_k^T y_k} = 1. \tag{5.4.136}$$

Proof. Suppose that (5.4.134) holds, then we have

$$\lim_{k\to\infty} \cos^2\langle B_k^{-1} g_k, x_k - x^*\rangle = 1. \tag{5.4.137}$$

Note also that

$$\lim_{k\to\infty} \cos^2\langle x_k - x^*, -\nabla^2 f(x^*)^{-1} g_k\rangle = 1. \tag{5.4.138}$$

Hence (5.4.135) holds.

By (5.4.134) and the positive definiteness of $\nabla^2 f(x^*)$, we have

$$\lim_{k\to\infty} \frac{\|g_k + y_k\|}{\|g_k\|} = 0,$$

which implies

$$\lim_{k\to\infty} \frac{s_k^T g_k + s_k^T y_k}{\|s_k\|\|g_k\|} = 0.$$

Therefore

$$\lim_{k\to\infty} \frac{-s_k^T g_k}{s_k^T y_k} = 1, \tag{5.4.139}$$

which means (5.4.136).

Conversely, assume that (5.4.135) and (5.4.136) hold. By (5.4.135) and (5.4.138) we deduce that (5.4.137) holds. Also, (5.4.136) means (5.4.139). Then, we obtain

$$\lim_{k\to\infty} \frac{s_k^T g_k + s_k^T \nabla^2 f(x^*) s_k}{s_k^T y_k} = 0,$$

which is

$$\lim_{k\to\infty} \frac{s_k^T \nabla^2 f(x^*)[s_k + \nabla^2 f(x^*)^{-1} g_k]}{s_k^T \nabla^2 f(x^*) s_k} = 0. \tag{5.4.140}$$

Then, (5.4.140) and (5.4.135) gives

$$\lim_{k\to\infty} \frac{\|s_k + \nabla^2 f(x^*)^{-1} g_k\|}{\|s_k\|} = 0, \tag{5.4.141}$$

which is equivalent to (5.4.134). We complete the proof. □

5.5 Self-Scaling Variable Metric (SSVM) Methods

5.5.1 Motivation to SSVM Method

We have seen that DFP method is a typical rank-two quasi-Newton method. However, numerical experiments show that its implementation is not ideal. Why? Below, we would like to give some analysis.

First, we clarify that the single-step convergence Theorem 3.1.5 of the steepest descent method is also true for various Newton-like methods. Let

$$f(x) = \frac{1}{2} x^T G x - b^T x, \tag{5.5.1}$$

where G is an $n \times n$ symmetric and positive definite matrix. Let the Newton-like method be defined by

$$x_{k+1} = x_k - \alpha_k H_k g_k, \tag{5.5.2}$$

where

$$g_k = G x_k - b, \tag{5.5.3}$$

$$\alpha_k = g_k^T H_k g_k / g_k^T H_k G H_k g_k, \tag{5.5.4}$$

then we have the following theorem.

Theorem 5.5.1 *Let x^* be a minimizer of the quadratic function (5.5.1), and let Newton-like methods be defined by (5.5.2). Then, the single-step convergence rate satisfies the following bound:*

$$\frac{f(x_{k+1}) - f(x^*)}{f(x_k) - f(x^*)} \leq \frac{(\lambda_1 - \lambda_n)^2}{(\lambda_1 + \lambda_n)^2}, \tag{5.5.5}$$

$$E(x_{k+1}) \leq \frac{(\lambda_1 - \lambda_n)^2}{(\lambda_1 + \lambda_n)^2} E(x_k), \tag{5.5.6}$$

where $E(x_k) = \frac{1}{2}(x_k - x^*)^T G(x_k - x^*)$, λ_1 and λ_n are the largest and the smallest eigenvalues of matrix $H_k G$ respectively.

Proof. Since

$$x^* = x_k - G^{-1} g_k \tag{5.5.7}$$

and

$$f(x_k) - f(x^*) = \frac{1}{2} g_k^T G^{-1} g_k, \tag{5.5.8}$$

and since the exact line search factor α_k is represented by (5.5.4), we have

$$f(x_{k+1}) = f(x_k) - \frac{1}{2} \alpha_k^2 g_k^T H_k G H_k g_k$$

and

$$f(x_{k+1}) - f(x^*) = \frac{1}{2} g_k^T G^{-1} g_k - \frac{1}{2} \alpha_k^2 g_k^T H_k G H_k g_k.$$

Hence

$$\begin{aligned} & \frac{f(x_{k+1}) - f(x^*)}{f(x_k) - f(x^*)} \\ = \; & 1 - \frac{(g_k^T H_k g_k)^2}{(g_k^T G^{-1} g_k)(g_k^T H_k G H_k g_k)} \\ = \; & 1 - \frac{(z_k^T z_k)^2}{(z_k^T (H_k^{-\frac{1}{2}})^T G^{-1} H_k^{-\frac{1}{2}} z_k)(z_k^T H_k^{\frac{1}{2}} G (H_k^{\frac{1}{2}})^T z_k)}, \end{aligned} \tag{5.5.9}$$

where $z_k = H_k^{\frac{1}{2}} g_k$. Then the conclusion (5.5.5) is obtained by using Kantorovich Theorem 3.1.10.

Similarly, we have

$$\frac{E(x_k) - E(x_{k+1})}{E(x_k)} = \frac{(z_k^T z_k)^2}{(z_k^T T_k z_k)(z_k^T T_k^{-1} z_k)},$$

where $T_k = H_k^{\frac{1}{2}} G H_k^{\frac{1}{2}}$. By using Kantorovich Theorem 3.1.10 and noting that $H_k G$ and T_k are similar, we immediately obtain the conclusion (5.5.6). □

From Theorem 5.5.1, we may see that if the condition number $\kappa(T_k)$ is very large, the single-step convergence rate will be very slow. In order to obtain a rapid rate in every iteration, we should make

$$\left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}\right)^2 \text{ or } \left[\frac{\kappa(T_k) - 1}{\kappa(T_k) + 1}\right]^2 \tag{5.5.10}$$

as small as possible, where $\kappa(T_k) = \lambda_1/\lambda_n$.

Second, let us observe carefully the DFP method. It is not difficult to see the fact that, usually, the eigenvalues of $H_0 G$ are greater than 1, and that DFP method and Broyden class method make one eigenvalue to being 1 in essence in each iteration. Hence, in the iterative procedure, a non-ideal eigen-ratio of $\{H_k G\}$ is produced. Also since $H_k G$ and T_k are similar, the eigen-ratio of $\{T_k\}$ is also non-ideal.

In fact, if we let

$$R_k = G^{\frac{1}{2}} H_k G^{\frac{1}{2}},\ r_k = G^{\frac{1}{2}} s_k, \tag{5.5.11}$$

then R_k is similar to $H_k G$, and further to T_k. By using $y_k = G^{\frac{1}{2}} s_k$, the DFP formula (5.1.30) is equivalent to

$$R_{k+1} = R_k - \frac{R_k r_k r_k^T R_k}{r_k^T R_k r_k} + \frac{r_k r_k^T}{r_k^T r_k}. \tag{5.5.12}$$

Let the eigenvalues of R_k satisfy $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0$. Let

$$P = R_k - \frac{R_k r_k r_k^T R_k}{r_k^T R_k r_k} \tag{5.5.13}$$

with eigenvalues $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$. Obviously, $P r_k = 0$. Then we have

$$\lambda_1 \geq \mu_1 \geq \lambda_2 \geq \mu_2 \geq \cdots \geq \lambda_n \geq \mu_n = 0. \tag{5.5.14}$$

From (5.5.12), it follows that

$$R_{k+1} = P + \frac{r_k r_k^T}{r_k^T r_k} \tag{5.5.15}$$

and

$$R_{k+1} r_k = r_k. \tag{5.5.16}$$

Since r_k is the eigenvector of P, and since P is symmetric, then all other eigenvectors of P are orthogonal to r_k. So, the unique different eigenvalue between R_{k+1} and P is the eigenvalue associated to r_k, which is 1. This shows that DFP method moves one eigenvalue of R_k to 1 in each iteration. Note that R_k is similar to $H_k G$, thus, it implies that if all eigenvalues of $H_0 G$ are greater than 1, then the eigen-ratio of $H_k G$ will worsen.

However, if $1 \in [\lambda_n, \lambda_1]$, then, it follows from the above discussion that the eigenvalues $\mu_1, \mu_2, \cdots, \mu_{n-1}$ of R_{k+1} and 1 will be contained in $[\lambda_n, \lambda_1]$. Hence, in this case, the eigen-ratio of $H_k G$ will not worsen. This conclusion is true for updates of Broyden class with $0 \leq \phi \leq 1$.

Theorem 5.5.2 *Let $\lambda_1, \lambda_2, \cdots, \lambda_n$ be eigenvalues of $H_k G$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0$. Suppose that $1 \in [\lambda_n, \lambda_1]$. Then, for any ϕ with $0 \leq \phi \leq 1$, the eigenvalues of $H_{k+1}^{\phi} G$ are contained in $[\lambda_n, \lambda_1]$, where H_{k+1}^{ϕ} is the Broyden class update defined by (5.2.4).*

Proof. The case $\phi = 0$ has been proved as before.

Now we consider the case $\phi = 1$ (BFGS update). The BFGS formula (5.1.45) can be written as

$$H_{k+1}^{-1} = H_k^{-1} + \frac{y_k y_k^T}{s_k^T y_k} - \frac{H_k^{-1} s_k s_k^T H_k^{-1}}{s_k^T H_k^{-1} s_k},$$

which is equivalent to

$$R_{k+1}^{-1} = R_k^{-1} - \frac{R_k^{-1} r_k r_k^T R_k^{-1}}{r_k^T R_k^{-1} r_k} + \frac{r_k r_k^T}{r_k^T r_k}. \tag{5.5.17}$$

Since the eigenvalues of R_k^{-1} satisfy

$$\frac{1}{\lambda_1} \leq \frac{1}{\lambda_2} \leq \cdots \leq \frac{1}{\lambda_n},$$

then we have $1 \in [1/\lambda_1, 1/\lambda_n]$. Similar to the above discussion, we know that if the eigenvalues of R_{k+1}^{-1} satisfy $1/\mu_1 \leq 1/\mu_2 \leq \cdots \leq 1/\mu_n$, then these eigenvalues are contained in $[1/\lambda_1, 1/\lambda_n]$. Hence, we have that $1/\lambda_1 \leq 1/\mu_1$ and $1/\lambda_n \geq 1/\mu_n$, i.e., $\mu_n \geq \lambda_n$ and $\mu_1 \leq \lambda_1$. This shows that all eigenvalues of R_{k+1} are contained in $[\lambda_n, \lambda_1]$. Therefore, the conclusion holds for $\phi = 1$.

Finally, we know that Broyden class updating formula (5.2.4) is equivalent to

$$R_{k+1}^{\phi} = R_k - \frac{R_k r_k r_k^T R_k}{r_k^T R_k r_k} + \frac{r_k r_k^T}{r_k^T r_k} + \phi u_k u_k^T, \tag{5.5.18}$$

where

$$u_k = G^{\frac{1}{2}} v_k = (r_k^T R_k r_k)^{\frac{1}{2}} \left[\frac{r_k}{r_k^T r_k} - \frac{R_k r_k}{r_k^T R_k r_k} \right]. \tag{5.5.19}$$

Clearly, the eigenvalues of R_{k+1}^{ϕ} are increasing monotonically as k increases. Since, for $\phi = 0$ and $\phi = 1$, the eigenvalues of R_{k+1}^{ϕ} are contained in $[\lambda_n, \lambda_1]$, then, for $0 \leq \phi \leq 1$, the eigenvalues of R_{k+1}^{ϕ} are also contained in $[\lambda_n, \lambda_1]$. Thus, from the fact that R_{k+1}^{ϕ} and $H_{k+1}^{\phi} G$ are similar, we obtain the conclusion. □

The above theorem says that if we scale the matrix H_k such that the eigenvalues of $H_k G$ satisfy $1 \in [\lambda_n, \lambda_1]$, the eigenvalue structure of R_{k+1}^{ϕ} will be improved.

Obviously, for a quadratic function, it is enough to scale only the initial matrix H_0. However, in general, it is useful to scale each H_k.

5.5.2 Self-Scaling Variable Metric (SSVM) Method

In this section we describe SSVM method due to Oren [237]. Multiplying H_k by γ_k and then replacing H_k by $\gamma_k H_k$ in (5.2.2) yield

$$H_{k+1}^{(\phi,\gamma_k)} = \left(H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \phi v_k v_k^T \right) \gamma_k + \frac{s_k s_k^T}{s_k^T y_k}, \tag{5.5.20}$$

where

$$v_k = (y_k^T H_k y_k)^{\frac{1}{2}} [s_k / s_k^T y_k - H_k y_k / y_k^T H_k y_k],$$

where ϕ is a parameter of Broyden class and γ_k a self-scaling parameter. The formula (5.5.20) is referred to as the self-scaling variable metric (SSVM) formula. When $\gamma_k = 1$, it is reduced to Broyden class update.

Algorithm 5.5.3 *(SSVM Algorithm)*

Step 0. Given an initial matrix H_0 and a starting point x_0. Set $k = 0$.

Step 1. Set $d_k = -H_k g_k$.

Step 2. Find stepsize α_k, and set $x_{k+1} = x_k + \alpha_k d_k$, compute g_{k+1} and set $y_k = g_{k+1} - g_k$.

Step 3. Choose Broyden's class parameter $\phi \geq 0$ and self-scaling parameter $\gamma_k > 0$, and compute $H_{k+1}^{(\phi,\gamma_k)}$ by (5.5.20).

Step 4. $k := k + 1$, go to Step 1. □

Similar to the discussion of DFP method in §5.1, we can prove that the SSVM method has the following properties. The proof is omitted.

Theorem 5.5.4 *(Properties of SSVM Method)*

1. *If H_k is positive definite and $s_k^T y_k > 0$, then when $\phi \geq 0$ and $\gamma_k > 0$, the matrix $H_{k+1}^{(\phi,\gamma_k)}$ produced by (5.5.20) is positive definite.*

2. *If $f(x)$ is a quadratic function with Hessian G, the vectors $s_0, s_1, \cdots, s_{n-1}$ produced by SSVM method are G-conjugate, i.e., satisfy*

$$s_i^T G s_j = 0,\ i \neq j;\ i, j = 0, 1, \cdots, n-1, \tag{5.5.21}$$

and for each k, $s_0, s_1, \cdots, s_k$ are the eigenvalues of $H_{k+1}^{(\phi,\gamma_k)} G$, i.e., satisfy

$$H_{k+1}^{(\phi,\gamma_k)} G s_i = \bar{\gamma}_{i,k} s_i,\ 0 < i < k, \tag{5.5.22}$$

where $\bar{\gamma}_{i,k} = \prod_{j=i+1}^{k} \gamma_j$, $\bar{\gamma}_{ii} = 1$.

This theorem shows that although the property $H_n^{(\phi,\gamma_{n-1})} = G^{-1}$ is not retained for quadratic functions by SSVM method, the property of conjugate directions is still retained. Therefore, for quadratic functions, the sequence generated from SSVM method converges to a minimizer in at most n steps.

5.5.3 Choices of the Scaling Factor

Now, the problem is how to choose a suitable scaling factor. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0$ be eigenvalues of $H_k G$. Clearly, they are also the eigenvalues of R_k. We hope to choose a suitable scaling factor which is used to multiply H_k, such that 1 is contained among the new eigenvalues and thus the eigenstructure is improved. Therefore we get $\kappa(R_{k+1}^{\phi}) \leq \kappa(R_k)$. The following theorem is a consequence of Theorem 5.5.2.

Theorem 5.5.5 *Let $\phi \in [0,1]$ and $\gamma_k > 0$. Let R_k and R_{k+1}^{ϕ} be defined respectively by (5.5.11) and (5.5.24). Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ and $\mu_1^{\phi} \geq \mu_2^{\phi} \geq \cdots \geq \mu_n^{\phi}$ be eigenvalues of R_k and R_{k+1}^{ϕ} respectively. Then the following statements hold.*

1. *If $\gamma_k \lambda_n \geq 1$, then $\mu_n^{\phi} = 1$ and $1 \leq \gamma_k \lambda_{i+1} \leq \mu_i^{\phi} \leq \gamma_k \lambda_i, i = 1, 2, \cdots, n-1$.*
2. *If $\gamma_k \lambda_1 \leq 1$, then $\mu_1^{\phi} = 1$ and $\gamma_k \lambda_i \leq \mu_i^{\phi} \leq \gamma_k \lambda_{i-1} \leq 1,\ i = 2, 3, \cdots, n$.*
3. *If $\gamma_k \lambda_n \leq 1 \leq \gamma_k \lambda_1$ and i_0 is an index with $\gamma_k \lambda_{i_0+1} \leq 1 \leq \gamma_k \lambda_{i_0}$, then*

$$\begin{aligned} \gamma_k \lambda_1 \geq \mu_1^{\phi} \geq \gamma_k \lambda_2 &\geq \mu_2^{\phi} \geq \cdots \geq \gamma_k \lambda_{i_0} \geq \mu_{i_0} \geq 1 \geq \mu_{i_0+1} \\ &\geq \gamma_k \lambda_{i_0+1} \geq \cdots \geq \gamma_k \lambda_n, \end{aligned} \tag{5.5.23}$$

and there is at least one eigenvalue in $\mu_{i_0}^{\phi}$ and $\mu_{i_0+1}^{\phi}$ which equals 1.

Proof. This theorem is a direct consequence of Theorem 5.5.2. Since SSVM method is equivalent to

$$R_{k+1}^{\phi} = \left(R_k - \frac{R_k r_k r_k^T R_k}{r_k^T R_k r_k} + \phi u_k u_k^T \right) \gamma_k + \frac{r_k r_k^T}{r_k^T r_k}, \tag{5.5.24}$$

where r_k and u_k are defined respectively by (5.5.11) and (5.5.19), the above expression is just obtained by replacing R_k by $\gamma_k R_k$ in (5.5.18). Therefore, from Theorem 5.5.2, replacing $\lambda_1, \lambda_2, \cdots, \lambda_n$ by use of $\gamma_k \lambda_1, \cdots, \gamma_k \lambda_n$ gives our conclusion. □

Corollary 5.5.6 *Let $\phi \in [0,1]$ and $\gamma_k = 1$. Then*

$$|\mu_k^{\phi} - 1| \leq |\lambda_k - 1|. \tag{5.5.25}$$

Proof. From Theorem 5.5.5, for $\gamma_k = 1$, one of the following cases will hold:

(a) $\lambda_i \geq \mu_i^{\phi} \geq 1$;

(b) $\lambda_i \leq \mu_i^{\phi} \leq 1$.

Hence the conclusion (5.5.25) is obtained. □

Obviously, if we choose γ_k such that

$$\lambda_n \leq \frac{1}{\gamma_k} \leq \lambda_1, \tag{5.5.26}$$

we have

$$\gamma_k \lambda_n \leq 1 \leq \gamma_k \lambda_1, \tag{5.5.27}$$

which says that 1 is included in the interval of scaled eigenvalues. In addition, we have

Corollary 5.5.7 *Let $\phi \in [0, 1]$ and $\gamma_k > 0$. Let $\kappa(\cdot)$ denote the condition number. If $\lambda_n \leq \frac{1}{\gamma_k} \leq \lambda_1$, then, for (5.5.24), we have*

$$\kappa(R_{k+1}^{\phi}) \leq \kappa(R_k). \tag{5.5.28}$$

Proof. From Theorem 5.5.5 (3), it follows that

$$\gamma_k \lambda_1 \geq \mu_1^{\phi} \geq 1 \geq \mu_n^{\phi} \geq \gamma_k \lambda_n, \tag{5.5.29}$$

which gives

$$\frac{\mu_1^{\phi}}{\mu_n^{\phi}} \leq \frac{\lambda_1}{\lambda_n}.$$

Thus, we complete the proof. □

In the above discussion about the condition of γ_k, we always restrict the Broyden class parameter $\phi \in [0, 1]$. In fact, this restriction is sufficient and also necessary for the statement that if $\lambda_n \leq \frac{1}{\gamma_k} \leq \lambda_1$, then $\kappa(R_{k+1}^{\phi}) \leq \kappa(R_k)$ and $H_{k+1}^{(\phi,\gamma_k)}$ is positive definite.

Corollary 5.5.7 says that $\lambda_n \leq \frac{1}{\gamma_k} \leq \lambda_1$ is a suitable requirement to choose a scaling factor. Note that

$$\frac{r_k^T R_k r_k}{r_k^T r_k} = \frac{y_k^T H_k y_k}{s_k^T y_k}$$

and

$$\lambda_n \leq \frac{r_k^T R_k r_k}{r_k^T r_k} \leq \lambda_1,$$

it follows that

$$\gamma_k = \frac{s_k^T y_k}{y_k^T H_k y_k} \tag{5.5.30}$$

is a suitable scaling factor. Similarly, since

$$\frac{r_k^T R_k^{-1} r_k}{r_k^T r_k} = \frac{s_k^T H_k^{-1} s_k}{s_k^T y_k}$$

and

$$\frac{1}{\lambda_1} \leq \frac{r_k^T R_k^{-1} r_k}{r_k^T r_k} \leq \frac{1}{\lambda_n},$$

we have that

$$\gamma_k = \frac{s_k^T H_k^{-1} s_k}{s_k^T y_k} = -\frac{\alpha_k s_k^T g_k}{s_k^T y_k} = \frac{s_k^T g_k}{g_k^T H_k y_k} \tag{5.5.31}$$

is also a suitable scaling factor. Noting that when α_k is an optimal stepsize, we have that $s_k^T y_k = -s_k^T g_k$, and thus

$$\gamma_k = \alpha_k. \tag{5.5.32}$$

The above (5.5.32) shows an interesting fact, that we may choose directly an optimal stepsize as a scaling factor.

For any $\omega \in [0, 1]$,

$$\gamma_k = (1-\omega)\frac{s_k^T y_k}{y_k^T H_k y_k} + \omega \frac{s_k^T H_k^{-1} s_k}{s_k^T y_k} \tag{5.5.33}$$

is a convex combination of (5.5.30) and (5.5.31). Hence (5.5.33) gives a convex class of suitable scaling factors. For this convex class, Oren [239] presented the following switch rule of parameters ϕ and ω.

If $\frac{s_k^T y_k}{y_k^T H_k y_k} > 1$, choose $\phi = 1$ and $\omega = 0$,
(i.e., $\phi = 1, \gamma_k = s_k^T y_k / y_k^T H_k y_k$).
If $\frac{s_k^T H_k^{-1} s_k}{s_k^T y_k} < 1$, choose $\phi = 0, \omega = 1$.
(i.e., $\phi = 0, \gamma_k = \frac{s_k^T H_k^{-1} s_k}{s_k^T y_k}$).

If $\frac{s_k^T y_k}{y_k^T H_k y_k} \leq 1 \leq \frac{s_k^T H_k^{-1} s_k}{s_k^T y_k}$, choose

$$\omega = \phi = \frac{s_k^T y_k (y_k^T H_k y_k - s_k^T y_k)}{(s_k^T H_k^{-1} s_k)(y_k^T H_k y_k) - (s_k^T y_k)^2}, \text{ (i.e., } \gamma_k = 1).$$

Another technique is an initial scaling method presented by Shanno and Phua [306]. At the beginning, set $H_0 = I$, and the stepsize α_0 is determined by some line search, such that the objective function descends sufficiently. Before computing H_1, instead of H_0, we use

$$\hat{H}_0 = \alpha_0 H_0, \tag{5.5.34}$$

and compute H_1 from $\hat{H}_0$, where α_0 is a stepsize or determined by

$$\alpha_0 := \gamma_0 = \frac{s_0^T y_0}{y_0^T H_0 y_0}. \tag{5.5.35}$$

The difference between the initial scaling and SSVM is that SSVM does process scaling in each iteration, but the initial scaling method does only at the beginning. Numerical experiments show that the initial scaling is simple and effective for a lot of problems in which the curvature changes smoothly.

By the way, a special self-scaling BFGS formula

$$B_{k+1} = \frac{s_k^T y_k}{s_k^T B_k s_k} \left(B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} \right) + \frac{y_k y_k^T}{s_k^T y_k} \tag{5.5.36}$$

is used widely in practice.

5.6 Sparse Quasi-Newton Methods

Schubert [303] first extended quasi-Newton update to an unsymmetric sparse matrix and proposed a sparse quasi-Newton method for solving nonlinear equations. Powell and Toint [276], Toint [341] derived sparse quasi-Newton update respectively, and Steihaug [321] presented a sparse quasi-Newton method with preconditioning and established the convergence.

The sparse quasi-Newton method requires generating sparse quasi-Newton updates which have the same (or similar) sparsity pattern as the true Hessian. It means that the current Hessian approximation B_k reflects the nonzero structure of the true Hessian, i.e.,

$$(B_k)_{ij} = 0 \text{ for } (i,j) \in I, \tag{5.6.1}$$

where

$$I \triangleq \{(i,j) \,|\, [\nabla^2 f(x)]_{ij} = 0\} \tag{5.6.2}$$

is a set of integer pairs. We also define

$$J \triangleq \{(i,j) \,|\, [\nabla^2 f(x)]_{ij} \neq 0\}. \tag{5.6.3}$$

It says that J, a set of integer pairs, is a complement of I. So, we demand that B_{k+1} satisfies the quasi-Newton condition

$$B_{k+1} s_k = y_k, \tag{5.6.4}$$

and keeps symmetry and sparsity. Neglecting the subscript, we would like to find $\bar{B}$, such that

$$\bar{B} = B + E, \tag{5.6.5}$$

where E satisfies

$$Es = y - Bs, \tag{5.6.6}$$
$$E = E^T, \tag{5.6.7}$$
$$E_{ij} = 0, \ (i,j) \in I, \tag{5.6.8}$$

where E_{ij} are elements of the matrix E. If we determined E, we can get $\bar{B}$ from (5.6.5). However, (5.6.6)–(5.6.8) cannot determine completely the matrix E. So, to this end, we require that $\bar{B}$ is as close as possible to B in Frobenius norm. Therefore, we consider the following minimization problem:

$$\min \quad \frac{1}{2}\|E\|_F^2 \tag{5.6.9}$$
$$\text{s.t.} \quad Es = r, \tag{5.6.10}$$
$$E = E^T, \tag{5.6.11}$$
$$E_{ij} = 0, \ (i,j) \in I, \tag{5.6.12}$$

where r is assumed to be

$$r = y - Bs. \tag{5.6.13}$$

In the left part of the section, we denote the j-th component of the vector s by s_j, and define the component of vector $s(i)$ as

$$s(i)_j = \begin{cases} s_j, & (i,j) \in J \\ 0, & (i,j) \in I. \end{cases} \tag{5.6.14}$$

Then the condition (5.6.10) can be written as

$$\sum_{j=1}^{n} E_{ij} s(i)_j = r_i, \ i = 1, \cdots, n. \tag{5.6.15}$$

In order to let E be symmetric, take

$$E = \frac{1}{2}(A + A^T). \tag{5.6.16}$$

Then the problem (5.6.9)-(5.6.12) becomes the following problem: finding a matrix A, such that

$$\min \quad \frac{1}{8}\|A + A^T\|_F \tag{5.6.17}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} (A_{ij} + A_{ji}) s(i)_j = 2r_i, \ i = 1, \cdots, n, \tag{5.6.18}$$

where A_{ij} denote the elements of A.

Now, we discuss solving the problem (5.6.17)–(5.6.18). The Lagrangian function is

$$\begin{aligned} \Phi(A, \lambda) &= \frac{1}{8} \sum_{i=1}^{n} \sum_{j=1}^{n} (A_{ij}^2 + A_{ji}^2 + 2A_{ij}A_{ji}) \\ &\quad - \sum_{i=1}^{n} \lambda_i \left[\sum_{j=1}^{n} (A_{ij} + A_{ji}) s(i)_j - 2r_i \right]. \end{aligned} \tag{5.6.19}$$

Setting the derivative with respect to A_{ij} to be zero, we have

$$\begin{aligned} \frac{\partial \Phi(A, \lambda)}{\partial A_{ij}} &= \frac{1}{2}(A_{ij} + A_{ji}) - \lambda_i s(i)_j - \lambda_j s(j)_i = 0, \\ &\quad i, j = 1, \cdots, n. \end{aligned} \tag{5.6.20}$$

By using (5.6.16), the above expression is just

$$E_{ij} = \lambda_i s(i)_j + \lambda_j s(j)_i, \ i, j = 1, \cdots, n. \tag{5.6.21}$$

In place of (5.6.18), we employ (5.6.15). Substituting (5.6.21) into (5.6.15), we obtain

$$\sum_{j=1}^{n} [\lambda_i s(i)_j + \lambda_j s(j)_i] s(i)_j = r_i, \ i = 1, \cdots, n, \tag{5.6.22}$$

which is

$$\lambda_i \sum_{j=1}^{n} [s(i)_j]^2 + \sum_{j=1}^{n} \lambda_j s(j)_i s(i)_j = r_i, \ i = 1, \cdots, n. \tag{5.6.23}$$

Thus, we derive the update formula

$$\bar{B} = B + E, \tag{5.6.24}$$

which is, from (5.6.21), that

$$\bar{B} = B + \sum_{i=1}^{n} \lambda_i [e_i s(i)^T + s(i) e_i^T], \tag{5.6.25}$$

where e_i is the i-th unit vector and λ is a Lagrange multiplier vector satisfying

$$Q\lambda = r, \tag{5.6.26}$$

where

$$Q = \sum_{i=1}^{n} (s(i)^T s e_i + e_i^T s s(i)) e_i^T. \tag{5.6.27}$$

In fact, as long as we notice that

$$\begin{aligned} Q\lambda &= r = Es = \sum_{i=1}^{n} \lambda_i [e_i s(i)^T s + s(i) e_i^T s] \\ &= \sum_{i=1}^{n} [s(i)^T s e_i + e_i^T s s(i)] e_i^T \lambda, \end{aligned}$$

we can immediately obtain (5.6.27).

The matrix Q defined above satisfies symmetry, sparsity and positive definiteness. The properties of symmetry and sparsity can be seen direct from (5.6.27). As to the positive definiteness of Q, we give the following theorem.

Theorem 5.6.1 *If all vectors $s(i)$ $(i = 1, \cdots, n)$ are nonzero, then the matrix Q is positive definite, that is*

$$z^T Q z > 0, \ \forall z \in R^n, z \neq 0. \tag{5.6.28}$$

Proof. Take $z \neq 0, z \in R^n$. Let z_i denote the components of vector z. From (5.6.27),

$$\begin{aligned} z^T Q z &= \sum_{i=1}^{n}\sum_{j=1}^{n} z_i^T Q_{ij} z_j \\ &= \sum_{i=1}^{n}\sum_{i=1}^{n} z_i s(i)_j s(j)_i z_j + \sum_{i=1}^{n}\sum_{j=1}^{n} [s(i)_j]^2 z_i^2 \\ &= \sum_{(i,j)\in J} [z_i s_i s_j z_j + z_i^2 s_j^2] \\ &= \frac{1}{2}\sum_{(i,j)\in J} [s_i z_j + s_j z_i]^2 \\ &= 2\sum_{i=1}^{n} z_i^2 s_i^2 + \frac{1}{2}\sum_{\substack{(i,j)\in J \\ i\neq j}} (z_i s_j + z_j s_i)^2 \\ &\geq 0. \end{aligned} \tag{5.6.29}$$

Suppose that $z^T Q z = 0$; since $z \neq 0$, there exists a component of z, for example, $z_k \neq 0$, such that by (5.6.29) we have

$$z_k s_k = 0, \tag{5.6.30}$$

$$z_k s_j + z_j s_k = 0, \ (k, j) \in J, j \neq k. \tag{5.6.31}$$

Thus, $s_k = 0$. Furthermore, $s_j = 0, j \neq k, (k, j) \in J$. This is equivalent to $s(k) = 0$, which contradicts the assumption. We complete the proof. □

Since Q is positive definite, it follows from (5.6.21) and (5.6.26) that

$$E_{ij} = (Q^{-1}r)_i s(i)_j + (Q^{-1}r)_j s(j)_i, \tag{5.6.32}$$

which can be written as

$$E_{ij} = \begin{cases} 0, & (i,j) \in I, \\ \lambda_i s_j + \lambda_j s_i, & (i,j) \in J. \end{cases} \tag{5.6.33}$$

The above discussion gives the derivation of general sparse quasi-Newton update.

Now, we turn to the sparse PSB update.

Let $F : R^n \to R^n$. For solving sparse nonlinear equations $F(x) = 0$, Schuburt [303] first suggested that Broyden's rank-one update

$$\bar{B} = B + \frac{(y - Bs)s^T}{s^T s} \tag{5.6.34}$$

can be written in the following form

$$\bar{B} = B + \sum_{i=1}^{n} e_i e_i^T \frac{(y - Bs)s^T}{s^T s}, \tag{5.6.35}$$

which is an update by row, where e_i is the i-th unit vector. By use of notation $s(i)$, one knows that

$$\bar{B} = B + \sum_{i=1}^{n} e_i e_i^T \frac{(y - Bs)s(i)^T}{s(i)^T s} \tag{5.6.36}$$

satisfies the quasi-Newton condition $\bar{B}s = y$, and has the sparsity pattern desired.

The general form of Schuburt sparse update is

$$\bar{B} = B + \sum_{i=1}^{n} \alpha_i e_i z(i)^T, \tag{5.6.37}$$

where

$$\alpha_i = \frac{e_i^T(y - Bs)}{s(i)^T s},\ z(i)_j = \begin{cases} z_j, & (i,j) \in J, \\ 0, & (i,j) \in I. \end{cases} \tag{5.6.38}$$

Now we employ symmetrization to (5.6.37) and deduce that

$$\bar{B} = B + \sum_{i=1}^{n} \alpha_i (e_i z(i)^T + z(i)e_i^T). \tag{5.6.39}$$

Let us choose α_i, such that $\bar{B}$ satisfies the quasi-Newton condition. Obviously, $\bar{B}$ is symmetric and satisfies sparsity.

Similar to the discussion before, we can obtain that α satisfies

$$T\alpha = r, \tag{5.6.40}$$

where

$$T = \sum_{i=1}^{n} [z(i)^T s e_i + e_i^T s z(i)] e_i^T. \tag{5.6.41}$$

In particular, setting $z(i) = s(i)$, we immediately get (5.6.25)–(5.6.27), which is sparse PSB update.

Next, let us proceed to the sparse BFGS update.

For clarity, we repeat the BFGS update given in (5.1.45):

$$\bar{B} = B + \frac{yy^T}{s^T y} - \frac{Bss^T B}{s^T Bs}, \tag{5.6.42}$$

where B is assumed to have some sparsity pattern. Since the $\bar{B}$ defined by the above formula has not such a structure, we modify it and make it have this kind of sparsity structure. Define

$$\hat{B} = \bar{B} + E. \tag{5.6.43}$$

We demand that $\hat{B}$ satisfies the following conditions:

(i) $\hat{B}$ satisfies the quasi-Newton condition.

(ii) $\hat{B}$ is symmetric.

(iii) $\hat{B}$ is the closest to $\bar{B}$ in Frobenius norm.

So, we consider the following minimization problem:

$$\begin{aligned} \min \quad & \|E\|_F = \frac{1}{2}\mathrm{Tr}(E^T E) && (5.6.44)\\ \text{s.t.} \quad & Es = 0, && (5.6.45)\\ & E_{ij} = -\bar{B}_{ij},\ (i,j) \in I, && (5.6.46)\\ & E = E^T. && (5.6.47) \end{aligned}$$

To solve (5.6.44)–(5.6.47), we define the Lagrange function Φ as follows:

$$\begin{aligned} \Phi(E,\mu,\Lambda,\lambda) &= \frac{1}{2}\mathrm{Tr}(E^T E) - \mathrm{Tr}(Es\mu^T) - \mathrm{Tr}(\Lambda(E - E^T)) \\ &\quad - \sum_{(i,j)\in I} \lambda_{ij}\mathrm{Tr}(E + \bar{B})e_j e_i^T \\ &= \frac{1}{2}\mathrm{Tr}(E^T E) - \mathrm{Tr}(Es\mu^T) - \mathrm{Tr}(\Lambda(E - E^T)) \\ &\quad -\mathrm{Tr}(\Lambda^T(E + \bar{B})), \end{aligned} \tag{5.6.48}$$

where μ is the multiplier vector, Λ and Δ are multiplier matrices, and λ_{ij} are the elements of the matrix Δ. When $(i,j) \in J, \lambda_{ij} = 0$.

Differentiating (5.6.48) and setting $\frac{\partial \Phi}{\partial E} = 0$, we have

$$\frac{\partial \Phi}{\partial E} = E - s\mu^T - \Lambda^T + \Lambda - \Delta = 0, \tag{5.6.49}$$

which gives

$$E = s\mu^T + \Delta - \Lambda + \Lambda^T \tag{5.6.50}$$

and

$$E^T = \mu s^T + \Delta^T - \Lambda^T + \Lambda. \tag{5.6.51}$$

By using (5.6.47), we get

$$E - E^T = s\mu^T + \Delta - \Delta^T + 2(\Lambda^T - \Lambda) = 0, \tag{5.6.52}$$

that is

$$\Lambda - \Lambda^T = \frac{1}{2}(s\mu^T - \mu s^T + \Delta - \Delta^T). \tag{5.6.53}$$

By use of (5.6.50) and (5.6.53), we have

$$E = \frac{1}{2}(s\mu^T + \mu s^T + \Delta + \Delta^T), \tag{5.6.54}$$

which gives, by (5.6.46), that

$$e_i^T E e_j = \frac{1}{2}(e_i^T \mu s^T e_j + e_i^T s\mu^T e_j + \lambda_{ij} + \lambda_{ji}) = -\bar{B}_{ij},$$

that is

$$\lambda_{ij} + \lambda_{ji} = -2\bar{B}_{ij} - e_i^T \mu s^T e_j - e_i^T s\mu^T e_j, \ (i,j) \in I. \tag{5.6.55}$$

The above expression can be written in matrix form:

$$\Delta + \Delta^T = -2\bar{B}_{ij}^{(I)} - \sum_{i=1}^{n} e_i e_i^T (\mu \hat{s}(i)^T + s\hat{\mu}(i)^T), \tag{5.6.56}$$

where

$$\bar{B}_{ij}^{(I)} = \begin{cases} \bar{B}_{ij}, & (i,j) \in I, \\ 0, & (i,j) \in J, \end{cases} \tag{5.6.57}$$

$$\hat{s}(i)_j = \begin{cases} s_j, & (i,j) \in I, \\ 0, & (i,j) \in J, \end{cases} \quad \hat{\mu}(i)_j = \begin{cases} \mu_j, & (i,j) \in I, \\ 0, & (i,j) \in J. \end{cases} \tag{5.6.58}$$

By (5.6.54) and (5.6.56), we deduce that

$$\begin{aligned} E &= \frac{1}{2}\left[\sum_{i=1}^{n} e_i e_i^T (\mu s^T + s\mu^T) - 2\bar{B}^{(I)} - \sum_{i=1}^{n} e_i e_i^T (\mu \hat{s}(i)^T + s\hat{\mu}(i)^T)\right] \\ &= \frac{1}{2}\left[\sum_{i=1}^{n} e_i e_i^T (\mu s(i)^T + s\mu(i)^T) - 2\bar{B}^{(I)}\right], \end{aligned} \tag{5.6.59}$$

where

$$\mu(i)_j = \begin{cases} \mu_i, & (i,j) \in J, \\ 0, & (i,j) \in I. \end{cases} \tag{5.6.60}$$

Also, by (5.6.45), we have

$$Es = \frac{1}{2}\left[\sum_{i=1}^{n} e_i e_i^T (\mu s(i)^T + s\mu(i)^T) - 2\bar{B}^{(I)}\right] s = 0, \tag{5.6.61}$$

which is

$$\sum_{i=1}^{n} e_i e_i^T (\mu s(i)^T s + s\mu(i)^T s) = 2\bar{B}^{(I)} s. \tag{5.6.62}$$

Note that

$$\sum_{i=1}^{n} e_i e_i^T s\mu(i)^T = \sum_{i=1}^{n} \mu_i s(i) e_i^T,$$

and we can rewrite (5.6.62) as

$$\sum_{i=1}^{n} \mu_i (e_i s(i)^T + s(i) e_i^T) s = t, \tag{5.6.63}$$

where $t = 2\bar{B}^{(I)} s$.

Then, provided that we solve (5.6.63) for μ_i and substitute μ_i into (5.6.59), we can deduce that

$$E = \frac{1}{2} \sum_{i=1}^{n} \mu_i (e_i s(i)^T + s(i) e_i^T) - \bar{B}^{(I)}. \tag{5.6.64}$$

Thus,

$$\begin{aligned} \hat{B} &= \bar{B} + E \\ &= \bar{B} + \frac{1}{2} \sum_{i=1}^{n} \mu_i (e_i s(i)^T + s(i) e_i^T) - \bar{B}^{(I)} \\ &= \bar{B}^{(J)} + \frac{1}{2} \sum_{i=1}^{n} \mu_i (e_i s(i)^T + s(i) e_i^T) \end{aligned} \tag{5.6.65}$$

where

$$\bar{B}_{ij}^{(J)} = \begin{cases} \bar{B}_{ij}, & (i,j) \in J, \\ 0, & (i,j) \in I. \end{cases} \tag{5.6.66}$$

The formula (5.6.65) is said to be sparse BFGS update. Similarly, we can derive the sparse update for other quasi-Newton updates.

Note that the above formula (5.6.65) is obtained by minimization of problem (5.6.44)–(5.6.47) in Frobenius norm. Instead, we consider this minimization problem in the weighted Frobenius norm, i.e., consider the problem

$$\begin{aligned} \min \quad & \|E\|_{W,F} = \frac{1}{2}\mathrm{Tr}(WE^TWE) && (5.6.67)\\ \text{s.t.} \quad & Es = 0, && (5.6.68)\\ & E_{ij} = -\bar{B}_{ij},\ (i,j) \in I, && (5.6.69)\\ & E = E^T. && (5.6.70) \end{aligned}$$

Then, corresponding to (5.6.54), we have

$$E = \frac{1}{2}[z(s^TM) + (Ms)z^T + M(\Delta + \Delta^T)M], \tag{5.6.71}$$

where $M = W^{-1}, z = M\mu$.

Set $p = Ms$. We can obtain that if and only if $M(\Delta+\Delta^T)M$ and $\Delta+\Delta^T$ have the same sparsity pattern, the solution of (5.6.67)–(5.6.70) is

$$\hat{B} = \bar{B}^{(I)} + \sum_{i=1}^{n} z_i(e_ip(i)^T + p(i)e_i^T), \tag{5.6.72}$$

where

$$p(i)_j = \begin{cases} p_j, & (i,j) \in J, \\ 0, & (i,j) \in I, \end{cases} \tag{5.6.73}$$

z_i is the solution of the equations

$$\sum_{i=1}^{n} z_i(e_ip(i)^T + p(i)e_i^T)s = 2\bar{B}^{(I)}s. \tag{5.6.74}$$

Clearly, if W is a positive definite and diagonal matrix, $M(\Delta + \Delta^T)M$ and $\Delta + \Delta^T$ have the same sparsity structure.

Toint [341] considered sparse quasi-Newton update in the case that the weighted matrix is a non-diagonal matrix. For solving efficiently the sparse equations about μ_i, Steihaug [321] presented a preconditioned conjugate gradient method to solve the linear equations.

An alternative approach is to relax the quasi-Newton equation, making sure that it is approximately satisfied along the last few steps rather than

requiring it to hold strictly on the latest step. Define the $n \times m$ matrices S_k and Y_k by

$$S_k = [s_{k-m}, \cdots, s_{k-1}],\ Y_k = [y_{k-m}, \cdots, y_{k-1}]. \tag{5.6.75}$$

We ask B_{k+1} to be a solution of

$$\begin{aligned} \min \quad & \|\bar{B}S_k - Y_k\|_F^2 && (5.6.76)\\ \text{s.t.} \quad & \bar{B} = \bar{B}^T, && (5.6.77)\\ & \bar{B}_{ij} = 0,\ (i,j) \in I. && (5.6.78) \end{aligned}$$

In general, sparse quasi-Newton methods lost some advantages of dense quasi-Newton methods.

(1) Because of the complexity of the sparse pattern, the modified matrix E is a rank-n matrix, rather than a rank-two matrix.

(2) To compute the matrix E, we must solve a sparse linear equation about μ_i.

(3) The positive definiteness of the update matrix $\{B_k\}$ cannot be guaranteed.

(4) So far, the numerical performance is not ideal.

We think that it is still a challenging topic to solve large-scale optimization problems by studying sparse quasi-Newton methods.

5.7 Limited Memory BFGS Method

Limited memory quasi-Newton methods are useful for solving large-scale optimization problems. For large-scale problems, the methods save only a few n-dimensional vectors, instead of storing and computing fully dense $n \times n$ approximations of the Hessian. Since BFGS method is the most efficient method for solving unconstrained optimization problems, in this section we consider the limited memory BFGS method, known as L-BFGS, which is based on BFGS method.

As we know that the BFGS formula for inverse Hessian approximation H_k is

$$H_{k+1} = \left(I - \frac{s_k y_k^T}{s_k^T y_k}\right) H_k \left(I - \frac{y_k s_k^T}{s_k^T y_k}\right) + \frac{s_k s_k^T}{s_k^T y_k}. \tag{5.7.1}$$

Set

$$\rho_k = \frac{1}{s_k^T y_k}, \quad V_k = I - \rho y_k s_k^T, \tag{5.7.2}$$

then

$$H_{k+1} = V_k^T H_k V_k + \rho_k s_k s_k^T. \tag{5.7.3}$$

The above equation says that the matrix H_{k+1} is obtained by updating H_k using the pair $\{s_k, y_k\}$. In L-BFGS method we save implicitly a modified version of H_k by storing m pairs $\{s_i, y_i\}(i = k-m, k-m+1, \cdots, k-1)$.

In the following, we describe the expression of the updating matrix H_k of the k-th iteration in L-BFGS method.

Choose some initial Hessian approximation $H_k^{(0)}$ for the k-th iteration. We apply the formula (5.7.3) m times repeatedly, i.e.,

$$H_k^{(j+1)} = V_{k-m+j}^T H_k^{(j)} V_{k-m+j} + \rho_{k-m+j} s_{k-m+j} s_{k-m+j}^T,\ j = 0, 1, \cdots, m-1, \tag{5.7.4}$$

and obtain

$$\begin{aligned} H_k &= (V_{k-1}^T \cdots V_{k-m}^T) H_k^{(0)} (V_{k-m} V_{k-m+1} \cdots V_{k-1}) \\ &\quad + \rho_{k-m} (V_{k-1}^T \cdots V_{k-m+1}^T) s_{k-m} s_{k-m}^T (V_{k-m+1} \cdots V_{k-1}) \\ &\quad + \rho_{k-m+1} (V_{k-1}^T \cdots V_{k-m+2}^T) s_{k-m+1} s_{k-m+1}^T (V_{k-m+2} \cdots V_{k-1}) \\ &\quad + \cdots \\ &\quad + \rho_{k-1} s_{k-1} s_{k-1}^T. \end{aligned} \tag{5.7.5}$$

It follows from the above expression that if we know pairs $\{s_i, y_i\}(i = k-m, k-m+1, \cdots, k-1)$, we can compute H_k. In fact, we need not compute and save H_k explicitly, instead, we only save the pairs $\{s_i, y_i\}$ and compute $H_k g_k$, where g_k is the gradient of f at x_k. So, we have

$$\begin{aligned} H_k g_k &= (V_{k-1}^T \cdots V_{k-m}^T) H_k^{(0)} (V_{k-m} V_{k-m+1} \cdots V_{k-1}) g_k \\ &\quad + \rho_{k-m} (V_{k-1}^T \cdots V_{k-m+1}^T) s_{k-m} s_{k-m}^T (V_{k-m+1} \cdots V_{k-1}) g_k \\ &\quad + \rho_{k-m+1} (V_{k-1}^T \cdots V_{k-m+2}^T) s_{k-m+1} s_{k-m+1}^T (V_{k-m+2} \cdots V_{k-1}) g_k \\ &\quad + \cdots \\ &\quad + \rho_{k-1} s_{k-1} s_{k-1}^T g_k. \end{aligned} \tag{5.7.6}$$

Since

$$V_i g_k = (I - \rho_i y_i s_i^T) g_k,\ i = k-1, k-2, \cdots, k-m,$$

we have the following algorithm to compute $H_k g_k$.

Algorithm 5.7.1 *(L-BFGS two-loop recursion for $H_k g_k$)*

Step 1. $q := g_k$;

Step 2. **for** $i = k-1, k-2, \cdots, k-m$
$\alpha_i := \rho_i s_i^T q$;
$q := q - \alpha_i y_i$;
end (for)

Step 3. $r := H_k^{(0)} q$;

Step 4. **for** $i = k-m, k-m+1, \cdots, k-1$
$\beta := \rho_i y_i^T r$;
$r := r + s_i(\alpha_i - \beta)$
end (for) □

By use of the above algorithm, we obtain $r = H_k g_k$. A choice of $H_k^{(0)}$ is

$$H_k^{(0)} = \frac{s_k^T y_k}{\|y_k\|^2} I. \tag{5.7.7}$$

The limited memory BFGS algorithm can be stated as follows.

Algorithm 5.7.2 *(L-BFGS Method)*

Step 1. Given a starting point $x_0 \in R^n$, an initial symmetric and positive definite matrix $H_0 \in R^{n\times n}$, a nonnegative integer $m \geq 0$, an error tolerance $\epsilon > 0$, $k := 0$.

Step 2. Compute $g_k = \nabla f(x_k)$. If $\|g_k\| \leq \epsilon$, we take $x^* = x_k$, stop; otherwise, compute $d_k = -H_k g_k$ from Algorithm 5.7.1.

Step 3. Find a step size $\alpha_k > 0$ by using Wolfe rule.

Step 4. Set $x_{k+1} = x_k + \alpha_k d_k$.

Step 5. If $k > m$, discard the vector pairs $\{s_{k-m}, y_{k-m}\}$ from storage;
Set $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$;
Take $H_k^{(0)} = \frac{s_k^T y_k}{\|y_k\|^2} I$.

Step 6. $k := k+1$ and go to Step 2. □

The above L-BFGS algorithm is equivalent to the usual BFGS algorithm if the initial matrix H_0 is the same in both algorithms, and if $H_k^{(0)} = H_0$ at each iteration. Normally, for large-scale problems, we take $m \ll n$. In practice, the choice of m is dependent on the dimension of the problem and the storage of employed computer. Usually, we take $3 \leq m \leq 30$.

In the following, we establish the convergence and convergence rate of L-BFGS method.

Lemma 5.7.3 *Let $f(x)$ be a twice continuously differentiable and uniformly convex function, i.e., there exist $0 < m \leq M$ such that*

$$m\|u\|^2 \leq u^T G(x)u \leq M\|u\|^2, \ \forall x \in L(x_0), \ u \in R^n, \tag{5.7.8}$$

where $G(x) = \nabla^2 f(x)$ and $L(x_0) = \{x \mid f(x) \leq f(x_0)\}$. Then

$$\frac{\|y\|}{\|s_k\|} \leq M, \ \ \frac{\|s_k\|^2}{s_k^T y_k} \leq \frac{1}{m}, \ \ \frac{\|y_k\|^2}{s_k^T y_k} \leq M. \tag{5.7.9}$$

Proof. 1) Let $\bar{G} = \int_0^1 G(x_k + \tau s_k)\mathrm{d}\tau$. Then

$$y_k = g_{k+1} - g_k = \int_0^1 G(x_k + \tau s_k)s_k \mathrm{d}\tau = \bar{G}s_k. \tag{5.7.10}$$

Taking the norm, then we obtain

$$\|y_k\| \leq \|s_k\| \int_0^1 \|G(x_k + \tau s_k)\|\mathrm{d}\tau. \tag{5.7.11}$$

From the assumptions, it follows that $L(x_0)$ is a bounded, closed and convex set, then $x_k + \tau s_k \in L(x_0)$. Then $\|G(x_k + \tau s_k)\| \leq M$. Thus we have that $\|y_k\| \leq M\|s_k\|$ which is the first conclusion in (5.7.9).

2) By use of (5.7.10), we have that

$$s_k^T y_k = \int_0^1 s_k^T G(x_k + \tau s_k)s_k \mathrm{d}\tau \geq m\|s_k\|^2, \tag{5.7.12}$$

which means

$$\frac{\|s_k\|^2}{s_k^T y_k} \leq \frac{1}{m}.$$

3) Since $y_k = \bar{G}s_k$, then

$$\begin{aligned}\frac{\|y_k\|^2}{s_k^T y_k} &= \frac{y_k^T y_k}{s_k^T y_k} = \frac{s_k^T \bar{G}^{\frac{1}{2}} \bar{G} \bar{G}^{\frac{1}{2}} s_k}{s_k^T \bar{G}^{\frac{1}{2}} \bar{G}^{\frac{1}{2}} s_k} \\ &= \frac{\gamma_k^T \bar{G} \gamma_k}{\gamma_k^T \gamma_k} \leq M, \end{aligned} \tag{5.7.13}$$

where $\gamma_k = \bar{G}^{\frac{1}{2}} s_k$. □

Theorem 5.7.4 *Let $f(x)$ be a twice continuously differentiable and uniformly convex function. Then the iterative sequence $\{x_k\}$ generated from L-BFGS Algorithm 5.7.2 converges to the unique minimizer x^* of $f(x)$.*

Proof. From Lemma 5.7.3, we have

$$\frac{\|s_k\|^2}{s_k^T y_k} \leq M, \ \frac{\|y_k\|^2}{s_k^T y_k} \leq M. \tag{5.7.14}$$

Then

$$\|V_k\| \leq 1 + M. \tag{5.7.15}$$

Let $\bar{m} = \min\{k, m\}$. Without loss of generality, we assume that $\|H_k^{(0)}\| \leq M$. Then by (5.7.5) and (5.7.14)-(5.7.15) we get

$$\begin{aligned}\|H_k\| &\leq M(1+M)^{2\bar{m}} + \sum_{j=1}^{\bar{m}} M(1+M)^{2(\bar{m}-j)} \\ &\leq M(1+M)^{2\bar{m}}(\bar{m}+1). \end{aligned} \tag{5.7.16}$$

On the other hand, write $B_k^{(0)} = (H_k^{(0)})^{-1}$. From (5.7.4) we have

$$B_k^{(j+1)} = B_k^{(j)} - \frac{B_k^{(j)} s_{k-\bar{m}+j} s_{k-\bar{m}+j}^T (B_k^{(j)})^T}{s_{k-\bar{m}+j}^T B_k^{(j)} s_{k-\bar{m}+j}} + \frac{y_{k-\bar{m}+j} y_{k-\bar{m}+j}^T}{y_{k-\bar{m}+j}^T s_{k-\bar{m}+j}}, \ j = 0, 1, \bar{m}-1.$$

Then

$$B_k^{(\bar{m})} = B_k = H_k^{-1}.$$

Since $\mathrm{Tr}(xy^T) = x^T y$ for $x, y \in R^n$ and $\mathrm{Tr}(A+B) = \mathrm{Tr}(A) + \mathrm{Tr}(B)$ for $n \times n$ matrices A and B, then it follows from (5.7.14) that

$$\begin{aligned}\mathrm{Tr}(B_k^{(j+1)}) &= \mathrm{Tr}(B_k^{(j)}) - \frac{\|B_k^{(j)} s_{k-\bar{m}+j}\|^2}{s_{k-\bar{m}+j}^T B_k^{(j)} s_{k-\bar{m}+j}} + \frac{\|y_{k-\bar{m}+j}\|^2}{y_{k-\bar{m}+j}^T s_{k-\bar{m}+j}} \\ &\leq \mathrm{Tr}(B_k^{(j)}) + M. \end{aligned} \tag{5.7.17}$$

Repeatedly applying (5.7.17) $\bar{m}$ times, and using (5.7.7) and (5.7.14), we obtain that

$$\begin{aligned} \mathrm{Tr}(B_k) &= \mathrm{Tr}(B_k^{(\bar{m})}) \leq \mathrm{Tr}(B_k^{(0)}) + \bar{m}M \\ &= \mathrm{Tr}((H_k^{(0)})^{-1}) + \bar{m}M \\ &\leq (n+\bar{m})M. \end{aligned} \tag{5.7.18}$$

Let the eigenvalues of B_k be $0 < \lambda_1 \leq \cdots \leq \lambda_n$, then the eigenvalues of H_k are

$$0 < \frac{1}{\lambda_n} \leq \frac{1}{\lambda_{n-1}} \leq \cdots \leq \frac{1}{\lambda_1}.$$

By use of the property of the Rayleigh quotient and $\mathrm{Tr}(B_k) = \sum_{j=1}^{n} \lambda_j$, we obtain

$$\begin{aligned} \cos\theta_k &= \frac{-d_k^T g_k}{\|d_k\|\|g_k\|} = \frac{g_k^T H_k g_k}{\|H_k g_k\|\|g_k\|} \\ &\geq \frac{\|g_k\|^2/\lambda_n}{\|H_k\|\|g_k\|^2} = \frac{1}{\lambda_n \|H_k\|} \\ &\geq \frac{1}{\mathrm{Tr}(B_k)\|H_k\|}. \end{aligned} \tag{5.7.19}$$

Then, it follows from (5.7.16) and (5.7.18) that there is a $\rho > 0$ such that

$$\cos\theta_k \geq \rho \tag{5.7.20}$$

holds for all k. This implies that there is $\bar{\mu} > 0$ such that

$$\theta_k \leq \frac{\pi}{2} - \bar{\mu}, \ \forall k. \tag{5.7.21}$$

The assumptions of the theorem and Theorem 1.3.19 indicate that the level $L(x_0)$ is bounded, closed and convex. Then, the continuous function $\nabla f(x)$ exists and is uniformly continuous on $L(x_0)$. Noting that α_k is determined by Wolfe rule, then we obtain, by Theorem 2.5.5 and (5.7.21), that the sequence $\{x_k\}$ converges to the unique minimizer x^* of $f(x)$. □

Next, we establish the convergence rate of L-BFGS method.

Lemma 5.7.5 *Let $f(x)$ be a twice continuously differentiable and uniformly convex function. Then*

$$f(x) - f(x^*) \leq \frac{1}{m}\|g(x)\|^2. \tag{5.7.22}$$

Proof. Since $f(x)$ is a convex function, for any $x \in R^n$ we have

$$f(x) - f(x^*) \leq g(x)^T(x - x^*) \leq \|g(x)\|\|x - x^*\|. \tag{5.7.23}$$

Note that

$$g(x) = g(x) - g(x^*) = \int_0^1 G(x^* + \tau(x - x^*))(x - x^*)\mathrm{d}\tau. \tag{5.7.24}$$

Writing $\bar{G} = \int_0^1 G(x^* + \tau(x - x^*))\mathrm{d}\tau$, then we have

$$g(x) = \bar{G}(x - x^*). \tag{5.7.25}$$

By use of (5.7.8), we get

$$\begin{aligned} m\|x - x^*\|^2 &\leq (x - x^*)^T\bar{G}(x - x^*) \leq (x - x^*)^T g(x) \\ &\leq \|x - x^*\|\|g(x)\|, \end{aligned} \tag{5.7.26}$$

that is

$$\|x - x^*\| \leq \|g(x)\|/m. \tag{5.7.27}$$

Substituting (5.7.27) into (5.7.23) gives (5.7.22). □

Lemma 5.7.6 *Let $f(x)$ be a twice continuously differentiable and uniformly convex function. Let $x_{k+1} = x_k + \alpha_k d_k$, where α_k is determined by Wolfe rule. Then*

$$c_1\|g_k\|\cos\theta_k \leq \|s_k\| \leq c_2\|g_k\|\cos\theta_k \tag{5.7.28}$$

and

$$f(x_{k+1}) - f(x^*) \leq (1 - \rho m c_1 \cos^2\theta_k)[f(x_k) - f(x^*)], \tag{5.7.29}$$

where $c_1 = (1 - \sigma)/M, c_2 = 2(1 - \rho)/m$, ρ and σ are defined by Wolfe rule, θ_k is an angle between d_k and $-g_k$.

Proof. From (5.7.9) we have that

$$\frac{y_k^T s_k}{\|s_k\|^2} \leq \frac{\|y_k\|\|s_k\|}{\|s_k\|^2} = \frac{\|y_k\|}{\|s_k\|} \leq M. \tag{5.7.30}$$

By using Wolfe rule, we have that

$$y_k^T s_k = g_{k+1}^T s_k - g_k^T s_k \geq -(1 - \sigma)g_k^T s_k. \tag{5.7.31}$$

Then the above expressions give

$$\|s_k\|^2 \geq \frac{y_k^T s_k}{M} \geq -\frac{1-\sigma}{M} g_k^T s_k = \frac{1-\sigma}{M}\|g_k\|\|s_k\|\cos\theta_k,$$

that is

$$\|s_k\| \geq \frac{1-\sigma}{M}\|g_k\|\cos\theta_k. \tag{5.7.32}$$

We obtain the left-hand side inequality of (5.7.28).

By Taylor expression and Wolfe rule, we have that

$$g_k^T s_k + \frac{1}{2} s_k^T G(\xi_k) s_k = f(x_{k+1}) - f(x_k) \leq \rho g_k^T s_k, \tag{5.7.33}$$

where ξ_k lies between x_k and x_{k+1}. Then

$$s_k^T G(\xi_k) s_k \leq -2(1-\rho) g_k^T s_k. \tag{5.7.34}$$

Since $L(x_0)$ is a bounded, closed and convex set, $\xi_k \in L(x_0)$. Then we have that

$$m\|s_k\|^2 \leq s_k^T G(\xi_k) s_k. \tag{5.7.35}$$

The inequalities (5.7.34) and (5.7.35) yield

$$\|s_k\| \leq \frac{2(1-\rho)}{M}\|g_k\|\cos\theta_k,$$

which is the right-hand side of (5.7.28).

Finally, we prove (5.7.29). By using Wolfe rule and (5.7.28), we have that

$$\begin{aligned} f(x_{k+1}) - f(x_k) &\leq \rho g_k^T s_k = -\rho\|g_k\|\|s_k\|\cos\theta_k \\ &\leq -\rho c_1 \|g_k\|^2 \cos^2\theta_k. \end{aligned} \tag{5.7.36}$$

From Lemma 5.7.5, we have

$$\|g_k\|^2 \geq m[f(x_k) - f(x^*)]. \tag{5.7.37}$$

So, we can substitute (5.7.37) into (5.7.36) to obtain that

$$f(x_{k+1}) - f(x_k) \leq -\rho m c_1 \cos^2\theta_k [f(x_k) - f(x^*)], \tag{5.7.38}$$

which gives result (5.7.29) by subtracting $f(x^*)$ from both sides. □

Theorem 5.7.7 *Let $f(x)$ be a twice continuously differentiable and uniformly convex function. Assume that the iterative sequence $\{x_k\}$ generated by L-BFGS Algorithm 5.7.2 converges to the unique minimizer x^* of $f(x)$. Then the rate of convergence is at least R-linear.*

Proof. From (5.7.29) we have

$$f(x_{k+1}) - f(x^*) \leq \delta(f(x_k) - f(x^*)),$$

where $\delta \in (0, 1)$. Also, since $f(x)$ is a uniformly convex function, there are $0 < m_1 \leq M_1$ such that

$$m_1\|u\|^2 \leq u^T G(x)u \leq M_1\|u\|^2, \ \forall x \in L(x_0), \ u \in R^n. \tag{5.7.39}$$

By using Taylor expression of $f(x_k)$ at x^* and (5.7.39), we obtain that

$$f(x_k) - f(x^*) \geq \frac{m_1}{2}\|x_k - x^*\|^2. \tag{5.7.40}$$

Hence

$$\begin{aligned} \|x_k - x^*\| &\leq \sqrt{\frac{2}{m_1}}(f(x_k) - f(x^*))^{\frac{1}{2}} \\ &\leq \sqrt{\frac{2}{m_1}}\delta^{\frac{1}{2}}(f(x_{k-1}) - f(x^*))^{\frac{1}{2}} \\ &\leq \cdots \\ &\leq \sqrt{\frac{2}{m_1}}(\delta^{\frac{1}{2}})^k(f(x_0) - f(x^*))^{\frac{1}{2}}. \end{aligned} \tag{5.7.41}$$

The above inequality shows that the sequence $\{x_k\}$ is R-linearly convergent. □

This theorem indicates that L-BFGS method often converges slowly, which leads to a relatively large number of function evaluations. Also, it is inefficient on highly ill-conditioned optimization problems. Though there are some weaknesses, L-BFGS method is a main choice for large-scale problems in which the true Hessian is not sparse, because, in this case, it may outperform other rival algorithms. For further details of L-BFGS method, please consult Liu and Nocedal [200] and Nash and Nocedal [228].

At the end of this section, we mension a memoryless BFGS formula. For BFGS formula (5.7.1) and (5.7.3), if we set $H_k = I$ at each iteration, we have

$$H_{k+1} = V_k^T V_k + \rho_k s_k s_k^T \tag{5.7.42}$$

$$= \left(I - \frac{s_k y_k^T}{s_k^T y_k}\right)\left(I - \frac{y_k s_k^T}{s_k^T y_k}\right) + \frac{s_k s_k^T}{s_k^T y_k}. \tag{5.7.43}$$

The above formula satisfies quasi-Newton condition and positive definiteness, and is called the memoryless BFGS formula. Obviously, if $m = 1$ and $H_k^{(0)} = I, \forall k$, the limited memory BFGS method is just the memoryless BFGS method.

Exercises

1. Using DFP method minimize the Rosenbrock function in Appendix 1.1 and the Extended Rosenbrock function in Appendix 1.2.

2. Using BFGS method minimize the Extended Rosenbrock function in Appendix 1.2 and the Powell singular function in Appendix 1.4.

3. State the properties of DFP and BFGS formulas and their relations.

4. Prove that if f is strong convex, $y_k^T s_k > 0$ holds.

5. Prove that H_{k+1}^{BFGS} given by (5.1.49) is the unique solution of problem (5.1.79).

6. Prove Theorem 5.2.1.

7. State the properties of Broyden class and Huang class, and their relations.

8. Prove Theorem 5.4.3.

9. Describe the motivation of self-scaling strategy in variable metric methods by observing DFP method.

10. Do programming of L-BFGS algorithm in §5.9 in MATLAB or FORTRAN.

Chapter 6

Trust-Region Methods and Conic Model Methods

6.1 Trust-Region Methods

6.1.1 Trust-Region Methods

The basic idea of Newton's method is to approximate the objective function $f(x)$ around x_k by choosing a quadratic model of the form

$$q^{(k)}(s) = f(x_k) + g_k^T s + \frac{1}{2} s^T G_k s,$$

where $g_k = \nabla f(x_k)$ and $G_k = \nabla^2 f(x_k)$, and use the minimizer s_k of $q^{(k)}(s)$ to modify x_k,

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

However, this method can only guarantee the local convergence, i.e., when s is small enough, the method is convergent locally. In Chapter 2, we have introduced line search approaches which guarantee the method is convergent globally. Line search approaches use the quadratic model to generate a search direction and then find a suitable stepsize α along the direction. Although it is successful at most time, it does not use sufficiently the n-dimensional quadratic model. The other disadvantage is that the Newton's method cannot be used if the Hessian matrices are not positive definite.

In this section the other class of global approaches is introduced, which is called the trust-region method. It not only replaces line search to get

the global convergence, but also circumvents the difficulty caused by non-positive definite Hessian matrices in line search. Besides, it produces more significant reduction in objective value f than line search approaches. In the trust-region method, we first define a region around the current iterate

$$\Omega_k = \{x \ : \ \|x - x_k\| \leq \Delta_k\},$$

where Δ_k is the radius of Ω_k, in which the model is trusted to be adequate to the objective function. And then we choose a step to be the approximate minimizer of the quadratic model in the trust-region, i.e., such that $x_k + s_k$ is the approximately best point on the generalized sphere

$$\{x_k + s \mid \|s\| \leq \Delta_k\}$$

with center x_k and radius Δ_k. If the step is not acceptable, we reduce the size of the trust-region and find a new minimizer. This method retains the rapid local convergence rate of Newton's method and quasi-Newton method, but also has ideal global convergence. Since the step is restricted by the trust-region, it is also called the restricted step method. The model subproblem of the trust-region method is

$$\begin{aligned} \min \quad & q^{(k)}(s) = f(x_k) + g_k^T s + \frac{1}{2} s^T B_k s \\ \text{s.t.} \quad & \|s\| \leq \Delta_k, \end{aligned} \tag{6.1.1}$$

where $\Delta_k > 0$ is the trust-region radius, B_k is symmetric and approximate to the Hessian G_k. Normally, we use l_2 norm $\|\cdot\|_2$ so that s_k is the minimizer of $q^{(k)}(s)$ in the ball of radius Δ_k. Other norms can also be used, however, the different norms define the different shapes of the trust-region. In (6.1.1), if we set $B_k = G_k$, the method is said to be a Newton-type trust-region method.

How to choose Δ_k at each iteration? In general, when there is good agreement between the model $q^{(k)}(s)$ and the objective function value $f(x_k + s)$, one should select Δ_k as large as possible. Let

$$Ared_k = f(x_k) - f(x_k + s_k) \tag{6.1.2}$$

which is called the *actual reduction*, and let

$$Pred_k = q^{(k)}(0) - q^{(k)}(s_k) \tag{6.1.3}$$

which is called the *predicted reduction*. Define the ratio

$$r_k = \frac{Ared_k}{Pred_k}, \tag{6.1.4}$$

which measures the agreement between the model function $q^{(k)}$ and the objective function f. This ratio r_k plays an important role in selecting new iterate x_{k+1} and updating the trust-region radius Δ_k. If r_k is close to 1, it means there is good agreement, and we can expand the trust-region for the next iteration; if r_k is close to zero or negative, we shrink the trust-region; otherwise, we do not alter the trust-region. The following is the trust-region algorithm.

Algorithm 6.1.1 *(Trust-Region Algorithm)*

Step 1. *Given initial point* $x_0, \bar{\Delta}, \Delta_0 \in (0, \bar{\Delta}), \epsilon \geq 0, 0 < \eta_1 \leq \eta_2 < 1$ *and* $0 < \gamma_1 < 1 < \gamma_2$, $k := 0$.

Step 2. *If* $\|g_k\| \leq \epsilon$, *stop.*

Step 3. *Approximately solve the subproblem (6.1.1) for* s_k.

Step 4. *Compute* $f(x_k + s_k)$ *and* r_k. *Set*

$$x_{k+1} = \begin{cases} x_k + s_k, & \text{if } r_k \geq \eta_1, \\ x_k, & \text{otherwise.} \end{cases}$$

Step 5. *If* $r_k < \eta_1$, *then* $\Delta_{k+1} \in (0, \gamma_1 \Delta_k]$;
If $r_k \in [\eta_1, \eta_2)$, *then* $\Delta_{k+1} \in [\gamma_1 \Delta_k, \Delta_k]$;
If $r_k \geq \eta_2$ *and* $\|s_k\| = \Delta_k$, *then* $\Delta_{k+1} \in [\Delta_k, \min\{\gamma_2 \Delta_k, \bar{\Delta}\}]$.

Step 6. *Generate* B_{k+1}, *update* $q^{(k)}$, *set* $k := k + 1$, *go to Step 2.*
□

In the above algorithm, $\bar{\Delta}$ is an overall bound for all Δ_k. The iterations for which $r_k \geq \eta_2$ and thus for which $\Delta_{k+1} \geq \Delta_k$, are said to be very successful iterations; the iterations for which $r_k \geq \eta_1$ and thus for which $x_{k+1} = x_k + s_k$, are said to be successful iterations; otherwise the iterations for which $r_k < \eta_1$ and thus for which $x_{k+1} = x_k$, are said to be unsuccessful iterations. Sometimes, the iterations in the first two cases are said to be successful iterations.

We like to point out some choices of the parameters, for instance, $\eta_1 = 0.01, \eta_2 = 0.75, \gamma_1 = 0.5, \gamma_2 = 2, \Delta_0 = 1$ or $\Delta_0 = \frac{1}{10}\|g_0\|$. However, the algorithm is insensitive to their change. In addition, Δ_{k+1} can be selected by polynomial interpolation. For example, if $r_k < 0.01$, then Δ_{k+1} can be chosen in an interval $(0.01, 0.5)\|s_k\|$ on the basis of a polynomial interpolation. Also, if we use quadratic interpolation, we have

$$\lambda = \frac{-g_k^T s_k}{2[f(x_k + s_k) - f(x_k) - g_k^T s_k]}, \tag{6.1.5}$$

and we set

$$\Delta_{k+1} = \lambda \|s_k\|. \tag{6.1.6}$$

Finally, to conclude this subsection, we give the characterization of the solution of subproblem (6.1.1). For convenience, we drop the subscripts in the following theorem.

Theorem 6.1.2 *The vector s^* is the solution of the subproblem*

$$\min \quad f + g^T s + \frac{1}{2} s^T B s \tag{6.1.7}$$

$$\text{s.t.} \quad \|s\|_2 \leq \Delta, \tag{6.1.8}$$

if and only if there is a scalar $\lambda^* \geq 0$ such that

$$(B + \lambda^* I)s^* = -g, \tag{6.1.9}$$

$$\|s^*\|_2 \leq \Delta, \tag{6.1.10}$$

$$\lambda^*(\Delta - \|s^*\|_2) = 0, \tag{6.1.11}$$

and $(B + \lambda^* I)$ is positive semidefinite.

Proof. Let s^* be the solution of subproblem (6.1.7)-(6.1.8). From the optimality condition of constrained optimization (see Chapter 8), there exists a multiplier $\lambda^* \geq 0$ such that (6.1.9)-(6.1.11) hold. We now need to prove that the matrix $(B + \lambda^* I)$ is positive semidefinite.

If $\|s^*\|_2 < \Delta$, then $\lambda^* = 0$ and s^* is an unconstrained minimizer of q, and thus B is positive semidefinite and furthermore $(B + \lambda^* I)$ is positive semidefinite.

If $\|s^*\|_2 = \Delta$, it follows from the second-order necessary condition (see §8.3) that

$$s^T(B + \lambda^* I)s \geq 0 \tag{6.1.12}$$

for all s satisfying $s^T s^* = 0$. If $s^T s^* \neq 0$, take $t = -2s^T s^*/\|s\|_2^2$, then $\|s^* + ts\|_2 = \Delta$. By the definition of s^*, we have

$$q(s^* + ts) + \frac{1}{2}\lambda^* \|s^* + ts\|_2^2 \geq q(s^*) + \frac{1}{2}\lambda^* \|s^*\|_2^2. \tag{6.1.13}$$

Developing $q(\cdot)$ yields

$$\frac{1}{2}t^2 s^T (B + \lambda^* I)s \geq -t(s^T[g + (B + \lambda^* I)s^*]). \tag{6.1.14}$$

By using (6.1.9) we get that the right-hand side of (6.1.14) is equal to zero. Then the above inequality indicates that

$$s^T (B + \lambda^* I)s \geq 0$$

for all s with $s^T s^* \neq 0$. Therefore, $B + \lambda^* I$ is positive semidefinite.

Conversely, assume that there is $\lambda^* \geq 0$ such that s^* satisfies (6.1.9)-(6.1.11) and that $B + \lambda^* I$ is positive semidefinite. Then, for all s satisfying $\|s\|_2 \leq \Delta$, we have

$$\begin{aligned} q(s) &= f(x) + g^T s + \frac{1}{2}s^T (B + \lambda^* I)s - \frac{1}{2}\lambda^* \|s\|_2^2 \\ &\geq f(x) + g^T s^* + \frac{1}{2}(s^*)^T (B + \lambda^* I)s^* - \frac{1}{2}\lambda^* \|s\|_2^2 \\ &= q(s^*) + \frac{1}{2}\lambda^* [\|s^*\|_2^2 - \|s\|_2^2]. \end{aligned}$$

By use of (6.1.11), we have that $\lambda^*(\Delta^2 - (s^*)^T s^*) = 0$. So, the above inequality becomes

$$\begin{aligned} q(s) &\geq q(s^*) + \frac{1}{2}\lambda^* [(\|s^*\|_2^2 - \Delta^2) + (\Delta^2 - \|s\|_2^2)] \\ &= q(s^*) + \frac{1}{2}\lambda^* [\Delta^2 - \|s\|^2]. \end{aligned}$$

Thus, from $\lambda^* \geq 0$ and $\|s\|_2 \leq \Delta$, we immediately have

$$q(s) \geq q(s^*),$$

which implies s^* is the solution of (6.1.7)-(6.1.8). □

If $(B + \lambda^* I)$ is singular, we refer to this case as the *hard case*. In this case, d^* has the form

$$d^* = -(B + \lambda^* I)^+ g + v, \tag{6.1.15}$$

where $(B + \lambda^* I)^+$ denotes the generalized inverse of $(B + \lambda^* I)$, and v is a vector in null space of $(B + \lambda^* I)$.

Assume that $(B + \lambda^* I)$ is positive definite, then d^* can be obtained by solving

$$\lambda[\Delta - \|(B + \lambda I)^{-1} g\|_2] = 0, \tag{6.1.16}$$

$$\|(B + \lambda I)^{-1} g\|_2 \leq \Delta, \quad \lambda \geq 0 \tag{6.1.17}$$

for λ^* and then setting

$$d^* = -(B + \lambda^* I)^{-1} g.$$

If B is positive definite and $\|B^{-1} g\|_2 < \Delta$, then $d^* = -B^{-1} g$ simply. Otherwise, $\lambda^* > 0$. We need to solve

$$\psi(\lambda) = \frac{1}{\|(B + \lambda I)^{-1} g\|_2} - \frac{1}{\Delta} = 0. \tag{6.1.18}$$

We consider solving $\psi(\lambda) = 0$, instead of solving $\Delta - \|(B + \lambda I)^{-1} g\|_2 = 0$, because $\psi(\lambda)$ is nearly linear in the considered range. By direct computation, we have

$$\psi'(\lambda) = \frac{g^T H(\lambda)^{-3} g}{\|H(\lambda)^{-1} g\|_2^3}, \tag{6.1.19}$$

$$\psi''(\lambda) = -\frac{3 g^T H(\lambda)^{-4} g}{\|H(\lambda)^{-1} g\|_2^3} [1 - \cos^2(\langle H(\lambda)^{-1} g, H(\lambda)^{-2} g \rangle)], \tag{6.1.20}$$

where $H(\lambda) = B + \lambda I$. Therefore, for the most negative eigenvalue $\lambda_1 < 0$, if $\lambda > -\lambda_1$, $\psi(\lambda)$ is strictly increasing and concave. So, Newton's method can be used to solve (6.1.18), that is,

$$\begin{aligned} \lambda_+ &= \lambda - \frac{\psi(\lambda)}{\psi'(\lambda)} \\ &= \lambda - \frac{1}{\frac{g^T (B+\lambda I)^{-3} g}{\|(B+\lambda I)^{-1} g\|_2^3}} \left[\frac{1}{\|(B + \lambda I)^{-1} g\|_2} - \frac{1}{\Delta} \right]. \end{aligned} \tag{6.1.21}$$

6.1.2 Convergence of Trust-Region Methods

In order to discuss the convergence of trust-region methods, we first give some assumptions and technical lemmas.

We assume that the approximate Hessians B_k are uniformly bounded in norm, and that the level set

$$\{x \mid f(x) \leq f(x_0)\} \tag{6.1.22}$$

is bounded, on which the function $f : R^n \to R$ is continuously differentiable. For generality, we also allow the length of the approximate solution s_k of the subproblem (6.1.1) to exceed the trust-region bound, provided that it stays within a fixed multiple of the bound, that is

$$\|s_k\| \leq \tilde{\eta}\Delta_k, \tag{6.1.23}$$

where $\tilde{\eta}$ is a positive constant. The above assumptions are said to be Assumption (A_0).

For trust-region algorithm, in general, we do not seek an accurate solution of subproblem (6.1.1) but we are content with a nearly optimal solution of (6.1.1). Strong theoretical and numerical results can be obtained if the step s_k produced by Algorithm 6.1.1 satisfies

$$q_k(0) - q_k(s_k) \geq \beta_1 \|g_k\|_2 \min\left\{\Delta_k, \frac{\|g_k\|_2}{\|B_k\|_2}\right\}, \tag{6.1.24}$$

where $\beta_1 \in (0, 1]$. Below, we show that the Cauchy point s_k^c satisfies (6.1.24) with $\beta_1 = \frac{1}{2}$ and that the exact solution s_k of the subproblem (6.1.1) satisfies (6.1.24) with $\beta_1 = \frac{1}{2}$. If s_k is an approximate solution of the subproblem (6.1.1) with $q^{(k)}(0) - q^{(k)}(s_k) \geq \beta_2(q^{(k)}(0) - q^{(k)}(s_k^c))$, then it satisfies (6.1.24) with $\beta_1 = \frac{1}{2}\beta_2$.

Lemma 6.1.3 *Let s_k be the solution of (6.1.1), let $\|\cdot\| = \|\cdot\|_2$, then*

$$\begin{aligned} Pred_k &= q^{(k)}(0) - q^{(k)}(s_k) \\ &\geq \frac{1}{2}\|g_k\|_2 \min\left\{\Delta_k, \frac{\|g_k\|_2}{\|B_k\|_2}\right\}. \end{aligned} \tag{6.1.25}$$

Proof. By the definition of s_k, for all $\alpha \in [0, 1]$, we have

$$\begin{aligned} q^{(k)}(0) - q^{(k)}(s_k) &\geq q^{(k)}(0) - q^{(k)}(-\alpha\frac{\Delta_k}{\|g_k\|_2}g_k) \\ &= \alpha\Delta_k\|g_k\|_2 - \frac{1}{2}\alpha^2\Delta_k^2 g_k^T B_k g_k/\|g_k\|_2^2 \\ &\geq \alpha\Delta_k\|g_k\|_2 - \frac{1}{2}\alpha^2\Delta_k^2\|B_k\|_2. \end{aligned} \tag{6.1.26}$$

Therefore we must have

$$\begin{aligned} Pred_k &\geq \max_{0\leq\alpha\leq 1}[\alpha\Delta_k\|g_k\|_2 - \frac{1}{2}\alpha^2\Delta_k^2\|B_k\|_2] \\ &\geq \frac{1}{2}\|g_k\|_2 \min\left\{\Delta_k, \frac{\|g_k\|_2}{\|B_k\|_2}\right\}. \quad \square \end{aligned} \tag{6.1.27}$$

The Cauchy point of the subproblem (6.1.1) can be defined by

$$q^{(k)}(s_k^c) = \min\{q^{(k)}(s) \mid s = \tau s_k^G,\ \|s\| \leq \Delta_k\}, \tag{6.1.28}$$

where s_k^G solves a linear version of subproblem (6.1.1):

$$\begin{aligned} \min \quad & f(x_k) + g_k^T s \\ \text{s.t.} \quad & \|s\| \leq \Delta_k. \end{aligned} \tag{6.1.29}$$

Obviously, the solution of (6.1.29) is

$$s_k^G = -\frac{\Delta_k}{\|g_k\|_2} g_k.$$

Therefore, the Cauchy point of the subproblem (6.1.1) can be expressed as

$$s_k^c = \tau_k s_k^G = -\tau_k \frac{\Delta_k}{\|g_k\|_2} g_k, \tag{6.1.30}$$

where

$$\tau_k = \begin{cases} 1 & \text{if } g_k^T B_k g_k \leq 0; \\ \min\{\|g_k\|_2^3/(\Delta_k g_k^T B_k g_k),\ 1\} & \text{otherwise.} \end{cases} \tag{6.1.31}$$

In fact, if $g_k^T B_k g_k \leq 0$, the function $q^{(k)}(s_k^c) = q^{(k)}(\tau s_k^G)$ decreases monotonically with τ when $g_k \neq 0$. Therefore, we can take τ as large as possible within $\|\tau s_k^G\| \leq \Delta_k$. In this case, by use of (6.1.30) and $\|\tau s_k^G\| \leq \Delta_k$, we have that $\tau_k = 1$. If $g_k^T B_k g_k > 0$, $q^{(k)}(\tau s_k^G)$ is a convex and quadratic function in τ. Then, by minimizing $q^{(k)}(\tau s_k^G)$, we obtain that τ_k equals $\|g_k\|_2^3/(\Delta_k g_k^T B_k g_k)$, or the boundary value 1.

Lemma 6.1.4 *The Cauchy point s_k^c satisfies*

$$q^{(k)}(0) - q^{(k)}(s_k^c) \geq \frac{1}{2}\|g_k\|_2 \min\left\{\Delta_k, \frac{\|g_k\|_2}{\|B_k\|_2}\right\}. \tag{6.1.32}$$

Proof. Consider first the case of $g_k^T B_k g_k \leq 0$. In this case, it follows from (6.1.31) that $\tau_k = 1$, and we have

$$\begin{aligned} q^{(k)}(0) - q^{(k)}(s_k^c) &= -q^{(k)}\left(-\frac{\Delta_k}{\|g_k\|_2} g_k\right) \\ &= \Delta_k \|g_k\|_2 - \frac{1}{2}\Delta_k^2 g_k^T B_k g_k / \|g_k\|_2^2 \\ &\geq \Delta_k \|g_k\|_2 \\ &\geq \|g_k\|_2 \min\left\{\Delta_k, \frac{\|g_k\|_2}{\|B_k\|_2}\right\}. \end{aligned} \tag{6.1.33}$$

Consider the case of $g_k^T B_k g_k > 0$ and

$$\frac{\|g_k\|^3}{\Delta_k g_k^T B_k g_k} \leq 1. \tag{6.1.34}$$

In this case, $\tau_k = \|g_k\|^3/(\Delta_k g_k^T B_k g_k)$, and we have

$$\begin{aligned} q^{(k)}(0) - q^{(k)}(s_k^c) &= \frac{\|g_k\|_2^4}{g_k^T B_k g_k} - \frac{1}{2} g_k^T B_k g_k \frac{\|g_k\|_2^4}{(g_k^T B_k g_k)^2} \\ &= \frac{1}{2}\frac{\|g_k\|_2^4}{g_k^T B_k g_k} \\ &\geq \frac{1}{2}\frac{\|g_k\|_2^2}{\|B_k\|_2} \\ &\geq \frac{1}{2}\|g_k\|_2 \min\left\{\Delta_k, \frac{\|g_k\|_2}{\|B_k\|_2}\right\}. \end{aligned} \tag{6.1.35}$$

Consider the case of $g_k^T B_k g_k > 0$ and

$$\frac{\|g_k\|_2^3}{\Delta_k g_k^T B_k g_k} > 1. \tag{6.1.36}$$

In this case, $\tau_k = 1$, and by use of (6.1.36) we have

$$\begin{aligned} q^{(k)}(0) - q^{(k)}(s_k^c) &= \Delta_k \|g_k\|_2 - \frac{1}{2}\Delta_k^2 g_k^T B_k g_k / \|g_k\|_2^2 \\ &\geq \Delta_k \|g_k\|_2 - \frac{1}{2}\frac{\Delta_k^2}{\|g_k\|_2^2}\frac{\|g_k\|_2^3}{\Delta_k} \\ &= \frac{1}{2}\Delta_k \|g_k\|_2 \\ &\geq \frac{1}{2}\|g_k\|_2 \min\left\{\Delta_k, \frac{\|g_k\|_2}{\|B_k\|_2}\right\}. \end{aligned} \tag{6.1.37}$$

The above discussion of three cases gives the result (6.1.32). □

Usually, we assume that s_k is an approximate solution of the subproblem (6.1.1) and satisfies

$$q^{(k)}(0) - q^{(k)}(s_k) \geq \beta_2(q^{(k)}(0) - q^{(k)}(s_k^e)), \tag{6.1.38}$$

where s_k^e is an exact solution of subproblem (6.1.1) and $\beta_2 \in (0, 1]$ is a constant. Since $q^{(k)}(s_k^e) \leq q^{(k)}(s_k^c)$, we immediately have

$$q^{(k)}(0) - q^{(k)}(s_k) \geq \beta_2(q^{(k)}(0) - q^{(k)}(s_k^c)), \tag{6.1.39}$$

where $s_k^c = -\tau_k \frac{\Delta_k}{\|g_k\|_2} g_k$ with $0 \leq \tau_k \leq 1$ is a Cauchy point. So, we immediately have

Lemma 6.1.5 *Let s_k be an approximate solution of (6.1.1) and satisfy (6.1.38) or (6.1.39). Then*

$$\begin{aligned} Pred_k &= q^{(k)}(0) - q^{(k)}(s_k) \\ &\geq \frac{1}{2}\beta_2 \|g_k\|_2 \min\left\{\Delta_k, \frac{\|g_k\|_2}{\|B_k\|_2}\right\}, \end{aligned} \tag{6.1.40}$$

where $\beta_2 \in (0, 1]$.

Next, in order to prove the global convergence theorem, we give some technical lemmas.

Lemma 6.1.6 *Let Assumption (A_0) hold. We have*

$$|f(x_k + s_k) - q^{(k)}(s_k)| \leq \frac{1}{2}M\|s_k\|^2 + C(\|s_k\|)\|s_k\|, \tag{6.1.41}$$

where $C(\|s_k\|)$ is arbitrarily small by restricting the size of s_k.

Proof. By Taylor's theorem,

$$f(x_k + s_k) = f(x_k) + g_k^T s_k + \int_0^1 [\nabla f(x_k + ts_k) - \nabla f(x_k)]^T s_k \mathrm{d}t.$$

Also,

$$q^{(k)}(s_k) = f(x_k) + g_k^T s_k + \frac{1}{2} s_k^T B_k s_k.$$

Then

$$\begin{aligned} |f(x_k + s_k) - q^{(k)}(s_k)| &= |\frac{1}{2} s_k^T B_k s_k - \int_0^1 [\nabla f(x_k + ts_k) - \nabla f(x_k)]^T s_k \mathrm{d}t| \\ &\leq \frac{1}{2}M\|s_k\|^2 + C(\|s_k\|)\|s_k\|. \quad \square \end{aligned}$$

Lemma 6.1.7 *Assume that Assumption* (A_0) *holds. Suppose that* $\|g_k\|_2 \geq \epsilon > 0$ *and that* Δ_k *is smaller than some threshold* $\tilde{\Delta}$*. Then the* k*-th iteration is a very successful iteration which satisfies* $\Delta_{k+1} \geq \Delta_k$*.*

Proof. By Lemma 6.1.5 and the assumptions,

$$\begin{aligned} Pred_k &= q^{(k)}(0) - q^{(k)}(s_k) \\ &\geq \frac{1}{2}\beta_2 \|g_k\|_2 \min\left\{\Delta_k, \frac{\|g_k\|_2}{\|B_k\|_2}\right\} \\ &\geq \frac{1}{2}\beta_2 \epsilon \min\left\{\Delta_k, \frac{\epsilon}{M}\right\}. \end{aligned} \tag{6.1.42}$$

From Algorithm 6.1.1, by use of (6.1.41), (6.1.42) and (6.1.23), we have

$$\begin{aligned} |r_k - 1| &= \left|\frac{(f(x_k) - f(x_k + s_k)) - (q^{(k)}(0) - q^{(k)}(s_k))}{q^{(k)}(0) - q^{(k)}(s_k)}\right| \\ &= \left|\frac{f(x_k + s_k) - q^{(k)}(s_k)}{q^{(k)}(0) - q^{(k)}(s_k)}\right| \\ &\leq \frac{\frac{1}{2}M\|s_k\|^2 + C(\|s_k\|)\|s_k\|}{\frac{1}{2}\beta_2\epsilon \min\{\Delta_k, \epsilon/M\}} \\ &\leq \frac{\tilde{\eta}\Delta_k(M\tilde{\eta}\Delta_k + 2C(\|s_k\|))}{\beta_2\epsilon \min\{\Delta_k, \epsilon/M\}}. \end{aligned} \tag{6.1.43}$$

Since Δ_k is smaller than some threshold $\tilde{\Delta}$, we may choose $\tilde{\Delta}$ to be small enough such that

$$\Delta_k \leq \tilde{\Delta} \leq \epsilon/M, \ M\tilde{\eta}\Delta_k + 2C(\|s_k\|) \leq (1 - \eta_2)\beta_2\epsilon/\tilde{\eta},$$

so we have $r_k \geq \eta_2$. It follows from Algorithm 6.1.1 that $\Delta_{k+1} \geq \Delta_k$. □

This lemma indicates that if the current iterate is not a first-order stationary point and the trust-region radius Δ_k is small enough, then we always have $\Delta_{k+1} \geq \Delta_k$ and the iteration is very successful. Now we are in a position to give the global convergence theorem.

First, we consider the case when there are only finitely many successful iterations.

Theorem 6.1.8 *Under Assumption* (A_0)*, if Algorithm 6.1.1 has finitely many successful iterations, then the algorithm converges to the first-order stationary point.*

Proof. Since the algorithm has only finitely many successful iterations, then, for sufficiently large k, the iteration is unsuccessful. Thus, the sequence $\{\Delta_k\}$ from the algorithm converges to zero.

Suppose that k_0 is the index of the last successful iteration. If $\|g_{k_0+1}\| > 0$, it follows from Lemma 6.1.7 that there must be a very successful iteration of index larger than k_0, which satisfies $\Delta_{k_0+1+1} > \Delta_{k_0+1}$. This is a contradiction to the assumption. The contradiction proves our theorem. □

Next, we only need to restrict our attention to the case where there are infinitely many successful iterations.

Theorem 6.1.9 *Let Assumption (A_0) hold. If Algorithm 6.1.1 has infinitely many successful iterations, then the sequence of Algorithm 6.1.1 satisfies*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{6.1.44}$$

Proof. Assume, by contradiction, that there is $\epsilon > 0$ and a positive index K such that

$$\|g_k\| \geq \epsilon \text{ for all } k \geq K.$$

From Algorithm 6.1.1 and Lemma 6.1.5, it follows for successful iterations that

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq \eta_1[q^{(k)}(0) - q^{(k)}(s_k)] \\ &\geq \frac{1}{2}\eta_1\beta_2\|g_k\|_2 \min\left[\Delta_k, \frac{\|g_k\|_2}{\|B_k\|_2}\right] \\ &\geq \frac{1}{2}\eta_1\beta_2\epsilon \min\left[\Delta_k, \frac{\epsilon}{\beta}\right], \end{aligned} \tag{6.1.45}$$

where $\beta = \max\{1 + \|B_k\|_2\}$ is an upper bound of the Hessian approximation. So,

$$\begin{aligned} f(x_0) - f(x_{k+1}) &= \sum_{j=0, j\in\mathcal{S}} [f(x_j) - f(x_{j+1})] \\ &\geq \frac{1}{2}\sigma_k\eta_1\beta_2\epsilon \min\left[\Delta_k, \frac{\epsilon}{\beta}\right], \end{aligned}$$

where σ_k is a number of successful iterations till the k-th iteration with

$$\lim_{k\to\infty} \sigma_k = +\infty,$$

and $\mathcal{S}$ is an index set of successful iterations.

Since f is bounded below, it follows from the above inequality that

$$\lim_{k\to\infty} \Delta_k = 0, \tag{6.1.46}$$

contradicting the conclusion of Lemma 6.1.7. □

Now we give a stronger result on the convergence which is for all limit points.

Theorem 6.1.10 *Suppose that Assumption* (A_0) *holds. Then*

$$\lim_{k\to\infty} g_k = 0. \tag{6.1.47}$$

Proof. Assume, by contradiction, that the conclusion does not hold, then there is a subsequence of successful iterations such that

$$\|g_{t_i}\| \geq 2\epsilon > 0 \tag{6.1.48}$$

for some $\epsilon > 0$ and for all i.

Theorem 6.1.9 guarantees that, for each i, there exists a first successful iteration $l(t_i) > t_i$ such that $\|g_{l(t_i)}\| < \epsilon$. We denote $l_i \stackrel{\Delta}{=} l(t_i)$. Thus, there exists another subsequence $\{l_i\}$ such that

$$\|g_k\| \geq \epsilon \quad \text{for} \quad t_i \leq k < l_i \quad \text{and} \quad \|g_{l_i}\| < \epsilon. \tag{6.1.49}$$

Since

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq \eta_1[q^{(k)}(0) - q^{(k)}(s_k)] \\ &\geq \frac{1}{2}\eta_1\beta_2\epsilon \min[\Delta_k, \epsilon/\beta], \end{aligned} \tag{6.1.50}$$

it follows from the monotonically decreasing and the bounded below of the sequence $\{f(x_k)\}$ that

$$\lim_{k\to\infty} \Delta_k = 0. \tag{6.1.51}$$

Then

$$\Delta_k \leq \frac{2}{\eta_1\beta_2\epsilon}[f(x_k) - f(x_{k+1})] \tag{6.1.52}$$

which implies that for i sufficiently large,

$$\begin{aligned} \|x_{t_i} - x_{l_i}\| &\leq \sum_{j=t_i}^{l_i-1} \|x_j - x_{j+1}\| \leq \sum_{j=t_i}^{l_i-1} \Delta_j \\ &\leq \frac{2}{\eta_1\beta_2\epsilon}[f(x_{t_i}) - f(x_{l_i})]. \end{aligned} \tag{6.1.53}$$

From the fact that the right-hand side converges to zero, we get

$$\|x_{t_i} - x_{l_i}\| \to 0, \text{ when } \quad i \to \infty,$$

which deduces from continuity of gradient that

$$\|g_{t_i} - g_{l_i}\| \to 0,$$

which contradicts (6.1.49), because (6.1.49) implies that $\|g_{t_i} - g_{l_i}\| \geq \epsilon$. The contradiction proves our conclusion. □

6.1.3 Solving A Trust-Region Subproblem

The Dogleg Method and The Double Dogleg Method

An efficient implementation to solve the trust-region subproblem is the so-called dogleg method which was presented by Powell [260]. To find an approximate solution of the subproblem (6.1.1), i.e., to find $x_{k+1} = x_k + s_k$ such that $\|s_k\| = \Delta_k$, Powell used a path consisting of two line segments to approximate s. The first line segment runs from the origin to the Cauchy point (a minimizer C.P. generated by the steepest descent method); the second line segment runs from the Cauchy point C.P. to the Newton point (the minimizer x_{k+1}^N generated by Newton method or quasi-Newton method). Let x_{k+1} be the intersection point of the path and the trust-region boundary. Obviously, $\|x_{k+1} - x_k\| = \Delta_k$. When the Newton step s_k^N satisfies $\|s_k^N\| \leq \Delta_k$, the new iterate x_{k+1} is just the Newton point, $x_{k+1} = x_{k+1}^N = x_k - B_k^{-1} g_k$.

Dennis and Mei [90] found that if the point generated by trust-region iteration is biased towards the Newton direction, the behavior of the algorithm will be further improved. Then we choose a point $\hat{N}$ on the Newton direction, and connect the Cauchy point C.P. to $\hat{N}$. The intersection point of the connection line and the trust-region boundary is taken as the new iterate x_{k+1} (see $x_{k+1}^{(2)}$ in Figure 6.1.1). Comparatively, $x_{k+1}^{(2)}$ is more biased to the Newton direction than $x_{k+1}^{(1)}$. We say $x_k \to C.P. \to x_{k+1}^N$ as dogleg, and $x_k \to C.P. \to \hat{N} \to x_{k+1}^N$ as double dogleg.

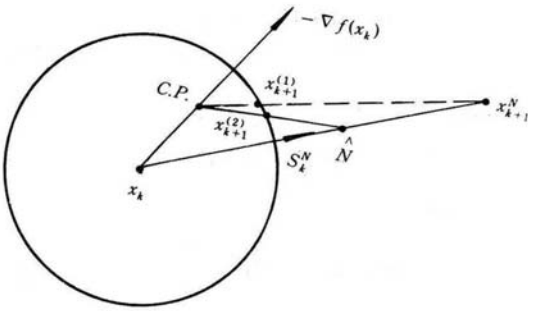


Figure 6.1.1 Dogleg method and double dogleg method

For quadratic model

$$q^{(k)}(x_k - \alpha g_k) = f(x_k) - \alpha \|g_k\|_2^2 + \frac{1}{2}\alpha^2 g_k^T B_k g_k,$$

the exact line search factor α_k has the obvious representation

$$\alpha_k = \frac{\|g_k\|_2^2}{g_k^T B_k g_k}.$$

Then the step along the steepest descent direction is

$$s_k^c = -\alpha_k g_k = -\frac{g_k^T g_k}{g_k^T B_k g_k} g_k. \tag{6.1.54}$$

If $\|s_k^c\|_2 = \|\alpha_k g_k\|_2 \geq \Delta_k$, we take

$$s_k = -\frac{\Delta_k}{\|g_k\|_2} g_k \tag{6.1.55}$$

and

$$x_{k+1} = x_k - \frac{\Delta_k}{\|g_k\|_2} g_k \tag{6.1.56}$$

which lies at the intersection of the negative gradient and the trust-region boundary. If $\|s_k^c\|_2 < \Delta_k$ and $\|s_k^N\|_2 > \Delta_k$, we take

$$s_k(\lambda) = s_k^c + \lambda(s_k^N - s_k^c),\ 0 \leq \lambda \leq 1,$$

and thus

$$x_{k+1} = x_k + s_k(\lambda) = x_k + s_k^c + \lambda(s_k^N - s_k^c),\ 0 \leq \lambda \leq 1, \tag{6.1.57}$$

where the value λ is obtained by solving the equation

$$\|s_k^c + \lambda(s_k^N - s_k^c)\|_2 = \Delta_k.$$

Otherwise, take

$$s_k = s_k^N = -B_k^{-1} g_k. \tag{6.1.58}$$

Combining (6.1.56), (6.1.57) and (6.1.58) yields

$$x_{k+1} = \begin{cases} x_k - \frac{\Delta_k}{\|g_k\|_2} g_k, & \text{when } \|s_k^c\|_2 \geq \Delta_k, \\ x_k + s_k^c + \lambda(s_k^N - s_k^c), & \text{when } \|s_k^c\|_2 < \Delta_k \text{ and } \|s_k^N\|_2 > \Delta_k, \\ x_k - B_k^{-1} g_k, & \text{when } \|s_k^c\|_2 < \Delta_k \text{ and } \|s_k^N\|_2 \leq \Delta_k, \end{cases} \tag{6.1.59}$$

where $0 \leq \lambda \leq 1$.

The following theorem demonstrates the property possessed by the dogleg method and the double dogleg method. In the following, as an example, we only consider the double dogleg method.

Theorem 6.1.11 *In the double dogleg method,*

1. *The distance from x_k to C.P., to $\hat{N}$, is increasing monotonically.*
2. *The model value $q^{(k)}(x_k+s)$ is decreasing monotonically when the point moves from x_k to C.P., to $\hat{N}$, and to x_{k+1}^N.*

Proof. (1) Since

$$\begin{aligned} \|s_k^c\| &= \| - \alpha_k g_k\| = \|g_k\|_2^3 / g_k^T B_k g_k \\ &\leq \frac{\|g_k\|_2^3}{g_k^T B_k g_k} \frac{\|g_k\|_2 \|B_k^{-1} g_k\|_2}{g_k^T B_k^{-1} g_k} \\ &= \frac{\|g_k\|_2^4}{(g_k^T B_k g_k)(g_k^T B_k^{-1} g_k)} \|s_k^N\|_2 \\ &\triangleq \gamma \|s_k^N\|_2, \end{aligned} \tag{6.1.60}$$

it follows from Kantorovich inequality (3.1.33) that $\gamma \leq 1$ and then

$$\|s_k^c\|_2 \leq \gamma \|s_k^N\|_2 \leq \|s_k^N\|_2. \tag{6.1.61}$$

Take $\hat{N}$ being

$$x^{\hat{N}} = x_k - \eta B_k^{-1} g_k = x_k + \eta s_k^N, \tag{6.1.62}$$

where

$$\gamma \leq \eta \leq 1. \tag{6.1.63}$$

Thus

$$\|x^c - x_k\|_2 \leq \|x^{\hat{N}} - x_k\|_2 \leq \|x_{k+1}^N - x_k\|_2 \tag{6.1.64}$$

which shows the property (1) holds.

(2) It is enough to prove that $q^{(k)}(x_k + s)$ decreases monotonically when the point moves from the point C.P. to the point $\hat{N}$. In fact,

$$x_{k+1}(\lambda) = x_k + s_k^c + \lambda(\eta s_k^N - s_k^c),\ 0 \leq \lambda \leq 1. \tag{6.1.65}$$

The direction derivative of $q^{(k)}$ at $x_{k+1}(\lambda)$ is

$$\begin{aligned} & \nabla q^{(k)}(x_{k+1}(\lambda))^T(\eta s_k^N - s_k^c) \\ = \ & (g_k + B_k s_k^c)^T(\eta s_k^N - s_k^c) + \lambda(\eta s_k^N - s_k^c)^T B_k(\eta s_k^N - s_k^c). \end{aligned} \tag{6.1.66}$$

When B_k is positive definite, the right-hand side of (6.1.66) is a monotone increasing function of λ. Therefore, in order to make the above equality negative when $0 \leq \lambda \leq 1$, it is enough to ask the above equality to be negative when $\lambda = 1$, i.e.,

$$(g_k + B_k s_k^c)^T(\eta s_k^N - s_k^c) + \lambda(\eta s_k^N - s_k^c)B_k(\eta_k s_k^N - s_k^c) < 0.$$

Developing and using $B_k s_k^N = -g_k$, the above inequality is equivalent to

$$0 > (1-\eta)(g_k^T(\eta s_k^N - s_k^c)) = (1-\eta)(\gamma - \eta)(g_k^T B_k^{-1} g_k). \tag{6.1.67}$$

Obviously, it is satisfied when $\gamma < \eta < 1$. Therefore the second property holds. □

In summary, the double dogleg method chooses the point $\hat{N}$ which is defined by

$$x_{k+1} = x_k + \eta s_k^N,\ \eta \in [\gamma, 1]. \tag{6.1.68}$$

When $\eta = 1$, the point $\hat{N}$ is just the Newton point x_{k+1}^N and the double dogleg step is just the dogleg step. Generally, we take $\eta = 0.8\gamma + 0.2$.

After generating the points C.P. and $\hat{N}$, we find $x_{k+1}(\lambda)$ by (6.1.65), such that

$$\|s_k^c + \lambda(\eta s_k^N - s_k^c)\|_2^2 = \Delta_k^2, \tag{6.1.69}$$

which is a one-dimensional root-finding problem and can be solved by Newton's method. If $x_{k+1}(\lambda)$ obtained satisfies the descent requirement

$$f(x_{k+1}(\lambda)) \leq f(x_k) + \rho g_k^T(x_{k+1}(\lambda) - x_k),\ \rho \in (0, \frac{1}{2}), \tag{6.1.70}$$

$x_{k+1}(\lambda)$ will be accepted as new iterate x_{k+1}, and the trust-region will be updated by Step 4 in Algorithm 6.1.1; if $x_{k+1}(\lambda)$ does not satisfy (6.1.70), then set $x_{k+1} := x_k$.

Steihaug-CG Method

The methods for solving the trust-region subproblem described above require the solution of a linear system. When the problem is large, the operation may be quite costly. Steihaug [322] proposed a technique based on a preconditioned and truncated conjugate gradient method and trust-region method, which solves the trust-region subproblem approximately. This method is usually called Steihaug-CG method. Since it was independently proposed by Toint [341], it is also called the Steihaug-Toint method.

Consider a scaled trust-region subproblem

$$\min \quad q(s) = g^T s + \frac{1}{2} s^T B s \tag{6.1.71}$$

$$\text{s.t.} \quad \|s\|_W \leq \Delta, \tag{6.1.72}$$

(we drop the subscripts here for simplicity) where W is a symmetric and positive definite matrix. Steihaug applied the preconditioned conjugate gradient method (PCG) to subproblem (6.1.71)–(6.1.72), and considered three possible termination rules. Firstly, if $d_k^T B d_k > 0$, the method corresponds to the convex interior solution. Secondly, if $d_k^T B d_k \leq 0$, we meet a direction of negative curvature. In this case, we move to the trust-region boundary along the line $s_k + \tau d_k$ with $\tau > 0$ so that $\|s_k + \tau d_k\|_W = \Delta$. Finally, if the solution lies outside the trust-region, we ask that the new point be on the boundary.

The Steihaug-CG algorithm for trust-region subproblem is as follows.

Algorithm 6.1.12 *(Steihaug-CG Algorithm for TR Subproblem)*

Step 0. *Given* $\varepsilon > 0$. *Let* $s_0 = 0, g_0 = g, v_0 = W^{-1} g_0, d_0 = -v_0$.
If $\|g_0\| < \varepsilon$, *set* $s = s_0$, *stop;*
For $j = 0, 1, \cdots$, *perform the following steps:*

Step 1. *If* $d_j^T B d_j \le 0$, *compute* $\tau > 0$ *so that* $\|s_j + \tau d_j\|_W = \Delta$,
set $s = s_j + \tau d_j$,
stop;
End if

Step 2. *Set* $\alpha_j = g_j^T v_j / d_j^T B d_j$;
Set $s_{j+1} = s_j + \alpha_j d_j$;
If $\|s_{j+1}\|_W \ge \Delta$, *compute* $\tau > 0$ *so that* $\|s_j + \tau d_j\|_W = \Delta$,
set $s = s_j + \tau d_j$,
stop;
End if

Step 3. *Set* $g_{j+1} = g_j + \alpha_j B d_j$;
If $\|g_{j+1}\|_W < \varepsilon \|g_0\|_W$, *set* $s = s_{j+1}$, *stop;*
End if

Step 4. *Set*
$v_{j+1} = W^{-1} g_{j+1}$,
$\beta_j = g_{j+1}^T v_{j+1} / g_j^T v_j$,
$d_{j+1} = -v_{j+1} + \beta_j d_j$.

This method has some properties similar to the dogleg method. Next, we state these properties. In the proof of the property theorem, we need the following lemma which is easy.

Lemma 6.1.13 *Assume* $d_i^T B d_i \ne 0$, *then we have*

$$g_i^T d_j = -g_j^T v_j, \ 0 \le i \le j, \tag{6.1.73}$$

$$d_i^T W d_j = \frac{g_j^T v_j}{g_i^T v_i} d_i^T W d_i, \ 0 \le i \le j, \tag{6.1.74}$$

$$q(s_{i+1}) = q(s_i) - \frac{1}{2} \frac{(g_i^T v_i)^2}{d_i^T B d_i}. \tag{6.1.75}$$

Proof. We can use the explicit formula for the steplength α and iterative scheme of PCG to get the results. In fact, by $g_j^T d_{j-1} = 0$ and the conjugacy of d_j and d_{j-1}, we have

$$\begin{aligned} g_j^T v_j &= g_j^T (-d_j + \beta_{j-1} d_{j-1}) \\ &= -g_j^T d_j \end{aligned}$$

$$\begin{aligned} &= -(g_{j-1} + \alpha_{j-1} B d_{j-1})^T d_j \\ &= -g_{j-1}^T d_j \\ &= \cdots \\ &= -g_i^T d_j, \quad i \leq j \end{aligned}$$

which shows (6.1.73).

Next, we prove (6.1.74).

$$\begin{aligned} d_j^T W d_i &= (-v_j + \beta_{j-1} d_{j-1})^T W d_i \\ &= -v_j^T W d_i + \beta_{j-1} d_{j-1}^T W d_i \\ &= -g_j^T d_i + \beta_{j-1} d_{j-1}^T W d_i \\ &= \beta_{j-1} d_{j-1}^T W d_i. \end{aligned}$$

By recurrence, we have

$$\begin{aligned} d_j^T W d_i &= \beta_{j-1} \beta_{j-2} \cdots \beta_i d_i^T W d_i \\ &= \frac{g_j^T v_j}{g_i^T v_i} d_i^T W d_i \end{aligned}$$

that shows (6.1.74).

For (6.1.75), it is a direct consequence of (4.3.40). □

Now we are in a position to state the properties of the Steihaug-CG algorithm.

Theorem 6.1.14 *Let $\|s_j\|$ be the iterates generated by PCG Algorithm 6.1.12. Then $q(s_j)$ in (6.1.71) is strictly decreasing, i.e.,*

$$q(s_{j+1}) < q(s_j). \tag{6.1.76}$$

Further, $\|s_j\|_W$ is strictly increasing:

$$0 = \|s_0\|_W < \cdots < \|s_j\|_W < \|s_{j+1}\|_W < \cdots < \|s\|_W \leq \Delta. \tag{6.1.77}$$

Proof. We first prove (6.1.76). From (6.1.75), $q(s_j)$ is strictly decreasing.

Consider the last iterate s. If $s = s_{j+1}$, then the result follows directly. From (6.1.73) we have that

$$g_j^T d_j = -g_j^T v_j = -(Bs_j + g)^T W^{-1} (Bs_j + g) < 0,$$

hence d_j is a descent direction for $q(s_j)$. If $d_j^T B d_j > 0$, then

$$q(s_j) \geq q(s_j + \tau d_j) \geq q(s_{j+1}), \text{ for } 0 < \tau \leq \alpha_j.$$

Since $\tau \leq \alpha_j$, we have the desired result.

For $d_j^T B d_j \leq 0$, then the quadratic term is non-positive, and we have

$$q(s_j) \geq q(s_j + \tau d_j), \text{ for } \tau \geq 0,$$

and the result follows.

Now we show that $\|s_j\|_W$ is strictly increasing and that (6.1.77) holds. From Algorithm 6.1.12, we have

$$s_j = s_0 + \sum_{k=0}^{j-1} \alpha_k d_k = \sum_{k=0}^{j-1} \alpha_k d_k \tag{6.1.78}$$

and

$$\alpha_k > 0, \ k = 0, 1, \cdots, j-1. \tag{6.1.79}$$

Hence, by (6.1.78) and (6.1.74), we have

$$s_j^T W d_j = \sum_{k=0}^{j-1} \alpha_k d_k^T W d_j > 0. \tag{6.1.80}$$

Using (6.1.80) and (6.1.79) gives

$$s_{j+1}^T W s_{j+1} = s_j^T W s_j + 2\alpha_j s_j^T W d_j + \alpha_j^2 d_j^T C d_j \geq s_j^T W s_j \tag{6.1.81}$$

which shows $\|s_j\|_W$ is strictly increasing.

If $s = s_{j+1}$, then (6.1.77) follows directly. If the algorithm stops because $d_j^T B d_j \leq 0$ or $\|s_{j+1}\|_W \geq \Delta$, then the final iterate s is chosen on the boundary, i.e., $\|s\|_W = \Delta$, which is the largest possible length any iterate can have. Therefore (6.1.77) is satisfied. □

Steihaug-CG method is used in Step 3 in Algorithm 6.1.1 for solving the trust-region subproblem. The trust-region method with Steihaug-CG technique is very useful for large-scale optimization problems.

About other techniques of solving subproblems, please consult Gay [144], Moré and Sorensen [222], and Rendl and Wolkowicz [286].

6.2 Conic Model and Collinear Scaling Algorithm

6.2.1 Conic Model

The well-known quadratic model usually considered is

$$q(d) = f(x_k) + g_k^T d + \frac{1}{2} d^T B_k d, \tag{6.2.1}$$

where $g_k = \nabla f(x_k)$ and B_k is a symmetric matrix that is intended to approximate the Hessian matrix. The model (6.2.1) satisfies

$$q(0) = f(x_k),\ \nabla q(0) = \nabla f(x_k). \tag{6.2.2}$$

In quasi-Newton method, the updates satisfy the quasi-Newton condition

$$B_k(x_k - x_{k-1}) = \nabla f(x_k) - \nabla f(x_{k-1}), \tag{6.2.3}$$

which is just the interpolation condition

$$\nabla q(-d) = \nabla f(x_{k-1}). \tag{6.2.4}$$

Therefore, a secant method based on a quadratic model satisfies the three interpolation conditions in (6.2.2) and (6.2.4). However, a quadratic function simply does not possess enough degrees of freedom to incorporate all of the information in the iterative procedure. It often leads to poor prediction of minimizer by these methods based on a quadratic model, especially for those functions with strong non-quadratic behavior or severely changed curvature.

Davidon [82] proposed a new class of algorithm which is able to interpolate richer information on functions and gradients. Such a model function is more general than the quadratic model. This new model is called a conic model. The new algorithm is called a conic model algorithm or a collinear scaling algorithm.

A smooth function is said to be conic if and only if it is a ratio of a quadratic function to the square of an affine function.

Now, we consider the conic model function

$$c(d) = f(x_k) + \frac{g_k^T d}{1 + b^T d} + \frac{1}{2} \frac{d^T A_k d}{(1 + b^T d)^2}. \tag{6.2.5}$$

Its gradient is

$$\begin{aligned}\nabla c(d) &= \frac{(1+b^Td)g_k - g_k^T db}{(1+b^Td)^2} + \frac{(1+b^Td)^2A_kd - (1+b^Td)d^TA_kdb}{(1+b^Td)^4} \\ &= \frac{(1+b^Td)I - bd^T}{1+b^Td} \cdot \frac{(1+b^Td)g_k + A_kd}{(1+b^Td)^2} \\ &= \frac{1}{1+b^Td}\left[I - \frac{bd^T}{1+b^Td}\right]\left[g_k + \frac{A_kd}{1+b^Td}\right]. \end{aligned} \tag{6.2.6}$$

This gradient vanishes, $\nabla c(d) = 0$, if and only if

$$g_k + \frac{A_kd}{1+b^Td} = 0. \tag{6.2.7}$$

In this time, the conic model $c(d)$ has minimizer which is by (6.2.7) that

$$d = \frac{-A_k^{-1}g_k}{1+b^TA_k^{-1}g_k}. \tag{6.2.8}$$

Hence, if $1 + b^TA_k^{-1}g_k \neq 0$, then the desired minimizer is

$$x_{k+1} = x_k - \frac{A_k^{-1}g_k}{1+b^TA_k^{-1}g_k}. \tag{6.2.9}$$

In fact, an essential ingredient of a conic model is to construct a collinear scaling

$$x(d) - x \triangleq \tilde{d} = \frac{d}{1+b^Td} \tag{6.2.10}$$

or

$$d = \frac{\tilde{d}}{1-b^T\tilde{d}}. \tag{6.2.11}$$

In new variable $\tilde{d}$-space, the conic model (6.2.5) becomes a quadratic model

$$c(\tilde{d}) = f(x_k) + g_k^T\tilde{d} + \frac{1}{2}\tilde{d}^TA_k\tilde{d}. \tag{6.2.12}$$

6.2.2 Generalized Quasi-Newton Equation

By means of collinear scaling, Sorensen [315] derived the generalized quasi-Newton equations that the conic model method satisfies.

Let collinear scaling be

$$x(w) = x + \frac{Jw}{1 + h^T w}, \tag{6.2.13}$$

where $J \in R^{n \times m}$, $h \in R^m$, and $w \in R^m$. The local quadratic model to the scaled objective function $\phi(w) = f(x(w))$ has the form

$$\psi(w) = \phi(0) + \phi'(0)w + \frac{1}{2} w^T B w. \tag{6.2.14}$$

Obviously,

$$\phi'(w) = f'(x(w))x'(w) \tag{6.2.15}$$

with

$$x'(w) = \frac{1}{1 + h^T w} J \left[I - \frac{w h^T}{1 + h^T w} \right]. \tag{6.2.16}$$

In terms of the objective function f and the matrix J in the collinear scaling, the quadratic model has the form

$$\psi(w) = f(x) + f'(x)Jw + \frac{1}{2} w^T B w. \tag{6.2.17}$$

If B is positive definite, then the step v that solves

$$v^T B = -f'(x)J \tag{6.2.18}$$

is a predicted minimizer of the scaled function $\phi(w)$. The step s from x to $\bar{x}$ is

$$s = \frac{Jv}{1 + h^T v}, \tag{6.2.19}$$

so that $\bar{x} = x(v)$ in (6.2.13).

Next, we develop the generalized quasi-Newton equations that the conic model satisfies. Let $\bar{x}$ be not an acceptable approximation to a local minimizer of f. Then we wish to update the collinear scaling and also the quadratic model of the new scaled function

$$\bar{\phi}(w) = f(\bar{x}(w)). \tag{6.2.20}$$

Here

$$\bar{x}(w) = \bar{x} + \frac{\bar{J}w}{1 + \bar{h}^T w} \tag{6.2.21}$$

is a collinear scaling with barred quantities $\bar{J}, \bar{h}, \bar{x}$ replacing J, h, x in (6.2.13). The corresponding new quadratic model of the new scaled function is

$$\bar{\psi}(w) = \bar{\phi}(0) + \bar{\phi}'(0)w + \frac{1}{2}w^T \bar{B}w. \tag{6.2.22}$$

For convenience of discussion, we write the derivatives as follows:

$$\bar{x}'(w) = \frac{(1 + \bar{h}^T w)\bar{J} - \bar{J}w\bar{h}^T}{(1 + \bar{h}^T w)^2}, \tag{6.2.23}$$

$$\bar{x}'(0) = \bar{J}, \tag{6.2.24}$$

$$\bar{x}'(-v) = \frac{(1 - \bar{h}^T v)\bar{J} + \bar{J}v\bar{h}^T}{(1 - \bar{h}^T v)^2}, \tag{6.2.25}$$

$$\bar{\phi}'(w) = f'(\bar{x}(w))\bar{x}'(w), \tag{6.2.26}$$

$$\bar{\phi}'(0) = f'(\bar{x}(0))\bar{x}'(0) = f'(\bar{x})\bar{J}, \tag{6.2.27}$$

$$\bar{\phi}'(-v) = f'(\bar{x}(-v))\bar{x}'(-v) = f'(x)(\bar{J} + s\bar{h}^T)/\gamma, \tag{6.2.28}$$

$$\bar{\psi}'(w) = \bar{\phi}'(0) + w^T \bar{B}, \tag{6.2.29}$$

$$\bar{\psi}'(0) = \bar{\phi}(0) = f'(\bar{x})\bar{J}, \tag{6.2.30}$$

$$\bar{\psi}'(-v) = \bar{\phi}'(0) - v^T \bar{B} = f'(\bar{x})\bar{J} - v^T \bar{B}, \tag{6.2.31}$$

where

$$\gamma = 1 + \bar{h}^T(-v) = 1 - \bar{h}^T v. \tag{6.2.32}$$

Then, (6.2.22) can be written as

$$\bar{\psi}(w) = f(\bar{x}) + \frac{\nabla f(\bar{x})^T \bar{s}}{1 - \bar{h}^T \bar{J}^{-1}\bar{s}} + \frac{1}{2}\frac{\bar{s}^T \bar{J}^{-T}\bar{B}\bar{J}^{-1}\bar{s}}{(1 - \bar{h}^T \bar{J}^{-1}\bar{s})^2} \tag{6.2.33}$$

$$= f(\bar{x}) + f'(\bar{x})\bar{J}w + \frac{1}{2}w^T \bar{B}w, \tag{6.2.34}$$

where

$$\bar{s} = \bar{J}w/(1 + \bar{h}^T w), \tag{6.2.35}$$

$$w = \bar{J}^{-1}\bar{s}/(1 - \bar{h}^T \bar{J}^{-1}\bar{s}). \tag{6.2.36}$$

To update J, h, B to $\bar{J}, \bar{h}, \bar{B}$, we introduce the consistency condition

$$\bar{x}(0) = \bar{x},\ \bar{x}(-v) = x \tag{6.2.37}$$

and the interpolation conditions:

$$\bar{\psi}(0) = \bar{\phi}(0),\ \bar{\psi}'(0) = \bar{\phi}'(0), \tag{6.2.38}$$

$$\bar{\psi}(-v) = \bar{\phi}(-v),\ \bar{\psi}'(-v) = \bar{\phi}'(-v). \tag{6.2.39}$$

From the consistency condition $\bar{x}(-v) = x$, we have

$$x = \bar{x}(-v) = \bar{x} - \bar{J}v/\gamma, \tag{6.2.40}$$

that is

$$\bar{J}v = \gamma s, \tag{6.2.41}$$

where $s = \bar{x} - x$. Obviously, conditions (6.2.38) are immediately met by the quadratic model (6.2.22). Also, consider the interpolation conditions (6.2.39); since

$$\begin{aligned} \bar{\psi}(-v) &= \bar{\phi}(0) - \bar{\phi}'(0)v + \frac{1}{2}v^T\bar{B}v \\ &= f(\bar{x}) - f'(\bar{x})\bar{J}v + \frac{1}{2}v^T\bar{B}v \\ &= f(\bar{x}) - \gamma f'(\bar{x})s + \frac{1}{2}v^T\bar{B}v \end{aligned}$$

and

$$\bar{\phi}(-v) = f(x),$$

then the first equation of (6.2.39) becomes

$$f(x) = f(\bar{x}) - \gamma f'(\bar{x})s + \frac{1}{2}v^T\bar{B}v. \tag{6.2.42}$$

Similarly, it follows from (6.2.31) and (6.2.28) that the second equation of (6.2.39) becomes

$$f'(x)(\bar{J} + s\bar{h}^T)/\gamma = f'(\bar{x})\bar{J} - v^T\bar{B}, \tag{6.2.43}$$

that can be written as

$$\bar{B}v = r, \tag{6.2.44}$$

where

$$r^T = \bar{\phi}'(0) - \bar{\phi}'(-v) = f'(\bar{x})\bar{J} - f'(x)(\bar{J} + s\bar{h}^T)/\gamma, \tag{6.2.45}$$

which is the gradient difference of the scaled function.

Then, we obtain a generalized quasi-Newton equation

$$\bar{B}v = r,\ \bar{J}v = \gamma s,\ \bar{h}^T v = 1 - \gamma, \tag{6.2.46}$$

where r is defined by (6.2.45). In particular, when $\bar{J} = I, \bar{h} = 0, \gamma = 1$, the generalized quasi-Newton equations reduce to the usual quasi-Newton equation

$$\bar{B}v = r. \tag{6.2.47}$$

At this time, $v = s = \bar{x} - x$ and $r = f'(\bar{x}) - f'(x)$.

It remains to determine the choices of γ. By the second and the third equations of (6.2.46), we have

$$(\bar{J} + s\bar{h}^T)v = s, \tag{6.2.48}$$

so that

$$v^T \bar{B} v = r^T v = (\gamma f'(\bar{x}) - f'(x)/\gamma)s \triangleq y^T s, \tag{6.2.49}$$

where

$$y = \gamma f'(\bar{x})^T - f'(x)^T/\gamma. \tag{6.2.50}$$

Substituting the above into (6.2.42) gives

$$\gamma^2 f'(\bar{x})s + 2\gamma[f(x) - f(\bar{x})] + f'(x)s = 0. \tag{6.2.51}$$

To make γ real, we must require

$$\rho^2 \triangleq (f(\bar{x}) - f(x))^2 - (f'(\bar{x})s)(f'(x)s) \geq 0. \tag{6.2.52}$$

If $\bar{B}$ is to be positive definite, then we obtain

$$v^T \bar{B} v = 2\rho \tag{6.2.53}$$

from (6.2.49) by taking

$$\gamma = \frac{-f'(x)s}{f(x) - f(\bar{x}) + \rho} \tag{6.2.54}$$

$$= \frac{f(x) - f(\bar{x}) + \rho}{-f'(\bar{x})s} \tag{6.2.55}$$

as the positive root of (6.2.51).

For the one-dimensional case, the corresponding conic model iteration is as follows.

Algorithm 6.2.1 *(Conic Model Algorithm for One-dimensional Case)*

Step 0. Given x_1, s_1, *evaluate* f_1, f_1' *at* x_1;

Step k. for $k = 1, 2, \cdots$

Step k.1 set $x_{k+1} = x_k + s_k$;

Step k.2 evaluate f_{k+1}, f_{k+1}';

Step k.3 set $\rho_k = ((f_k - f_{k+1})^2 - (f_k' s_k)(f_{k+1}' s_k))^{\frac{1}{2}}$;
$\gamma_k = -f_k' s_k / (f_k - f_{k+1} + \rho_k)$;

Step k.4 $s_{k+1} = s_k / [(1/\gamma_k^3)(f_k'/f_{k+1}') - 1]$. □

6.2.3 Updates that Preserve Past Information

Based on the generalized quasi-Newton equations and other criteria, we can obtain some updates about J, h, and B.

Let $\mathcal{W}$ be the linear span of previous scaled search directions and let $\bar{\mathcal{W}} = \text{span}\{\mathcal{W}, v\}$. Then a natural requirement is that

$$\bar{\phi}(w - v) = \phi(w), \ \forall w \in N_0 \subset \mathcal{W}, \tag{6.2.56}$$

where $N_0 = \{w \in \mathcal{W} : \ 1 + h^T w > 0\}$. Condition (6.2.56) immediately leads to the requirement

$$\bar{x}(w - v) = x(w), \ \forall w \in N_0 \subset \mathcal{W}. \tag{6.2.57}$$

Since $\bar{x}(-v) = x$ and $\bar{x}(0) = \bar{x}$, it follows that

$$\begin{aligned}
\bar{x}(w - v) &= \bar{x}(0) + \frac{\bar{J}(w - v)}{\bar{h}^T(w - v) + 1} \\
&= x + \frac{\bar{J}v}{\gamma} + \frac{\bar{J}(w - v)}{\bar{h}^T w + \gamma} \ \text{(by (6.2.46)(iii))} \\
&= x + \frac{\bar{J}v(\bar{h}^T w/\gamma) + \bar{J}w}{\bar{h}^T w + \gamma} \\
&= x + \frac{(\bar{J} + s\bar{h}^T)w}{\bar{h}^T w + \gamma} \ \text{(by (6.2.46)(ii))}.
\end{aligned} \tag{6.2.58}$$

By (6.2.57) and (6.2.58), we have

$$x + \frac{(\bar{J} + s\bar{h}^T)w}{\bar{h}^T w + \gamma} = x(w) = x + \frac{Jw}{h^T w + 1}. \tag{6.2.59}$$

Set $w = \alpha p, p \in N_0 \subset \mathcal{W}, \alpha \in [0, 1]$. Matching the coefficients of α on both sides of (6.2.59) yields

$$(\bar{J} + s\bar{h}^T)p = \gamma Jp,\ \bar{h}^T p = \gamma h^T p$$

for every $p \in N_0$. Then we obtain

$$(\bar{J} + s\bar{h}^T)w = \gamma Jw \tag{6.2.60}$$

and

$$\bar{h}^T w = \gamma h^T w,\ \forall w \in \mathcal{W}. \tag{6.2.61}$$

Since

$$s = \frac{Jv}{h^T v + 1},$$

then (6.2.60) becomes

$$(\bar{J} + \gamma s h^T)w = \gamma Jw,\ \forall w \in \mathcal{W},$$

that is

$$\bar{J} = \gamma(J - sh^T) \tag{6.2.62}$$

satisfying $\bar{J}v = \gamma s$ as well as (6.2.60). The equation (6.2.62) is an update about J.

Next, we discuss the update about h. Note that $\bar{h}$ satisfies

$$\bar{h}^T w = \gamma h^T w,\ \bar{h}^T v = 1 - \gamma. \tag{6.2.63}$$

Now let Q be an orthogonal projector on $\mathcal{W}$ and $P = I - Q$. Let

$$\bar{h} = Qc + Pd, \tag{6.2.64}$$

where c and d are arbitrary vectors. Multiplying (6.2.64) by w^T gives

$$\gamma h^T w = \bar{h}^T w = w^T Qc = c^T w,$$

then we take $c = \gamma h$. Further, multiplying (6.2.64) by v^T yields

$$1 - \gamma = \bar{h}^T v = \gamma v^T Qh + v^T Pd. \tag{6.2.65}$$

Then

$$\bar{h}^T v = 1 - \gamma = \gamma v^T Q h + \frac{1 - \gamma - \gamma v^T Q h}{v^T P d} v^T P d. \tag{6.2.66}$$

Hence, we take

$$\bar{h} = \gamma Q h + P d \tag{6.2.67}$$

or

$$\bar{h} = \gamma Q h + \frac{1 - \gamma - \gamma v^T Q h}{v^T P d} P d \tag{6.2.68}$$

as long as $v^T P d \neq 0$. So, we obtain the updates about h.

By use of (6.2.27) and (6.2.28), it follows from (6.2.60) that

$$\begin{aligned} \bar{\phi}'(-v)w &= \frac{f'(x)}{\gamma}(\bar{J} + s\bar{h}^T)w \\ &= \frac{f'(x)}{\gamma} \cdot \gamma J w \\ &= f'(x) J w \\ &= \phi'(0) w. \end{aligned} \tag{6.2.69}$$

To update the Hessian of the quadratic model of a scaled function, the following requirements are imposed:

$$\bar{\psi}(w - v) = \psi(w), \tag{6.2.70}$$

$$\bar{\psi}'(w - v)q = \psi'(w)q, \tag{6.2.71}$$

for all $w, q \in \mathcal{W}$. Condition (6.2.70) implies that

$$\bar{\phi}(0) + \bar{\phi}'(0)(w - v) + \frac{1}{2}(w - v)^T \bar{B}(w - v) = \phi(0) + \phi'(0)w + \frac{1}{2} w^T B w$$

for all $w \in \mathcal{W}$. Arranging it gives

$$\begin{aligned} &\left[\bar{\phi}(0) - \bar{\phi}'(0)v + \frac{1}{2} v^T \bar{B} v - \phi(0)\right] + [\bar{\phi}'(0) - \phi'(0) - v^T \bar{B}]w \\ &+ \frac{1}{2} w^T (\bar{B} - B) w = 0, \ \forall w \in \mathcal{W}. \end{aligned}$$

The first term vanishes identically due to (6.2.42), and the second term vanishes due to (6.2.60) and (6.2.43). Therefore, we have

$$w^T (\bar{B} - B) w = 0, \forall w \in \mathcal{W}. \tag{6.2.72}$$

Similarly, condition (6.2.71) implies

$$[\bar{\phi}'(0) + (w - v)^T \bar{B}]q = [\phi'(0) + w^T B]q,$$

that is

$$[\bar{\phi}'(0) - \phi'(0) - v^T \bar{B}]q + w^T(\bar{B} - B)q = 0, \ \forall w, q \in \mathcal{W}.$$

The first bracket in the left-hand side of the above equation vanishes due to (6.2.60) and (6.2.43). Then we also get

$$w^T(\bar{B} - B)q = 0, \ \forall w, q \in \mathcal{W}. \tag{6.2.73}$$

Hence, the above discussion shows that if and only if

$$w^T(\bar{B} - B)q = 0, \ \forall w, q \in \mathcal{W}, \tag{6.2.74}$$

both (6.2.70) and (6.2.71) are satisfied.

Consequently, the required update satisfies

$$\bar{B}v = r, \ w^T(\bar{B} - B)q = 0, \ \forall w, q \in \mathcal{W}. \tag{6.2.75}$$

The above can be written as

$$\begin{aligned} \bar{B} &= \mathcal{U}_Q(B, v, r) \\ &= \left\{ \bar{B} \;\middle|\; \begin{array}{l} \bar{B}v = r, Q^T(\bar{B} - B)Q = 0, \ \bar{B} \text{ symmetric}, \\ Q \text{ is an orthogonal projector in } \mathcal{W}. \end{array} \right\}. \end{aligned} \tag{6.2.76}$$

Here the update class coming from additional requirements (6.2.70)-(6.2.71) is bigger than the update class due to Schnabel [299]:

$$\left\{ \bar{B} \,|\, \bar{B}v = r, (\bar{B} - B)w = 0, \ \forall w \in \mathcal{W}, \bar{B} \text{ symmetric} \right\}.$$

Also, the update (6.2.75) includes the optimal conditioning update due to Davidon [80].

From the above discussion, we have obtained a class of updates:

$$\bar{J} = \gamma(J - sh^T), \tag{6.2.77}$$

$$\bar{h} = \gamma Qh + \frac{1 - \gamma - \gamma v^T Qh}{v^T Pd} Pd, \tag{6.2.78}$$

$$w^T(\bar{B} - B)q = 0, \ \forall w, q \in \mathcal{W}. \tag{6.2.79}$$

6.2.4 Collinear Scaling BFGS Algorithm

Sorensen [315] developed a collinear scaling BFGS algorithm without projections. That is, in (6.2.77)–(6.2.79) we take $P = I$ and $Q = 0$. Further, we take

$$d = \bar{J}^T g,\ s = \bar{x} - x,\ y = \gamma\bar{g} - g/\gamma.$$

Then we obtain the following updating formulas:

$$\bar{J} = \gamma(J - sh^T), \tag{6.2.80}$$

$$\bar{h} = \left(\frac{1-\gamma}{\gamma g^T s}\right)\bar{J}^T g, \tag{6.2.81}$$

$$\bar{H} = H + \frac{v(v-Hr)^T}{v^T r} + \frac{(v-Hr)v^T}{v^T r} - \frac{r^T(v-Hr)}{(v^T r)^2}vv^T, \tag{6.2.82}$$

which is called a collinear scaling BFGS formula for updating the inverse Hessian approximation H, where $H = B^{-1}$ and $\bar{H} = \bar{B}^{-1}$.

Further, denote

$$C = JHJ^T,\ \bar{C} = \bar{J}\bar{H}\bar{J}^T. \tag{6.2.83}$$

Using (6.2.80), (6.2.81) and (6.2.45), we have

$$\begin{aligned} r &= \bar{J}^T\bar{g} - \frac{1}{\gamma}(\bar{J}^T + \bar{h}s^T)g = \bar{J}^T\bar{g} - \frac{1}{\gamma}(\bar{J}^T + \frac{1-\gamma}{\gamma}\bar{J}^T)g \\ &= \bar{J}^T\bar{g} - \frac{1}{\gamma^2}\bar{J}^T g = (J - sh^T)^T y. \end{aligned} \tag{6.2.84}$$

Thus

$$\bar{J}\left(I - \frac{vr^T}{v^T r}\right) = \gamma(J - sh^T) - \frac{\gamma s(J^T y - hs^T y)^T}{s^T y} = \gamma\left(I - \frac{sy^T}{s^T y}\right)J. \tag{6.2.85}$$

Equation (6.2.85) allows us to obtain

$$\bar{C} = \gamma^2\left[\left(I - \frac{sy^T}{s^T y}\right)C\left(I - \frac{ys^T}{s^T y}\right) + \frac{ss^T}{s^T y}\right]. \tag{6.2.86}$$

So, instead of updating J and H, we only need to update C.

By (6.2.21), the scaled direction is

$$s_{k+1} = \frac{1}{1 + h_{k+1}^T v_{k+1}} J_{k+1}v_{k+1} \ \ (\text{note } v_{k+1} = -H_{k+1}J_{k+1}^T g_{k+1})$$

$$
\begin{aligned}
&= \frac{-J_{k+1}H_{k+1}J_{k+1}^T g_{k+1}}{1-(1-\gamma_k)g_k^T J_{k+1}H_{k+1}J_{k+1}^T g_{k+1}/(\gamma_k g_k^T s_k)} \\
&\triangleq \frac{-J_{k+1}H_{k+1}J_{k+1}^T g_{k+1}}{1+\delta_{k+1}} \\
&\triangleq -\theta_{k+1}C_{k+1}g_{k+1}. \qquad (6.2.87)
\end{aligned}
$$

Hence, we obtain the iterative scheme

$$x_{k+1} = x_k - \theta_k C_k g_k, \qquad (6.2.88)$$

where

$$
\begin{aligned}
&\theta_k = 1/(1+\delta_k), \\
&\delta_k = -(1-\gamma_{k-1})g_{k-1}^T C_k g_k/(\gamma_{k-1}g_{k-1}^T s_{k-1}), \\
&C_k = \mathcal{U}(C_{k-1}, s_{k-1}, y_{k-1}).
\end{aligned}
$$

In the following, we give a description of the algorithm.

Algorithm 6.2.2 *(Collinear Scaling BFGS Algorithm)*

Step 1. Initialize C_0 positive definite, $x_0, \delta_0, \alpha_{\max} > 0$. Compute f_0, g_0. Set $k = 0$.

Step 2. If $\delta_k < 0$, set $\bar{\alpha} = \min(\alpha_{\max}, -1/\delta_k)$, else $\bar{\alpha} := \alpha_{\max}$. Do line search for the function

$$\phi(\alpha) \triangleq f\left(x_k - \frac{\alpha}{1+\alpha\delta_k}C_k g_k\right)$$

and find $\alpha_k \in (0, \bar{\alpha})$. Set

$$
\begin{aligned}
&s_k = -\frac{\alpha_k}{1+\alpha_k\delta_k}C_k g_k, \\
&x_{k+1} = x_k + s_k, \\
&f_{k+1} = f(x_{k+1}),\ g_{k+1}^T = f'(x_{k+1}), \\
&\rho^2 = (f_k - f_{k+1})^2 - (g_{k+1}^T s_k)(g_k^T s_k),
\end{aligned}
$$

such that $\rho^2 > 0$ and $f_{k+1} < f_k$.

Step 3. If "convergence" then stop.

Step 4. Compute

$$\gamma_k = -g_k^T s_k/(f_k - f_{k+1} + \rho),\ y_k = \gamma_k g_{k+1} - g_k/\gamma_k,$$
$$C_{k+1} = \gamma_k^2[(I - s_k y_k^T/s_k^T y_k)C_k(I - y_k s_k^T/s_k^T y_k) + s_k s_k^T/s_k^T y_k],$$
$$\delta_{k+1} = -(1-\gamma_k)g_k^T C_{k+1} g_{k+1}/\gamma_k g_k^T s_k.$$

Set $k := k+1$, go to Step 2. □

Following the Broyden-Dennis-Moré convergence theory about quasi-Newton methods, we can establish Q-superlinear convergence of the collinear scaling BFGS algorithm, i.e.,

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

Furthermore, Di and Sun [101] propose a conic trust-region method for unconstrained optimization.

Let x denote the current approximation of the minimizer and let

$$f = f(x),\ g = g(x) = \nabla f(x). \tag{6.2.89}$$

Then the conic trust-region model of $f(x+s)$ is

$$\min \quad \psi(s) = f + \frac{g^T s}{1 - a^T s} + \frac{1}{2}\frac{s^T A s}{(1 - a^T s)^2}, \tag{6.2.90}$$
$$\text{s.t.} \quad \|Ds\| \le \Delta, \tag{6.2.91}$$

where $A \in R^{n\times n}$ is the Hessian approximation at x, $a \in R^n$ is a horizontal vector such that $1 - a^T s > 0$, D is a scaling matrix and Δ a trust-region radius. The above subproblem can be written as

$$\min \quad f + g^T J w + \frac{1}{2} w^T B w \tag{6.2.92}$$
$$\text{s.t.} \quad s = Jw/(1 + h^T w),\ \|Ds\| \le \Delta. \tag{6.2.93}$$

Di and Sun [101] discussed the necessary and sufficient condition of the solution for the conic trust-region subproblem, presented an algorithm and established the global and superlinear convergence. Besides, Zhu etc. [384] discussed a quasi-Newton type trust-region method based on a conic model for solving unconstrained optimization. Sun and Yuan [337], Sun, Yuan, and Yuan [338] studied a conic trust-region algorithm for linear and nonlinear constrained optimization respectively. About the topic of conic model method, readers are referred also to Grandinetti [162], Ariyawansa [3], Sun [333], Sheng [308], Han, Sun et al. [168].

6.3 Tensor Methods

The tensor method is also a generalization of the quadratic model method. In fact, the tensor method is based on the third- or fourth-order model for optimization problems, and intends to improve upon the efficiency and reliability of standard methods on problem where $\nabla^2 f(x^*)$ is singular.

The tensor method was introduced by Schnabel and Frank [302] for solving systems of nonlinear equations and by Schnabel and Chow [301] for unconstrained optimization, respectively. In this section, we will describe the tensor methods for nonlinear equations and for unconstrained optimization.

6.3.1 Tensor Method for Nonlinear Equations

Let $F : R^n \to R^n$. Consider solving nonlinear equations

$$F(x) = 0, \tag{6.3.1}$$

that is to find $x^* \in R^n$ so that $F(x^*) = 0$. Newton's method for (6.3.1) is defined as

$$x_+ = x_c - F'(x_c)^{-1} F(x_c), \tag{6.3.2}$$

when $F'(x_c)$ is nonsingular, where x_c and x_+ denote the current and the next iterate respectively. Newton's method is based on the linear model at x_c,

$$M(x_c + d) = F(x_c) + F'(x_c)d. \tag{6.3.3}$$

As we know, the outstanding advantage of Newton's method is its rapid convergence, that is if $F'(x_c)$ is Lipschitz continuous in the neighborhood of x^* and $F'(x^*)$ is nonsingular, then the sequence produced by (6.3.2) locally and quadratically converges to x^*. This implies that there are $\delta > 0$ and $c \geq 0$, such that when $\|x_0 - x^*\| \leq \delta$, the iterative sequence $\{x_k\}$ satisfies

$$\|x_{k+1} - x^*\| \leq c\|x_k - x^*\|^2. \tag{6.3.4}$$

However, if $F'(x^*)$ is singular, then the iterative sequence does not converge rapidly. The tensor method described in this section will overcome the shortcoming, and we can see that the tensor method still has rapid convergence when $F'(x^*)$ is singular.

Consider the second-order model

$$M_T(x_c + d) = F(x_c) + F'(x_c)d + \frac{1}{2}T_c dd, \tag{6.3.5}$$

where $T_c \in R^{n\times n\times n}$ is a three-dimensional tensor. Usually, the (6.3.5) is said to be a tensor model, and the corresponding method is called a tensor method. To discuss the tensor method, we first give a definition concerning these tensors.

Definition 6.3.1 *Let $T \in R^{n\times n\times n}$. Then T consists of n horizontal faces $H_i \in R^{n\times n}, i = 1, \cdots, n$, where $H_i[j,k] = T[i,j,k]$. For $v, w \in R^n$, we have $Tvw \in R^n$ with the i-th component*

$$Tvw[i] = v^T H_i w = \sum_{j=1}^{n}\sum_{k=1}^{n} T[i,j,k]v[j]w[k]. \tag{6.3.6}$$

Hence, the tensor model given in (6.3.5) is, in fact, an n-dimensional vector in which each component is a quadratic model of the component of $F(x)$, i.e.,

$$(M_T(x_c + d))[i] = f_i + g_i^T d + \frac{1}{2}d^T H_i d,\ i = 1, \cdots, n, \tag{6.3.7}$$

where $f_i = F(x_c)[i]$, g_i^T is the i-th row of $F'(x_c)$, and H_i the Hessian matrix of the i-th component of $F(x)$.

An obvious choice of T_c in (6.3.5) is $F''(x_c)$. However, the computational amount is prohibitive, since, in each iteration, it needs to compute n^3 second-order partial derivatives of $F''(x_c)$, store over $n^3/2$ elements, and solve n quadratic equations in n unknowns. To overcome these drawbacks, the tensor method constructs T_c in low-rank by using available information of function values and first derivatives. So, the additional efforts are small related to the standard method.

To construct T_c, we select p not necessarily consecutive past iterates $x_{-1}, \cdots, x_{-p}$ and ask the model (6.3.5) to interpolate $F(x)$ at these points, i.e.,

$$F(x_{-k}) = F(x_c) + F'(x_c)s_k + \frac{1}{2}T_c s_k s_k,\ k = 1, \cdots, p, \tag{6.3.8}$$

where

$$s_k = x_{-k} - x_c,\ k = 1, \cdots, p. \tag{6.3.9}$$

The selected directions $\{s_k\}$ are required to be strongly independent, i.e., make the angle between each direction s_k and the subspace spanned by other directions have at least θ degree. Values of θ between 20 and 40 degrees have proven to be best in practice. This procedure is easily implemented by

using a modified Gram-Schmidt algorithm. Since directions $\{s_k\}$ are linearly independent, then $p \leq n$. In practice, one takes

$$p \leq \sqrt{n}.$$

Now we write (6.3.8) as

$$T_c s_k s_k = z_k, \ k = 1, \cdots, p, \tag{6.3.10}$$

where

$$z_k = 2(F(x_{-k}) - F(x_c) - F'(x_c)s_k). \tag{6.3.11}$$

The (6.3.10) is a set of $np \leq n^{3/2}$ linear equations in n^3 unknowns $T_c[i, j, k]$, $1 \leq i, j, k \leq n$. We choose the smallest symmetric T_c, in the Frobenius norm, which satisfies the equations (6.3.10). Below, we choose T_c following the technique for a secant update with the smallest change in quasi-Newton methods (see Chapter 5).

First, we define the three-dimensional rank-one tensor.

Definition 6.3.2 *Let $u, v, w \in R^n$. The tensor $T \in R^{n\times n\times n}$, for which*

$$T[i, j, k] = u[i] \cdot v[j] \cdot w[k], \ (1 \leq i, j, k \leq n), \tag{6.3.12}$$

is called a third-order rank-one tensor of $T \in R^{n\times n\times n}$ and is denoted by

$$T = u \otimes v \otimes w. \tag{6.3.13}$$

Obviously, the i-th horizontal face of the rank-one tensor $u \otimes v \otimes w$ is a rank-one matrix $u[i](vw^T)$.

Theorem 6.3.3 *Let $p \leq n$. Let $s_k \in R^n$, $k = 1, \cdots, p$ with $\{s_k\}$ linearly independent, and let $z_k \in R^n$, $k = 1, \cdots, p$. Define $M \in R^{p\times p}$ by $M[i, j] = (s_i^T s_j)^2$, $1 \leq i, j \leq p$, and define $Z \in R^{n\times p}$ with z_k the k-th column, $k = 1, \cdots, p$. Then M is positive definite, and the solution to*

$$\min_{T_c \in R^{n\times n\times n}} \quad \|T_c\|_F \tag{6.3.14}$$

$$\text{s.t.} \quad T_c s_k s_k = z_k, \ k = 1, \cdots p \tag{6.3.15}$$

is

$$T_c = \sum_{k=1}^{p} (a_k \otimes s_k \otimes s_k), \tag{6.3.16}$$

where a_k is the k-th column of $A \in R^{n\times p}$ and $A = M^{-1}Z$.

Proof. Since the objective function and constraints can be decomposed into n separate objective functions and constraints, then (6.3.14)-(6.3.15) are equivalent to the following separate minimization problem

$$\begin{aligned} \min_{H_i \in R^{n\times n}} \quad & \|H_i\|_F && (6.3.17)\\ \text{s.t.} \quad & s_k^T H_i s_k = z_k[i],\ k = 1, \cdots, p, && (6.3.18) \end{aligned}$$

where H_i are the horizontal faces of T_c, $i = 1, \cdots, n$. Note that the problem (6.3.17)-(6.3.18) is a sub-determined set of p equations in n^2 unknowns.

Let $h_i \in R^{n^2}$,

$$\begin{aligned} h_i \quad = \quad & (H_i[1,1], H_i[1,2], \cdots, H_i[1,n], H_i[2,1], \cdots, \\ & H_i[2,n], \cdots, H_i[n,1], \cdots, H_i[n,n])^T. \end{aligned} \tag{6.3.19}$$

Let $\bar{S} \in R^{p\times n^2}$, the k-th row of $\bar{S}$ is

$$\bar{s}_k = (s_k[1]s_k^T, s_k[2]s_k^T, \cdots, s_k[n]s_k^T). \tag{6.3.20}$$

Let also the i-th row of $Z \in R^{n\times p}$ be $\bar{z}_i$,

$$\bar{z}_i \in R^p,\ \bar{z}_i[k] = z_k[i],\ 1 \le i \le n,\ 1 \le k \le p.$$

Then (6.3.17) is equivalent to

$$\begin{aligned} \min_{h_i \in R^{n^2}} \quad & \|h_i\|_2 && (6.3.21)\\ \text{s.t.} \quad & \bar{S}h_i = \bar{z}_i^T. && (6.3.22) \end{aligned}$$

Note that the $\{s_k\}$ are linearly independent, then $\bar{S}$ is full row rank, and hence the solution to (6.3.21)-(6.3.22) is

$$h_i = \bar{S}^T(\bar{S}\bar{S}^T)^{-1}\bar{z}_i^T. \tag{6.3.23}$$

Since $M = \bar{S}\bar{S}^T$, then M is positive definite. Also,

$$\begin{bmatrix} \bar{a}_1 \\ \vdots \\ \bar{a}_n \end{bmatrix} = A = M^{-1}Z = (\bar{S}\bar{S}^T)^{-1} \begin{bmatrix} \bar{z}_1 \\ \vdots \\ \bar{z}_n \end{bmatrix}. \tag{6.3.24}$$

Hence the i-th row of A is

$$\bar{a}_i = (\bar{S}\bar{S}_i^T)^{-1}\bar{z}_i. \tag{6.3.25}$$

Therefore (6.3.23) means

$$h_i = \bar{S}^T \bar{a}_i^T. \tag{6.3.26}$$

Note that here $\bar{a}_i$ is the i-th row of A and a_k is the k-th column of A, then

$$\bar{a}_i[k] = a_k[i],\ 1 \le i \le n,\ 1 \le k \le p.$$

Then it follows from (6.3.26) that

$$h_i = \sum_{k=1}^{p} \bar{a}_i[k]\bar{s}_k^T = \sum_{k=1}^{p} a_k[i]\bar{s}_k^T, \tag{6.3.27}$$

where $\bar{s}_k$ is defined by (6.3.20) and the k-th row of $\bar{S}$, the $\bar{s}_k^T$ denotes a transpose of $\bar{s}_k$ and a column vector with n^2 elements.

Returning to (6.3.27) in the terms of H_i and s_k, and using (6.3.19) and (6.3.20) give

$$H_i = \sum_{k=1}^{p} a_k[i]s_k s_k^T. \tag{6.3.28}$$

Finally, combining n matrices H_i gives the desired T_c in (6.3.16).

Substituting (6.3.16) into (6.3.5), the tensor model has the form

$$M_T(x_c + d) = F(x_c) + F'(x_c)d + \frac{1}{2}\sum_{k=1}^{p} a_k(d^T s_k)^2. \tag{6.3.29}$$

In the above model, the simple form of the second-order term is a key to efficiently find a minimizer of this model. In the tensor method, the additional $4pn$ storage are required to save $\{a_k\}, \{s_k\}, \{x_{-k}\}$ and $\{F(x_{-k})\}$. Additional cost is $n^2p+O(np^2)$ operations for computing $A = M^{-1}Z$. Since $p \le \sqrt{n}$, this is a very small additional cost, more than the cost of the standard quadratic model method.

6.3.2 Tensor Methods for Unconstrained Optimization

In this subsection, we extend the tensor method to solving unconstrained optimization problem

$$\min_{x \in R^n} f(x),\ f : R^n \to R. \tag{6.3.30}$$

Note that the standard quadratic model methods do not converge quickly if the Hessian $\nabla^2 f(x^*)$ is singular. In this case, the convergence rate is linear

at best. Furthermore, the third derivatives do not supply information in the direction where the second derivative matrix is lacking. Thus, adding an approximation to $\nabla^3 f(x_c)$ alone will not lead to better-than-linear convergence. Therefore, we consider employing the following fourth order tensor model

$$m_T(x_c+d) = f(x_c)+\nabla f(x_c)\cdot d+\frac{1}{2}\nabla^2 f(x_c)\cdot d^2+\frac{1}{6}T_c\cdot d^3+\frac{1}{24}V_c\cdot d^4, \quad (6.3.31)$$

where $T_c \in R^{n\times n\times n}$, a three-dimensional tensor and $V_c \in R^{n\times n\times n\times n}$, a four-dimensional tensor, are symmetric. Equation (6.3.31) is called a tensor model for unconstrained optimization; the methods based on (6.3.31) are referred to tensor methods.

How to Choose T_c and V_c?

To select T_c and V_c, we select p not necessarily consecutive past iterates $x_{-1}, \cdots, x_{-p}$, and ask that the model (6.3.31) interpolate $f(x)$ and $\nabla f(x)$ at these points, i.e.,

$$\begin{aligned} f(x_{-k}) &= f(x_c) + \nabla f(x_c)\cdot s_k + \frac{1}{2}\nabla^2 f(x_c)\cdot s_k^2 + \frac{1}{6}T_c\cdot s_k^3 \\ &\quad +\frac{1}{24}V_c\cdot s_k^4, \qquad (6.3.32) \\ \nabla f(x_{-k}) &= \nabla f(x_c) + \nabla^2 f(x_c)\cdot s_k + \frac{1}{2}T_c\cdot s_k^2 \\ &\quad +\frac{1}{6}V_c\cdot s_k^3, \qquad (6.3.33) \end{aligned}$$

where $s_k = x_{-k} - x_c$, $k = 1, \cdots, p$. As in the previous subsection, the directions $\{s_k\}$ are strongly linearly independent. We also set $p \le n^{1/3}$.

Multiplying (6.3.33) by s_k gives

$$\nabla f(x_{-k})\cdot s_k = \nabla f(x_c)\cdot s_k + \nabla^2 f(x_c)\cdot s_k^2 + \frac{1}{2}T_c\cdot s_k^3 + \frac{1}{6}V_c\cdot s_k^4. \quad (6.3.34)$$

Define $\alpha, \beta \in R^p$ respectively by

$$\alpha[k] = T_c\cdot s_k^3, \quad (6.3.35)$$

$$\beta[k] = V_c\cdot s_k^4, \quad (6.3.36)$$

where $k = 1, \cdots, p$. Then (6.3.34) and (6.3.32) have the following form respectively:

$$\frac{1}{2}\alpha[k] + \frac{1}{6}\beta[k] = q_1[k], \quad (6.3.37)$$

$$\frac{1}{6}\alpha[k] + \frac{1}{24}\beta[k] = q_2[k], \tag{6.3.38}$$

where

$$q_1[k] = \nabla f(x_{-k}) \cdot s_k - \nabla f(x_c) \cdot s_k - \nabla^2 f(x_c) \cdot s_k^2, \tag{6.3.39}$$

$$q_2[k] = f(x_{-k}) - f(x_c) - \nabla f(x_c) \cdot s_k - \frac{1}{2}\nabla^2 f(x_c) \cdot s_k^2, \tag{6.3.40}$$

for $k = 1, 2, \cdots, p$. The system (6.3.37)-(6.3.38) is nonsingular, so each $\alpha[k]$ and $\beta[k]$ are uniquely determined. Thus, we can determine V_c by the minimization problem

$$\begin{array}{ll} \min_{V_c \in R^{n\times n\times n\times n}} & \|V_c\|_F \\ \text{s.t.} & V_c \cdot s_k^4 = \beta[k],\ k = 1, \cdots, p. \\ & V_c \textit{ symmetric.} \end{array} \tag{6.3.41}$$

We then substitute the obtained value of V_c into (6.3.33), obtaining

$$T_c \cdot s_k^2 = a_k,\ k = 1, \cdots, p, \tag{6.3.42}$$

where

$$a_k = 2\left(\nabla f(x_{-k}) - \nabla f(x_c) - \nabla^2 f(x_c) \cdot s_k - \frac{1}{6}V \cdot s_k^3\right).$$

This is a set of $np \leq n^{4/3}$ linear equations in n^3 unknowns $T_c[i, j, k]$, $1 \leq i, j, k \leq n$. Then we determine T_c by the minimization problem

$$\begin{array}{ll} \min_{T_c \in R^{n\times n\times n}} & \|T_c\|_F \\ \text{s.t.} & T_c \cdot s_i^2 = a_i,\ i = 1, \cdots, p \\ & T_c \textit{ symmetric.} \end{array} \tag{6.3.43}$$

The following two theorems give the solutions of problems (6.3.41) and (6.3.43).

Theorem 6.3.4 *Let $p \leq n$. Let $s_k \in R^n$, $k = 1, \cdots, p$ with $\{s_k\}$ linearly independent, and let $\beta \in R^p$. Define $M \in R^{p\times p}$ by $M[i, j] = (s_i^T s_j)^4$, $1 \leq i, j \leq p$. Define $\gamma \in R^p$ by $\gamma = M^{-1}\beta$. Then the solution to (6.3.41) is*

$$V_c = \sum_{k=1}^{p} \gamma[k](s_k \otimes s_k \otimes s_k \otimes s_k). \tag{6.3.44}$$

Proof. Define $\hat{v} \in R^{n^4}$ by

$$\begin{aligned}\hat{v}^T = \; & (V_c[1,1,1,1], V_c[1,1,1,2], \cdots, V_c[1,1,1,n], \\ & V_c[1,1,2,1], \cdots, V_c[1,1,2,n], \cdots, V_c[n,n,n,n]).\end{aligned}$$

Let the matrix $\hat{S} \in R^{p \times n^4}$ with the k-th row as

$$(s_k[1])^4, (s_k[1])^3(s_k[2]), \cdots, (s_k[1])^3(s_k[n]), \cdots, (s_k[n])^4.$$

Then, (6.3.41) is equivalent to

$$\min_{\hat{v}} \quad \|\hat{v}\|_2 \tag{6.3.45}$$

$$\text{s.t.} \quad \hat{S}\hat{v} = \beta, \; V_c \textit{ symmetric}, \tag{6.3.46}$$

where V_c is the original form of $\hat{v}$. Since $\{s_k\}$ are linearly independent, $\hat{S}$ has full row rank. Hence, the solution to

$$\min_{\hat{v}} \quad \|\hat{v}\|_2 \tag{6.3.47}$$

$$\text{s.t.} \quad \hat{S}\hat{v} = \beta \tag{6.3.48}$$

is

$$\hat{v} = \hat{S}^T(\hat{S}\hat{S}^T)^{-1}\beta = \hat{S}^T M^{-1}\beta = \hat{S}^T\gamma, \tag{6.3.49}$$

where $M = \hat{S}\hat{S}^T$. By reversing the transformation from $\hat{v}$ to V_c, we get (6.3.44). Since V_c is symmetric, it is the solution of (6.3.41). □

Theorem 6.3.5 *Let $p \leq n$. Let $s_k \in R^n$, $k = 1, \cdots, p$ with $\{s_k\}$ linearly independent, and let $a_k \in R^n$, $k = 1, \cdots, p$. Then the solution to problem (6.3.43) is*

$$T_c = \sum_{k=1}^{p} (b_k \otimes s_k \otimes s_k + s_k \otimes b_k \otimes s_k + s_k \otimes s_k \otimes b_k), \tag{6.3.50}$$

where $b_k \in R^n$, $k = 1, \cdots, p$, and $\{b_k\}$ is the unique set of vectors for which (6.3.50) satisfies

$$T_c s_i^2 = a_i, \; i = 1, \cdots, p.$$

Proof. First, we show that the constraint set in (6.3.43) is feasible. Let $t_i \in R^n, i = 1, \cdots, p$ satisfy

$$t_i^T s_j = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases} \quad \text{for } j = 1, \cdots, p.$$

Since $\{s_i\}$ are linearly independent, such vectors t_i can be obtained via a QR factorization. Then

$$T = \sum_{i=1}^{p} (t_i \otimes t_i \otimes a_i + t_i \otimes a_i \otimes t_i + a_i \otimes t_i \otimes t_i - 2(a_i^T s_i)(t_i \otimes t_i \otimes t_i))$$

is a feasible solution to (6.3.43).

Dennis and Schnabel [93] assume that the set of tensors $T_j \in R^{n\times n\times n}$ is generated by the following procedure: $T_0 = 0$ and for $j = 0, 1, \cdots$, T_{2j+1} is the solution of

$$\min \quad \|T_{2j+1} - T_{2j}\|_F \tag{6.3.51}$$

$$\text{s.t.} \quad T_{2j+1} \cdot s_i^2 = a_i, \ i = 1, \cdots, p, \tag{6.3.52}$$

and T_{2j+2} is the solution of

$$\min \quad \|T_{2j+2} - T_{2j+1}\|_F \tag{6.3.53}$$

$$\text{s.t.} \quad T_{2j+2} \ \textit{symmetric.} \tag{6.3.54}$$

Then the sequence $\{T_j\}$ has a limit which is the unique solution to (6.3.43). (see the derivation of Powell symmetric Broyden update in §5.1).

Next, we show that this limit has form (6.3.50) for some set of vectors $\{b_k\}$, by showing that each T_{2j} has this form.

Trivially, it is true for T_0. Assume it is true for some j, i.e.,

$$T_{2j} = \sum_{k=1}^{p} (u_k \otimes s_k \otimes s_k + s_k \otimes u_k \otimes s_k + s_k \otimes s_k \otimes u_k) \tag{6.3.55}$$

for some set of vectors u_k. Then from Theorem 6.3.3, the solution to (6.3.51)-(6.3.52) is

$$T_{2j+1} = T_{2j} + \sum_{k=1}^{p} (v_k \otimes s_k \otimes s_k)$$

for some set of vectors $\{v_k\}$. Thus

$$\begin{aligned} T_{2j+2} &= T_{2j} + \frac{1}{3}\sum_{k=1}^{p} (v_k \otimes s_k \otimes s_k + s_k \otimes v_k \otimes s_k + s_k \otimes s_k \otimes v_k) \\ &= \sum_{k=1}^{p} \left(\left(u_k + \frac{v_k}{3} \right) \otimes s_k \otimes s_k + s_k \otimes \left(u_k + \frac{v_k}{3} \right) \otimes s_k \right. \\ &\quad \left. + s_k \otimes s_k \otimes \left(u_k + \frac{v_k}{3} \right) \right), \end{aligned} \tag{6.3.56}$$

which again has the form (6.3.55). Thus, by induction, the solution T_c must have the form (6.3.50) for some set of vectors $\{b_k\}$.

Finally, we show that the set of vectors $\{b_k\}$, for which T_c given by (6.3.50) satisfies

$$T_c s_i^2 = a_i, \ i = 1, \cdots, p, \tag{6.3.57}$$

is unique. This will mean that equations (6.3.50) and (6.3.57) uniquely determine the solution to (6.3.43). In fact, substituting (6.3.50) into (6.3.57) gives a system of np linear equations in np unknowns, where the matrix is a function of $\{s_k\}$, the unknowns are the elements of the $\{b_k\}$, and the right-hand side consists of the elements of the $\{a_k\}$.

Since we have showed above that (6.3.43) is feasible for any $\{a_k\}$, the above derivation and the theory of Dennis-Schnabel [93] imply that for any set $\{s_k\}$, this linear system has at least one solution for any right-hand side. Therefore, the linear system must be nonsingular and have a unique solution. This means that the set of vectors $\{b_k\}$ is uniquely determined. □

Solving the Tensor Model

Substituting the values of T_c and V_c in (6.3.50) and (6.3.44) into the tensor model (6.3.31) gives

$$\begin{aligned} m_T(x_c + d) &= f(x_c) + \nabla f(x_c) \cdot d + \frac{1}{2}\nabla^2 f(x_c) \cdot d^2 \\ &\quad \frac{1}{2}\sum_{k=1}^{p}(b_k^T d)(s_k^T d)^2 + \frac{1}{24}\sum_{k=1}^{p}\gamma[k](s_k^T d)^4 \\ &= f(x_c) + g^T d + \frac{1}{2}d^T H d \\ &\quad + \frac{1}{2}\sum_{k=1}^{p}(b_k^T d)(s_k^T d)^2 + \frac{1}{24}\sum_{k=1}^{p}\gamma[k](s_k^T d)^4, \end{aligned} \tag{6.3.58}$$

where $g = \nabla f(x_c)$, $H = \nabla^2 f(x_c)$.

Let $S \in R^{n\times p}$ with k-th column s_k. Let $Z \in R^{n\times(n-p)}$ and $W \in R^{n\times p}$ have full column rank and satisfy $Z^T S = 0$ and $W^T S = I$, respectively. The Z and W can be calculated through the QR factorization of S.

Write

$$d = Wu + Zt, \tag{6.3.59}$$

where $u \in R^p$ and $t \in R^{n-p}$. Substituting (6.3.59) into (6.3.58) gives

$$m_T(x_c + Wu + Zt) = f(x_c) + g^T W u + g^T Z t + \frac{1}{2}u^T W^T H W u$$

$$+u^TW^THZt+\frac{1}{2}t^TZ^THZt+\frac{1}{2}\sum_{k=1}^{p}u[k]^2(b_k^TWu+b_k^TZt)$$
$$+\frac{1}{24}\sum_{k=1}^{p}\gamma[k]u[k]^4, \tag{6.3.60}$$

which is a quadratic with respect to t. Therefore, for the tensor model to have a minimizer, Z^THZ must be positive definite and the derivative of the model with respect to t must be 0, i.e.,

$$Z^Tg+Z^THZt+Z^THWu+\frac{1}{2}Z^T\sum_{i=1}^{p}b_iu[i]^2=0. \tag{6.3.61}$$

Therefore

$$t=-(Z^THZ)^{-1}Z^T\left(g+HWu+\frac{1}{2}\sum_{i=1}^{p}b_iu[i]^2\right). \tag{6.3.62}$$

Substituting (6.3.62) into (6.3.60) reduces the problem of minimizing the tensor model to finding a minimizer of

$$\begin{aligned}\hat{m}_T(u) &= f+g^TWu+\frac{1}{2}u^TW^THWu+\frac{1}{2}\sum_{i=1}^{p}u[i]^2(b_i^TWu)\\ &\quad+\frac{1}{24}\sum_{i=1}^{p}\gamma[i]u[i]^4-\frac{1}{2}\left(g+HWu+\frac{1}{2}\sum_{i=1}^{p}b_iu[i]^2\right)^T\\ &\quad\cdot Z(Z^THZ)^{-1}Z^T\left(g+HWu+\frac{1}{2}\sum_{i=1}^{p}b_iu[i]^2\right),\end{aligned} \tag{6.3.63}$$

which is a fourth-degree polynomial in u-variable. If (6.3.63) has a minimizer u^*, then the minimizer of the original tensor model (6.3.58) is given by

$$d^*=Wu^*+Zt^*, \tag{6.3.64}$$

where t^* is determined by setting $u=u^*$ in (6.3.60).

In implementation we may employ line search or trust-region strategy. If the obtained direction d^* is a descent direction, but x_c+d^* is not acceptable, we set $x_+=x_c+\lambda d^*$ where λ is a steplength factor. If (6.3.63) has no minimizer, or d^* is not in a descent direction, we find the next iterate by using a line search algorithm based on the standard quadratic model.

Similarly, we can also use trust-region technique studied in §6.1. The trust-region tensor model is

$$\min_{d\in R^n} \quad m_T(x_c + d) \tag{6.3.65}$$
$$\text{s.t.} \quad \|d\|_2 \le \Delta_c, \tag{6.3.66}$$

where $\Delta_c \in R$ is the trust-region radius.

The tensor algorithm for unconstrained optimization is as follows.

Algorithm 6.3.6 *(Tensor Method) Given $x_c, f(x_c), \Delta_c$.*

Step 1. Calculate $\nabla f(x_c)$, and decide whether to stop. If not, go to Step 2.

Step 2. Calculate $\nabla^2 f(x_c)$.

Step 3. Select p past points from among the $n^{1/3}$ most recent past points.

Step 4. Calculate T_c and V_c.

Step 5. Find a potential acceptable next iterate $x_c + d_T$ and a potential new trust-region radius Δ_T by using the tensor model and a trust-region technique.

Step 6. Find a potential acceptable next iterate $x_c + d_N$ and a potential new trust-region radius Δ_N by using the quadratic model and a trust-region technique.

Step 7. If $f(x_c + d_T) \le f(x_c + d_N)$, then set

$$x_+ = x_c + d_T, \ \Delta_+ = \Delta_T;$$

else set

$$x_+ = x_c + d_N, \ \Delta_+ = \Delta_N.$$

Step 8. Set $x_c = x_+$, $f(x_c) = f(x_+)$, $\Delta_c = \Delta_+$, and go to Step 1. □

Note that in the tensor method, the Hessian can be replaced by finite difference Hessian approximation or secant updates, because the cost of computing a Hessian is large. Also, we would like to point out that the tensor

method is a generalization of the standard quadratic model method. However, there are still various problems waiting for us to solve. For example, the strategy of computing both tensor model and quadratic model at each iteration is not ideal; how to choose a suitable p, how to use the tensor method in constrained problems and so on. This kind of method is worth doing further study.

Exercises

1. Let $f(x) = x_1^4 + x_1^2 + x_2^2$. Let the current iterate $x^{(k)} = (1,1)^T, \Delta_k = \frac{1}{2}$. Try using double-dogleg method to find $x^{(k+1)}$.

2. Let $f(x) = \frac{1}{2}x_1^2 + x_2^2$. Let the starting point $x^{(0)} = (1,1)^T$. For $\Delta_0 = 1$ and $\Delta_0 = \frac{5}{4}$,
(1) Use dogleg method to find $x^{(1)}$.
(2) Use double-dogleg method to find $x^{(2)}$.

3. Let s_k be an approximate solution of subproblem (6.1.1). Show that s_k satisfies

$$q^{(k)}(0) - q^{(k)}(s_k) \geq \beta \|g_k\|_2 \min\left\{\Delta_k, \frac{\|g_k\|_2}{\|B_k\|_2}\right\},$$

where $\beta \in (0,1]$.

4. What is the attractive point of the trust-region method?

5. Use trust-region Newton method to minimize the Rosenbrock function (see Appendix: Problem 1.1).

6. Use trust-region quasi-Newton method to minimize the extended Rosenbrock function (see Appendix: Problem 1.2).

7. Consider using dogleg method to construct path $s(\tau)$. Show that $\|s(\tau)\|$ increases monotonically along this path.

8. Derive expression (6.1.30)–(6.1.31) of the Cauchy point.

9. (1) Let s_k^G solve

$$\begin{array}{ll} \min & f(x_k) + g_k^T s \\ \text{s.t.} & \|Ds\| \leq \Delta_k. \end{array}$$

Show that

$$s_k^G = -\frac{\Delta_k}{\|D^{-1}g_k\|_2} D^{-2} g_k.$$

(2) The generalized Cauchy point can be defined by

$$q^{(k)}(s_k^c) = \min\{q^{(k)}(s) \mid s = \tau s_k^G, \ \|Ds\| \leq \Delta_k\},$$

where s_k^G is defined by (1). Therefore, the generalized Cauchy point can be expressed as

$$s_k^c = \tau_k s_k^G = -\tau_k \frac{\Delta_k}{\|D^{-1}g_k\|_2} D^{-2} g_k, \tag{6.3.67}$$

where

$$\tau_k = \arg\min_{\tau > 0} q^{(k)}(\tau s_k^G) \quad \text{s.t. } \|\tau D s_k^G\| \leq \Delta_k.$$

Show:

$$\tau_k = \begin{cases} 1 & \text{if } g_k^T D^{-2} B_k D^{-2} g_k \leq 0; \\ \min\{\|D^{-1}g_k\|_2^3 / (\Delta_k g_k^T D^{-2} B_k D^{-2} g_k), \ 1\} & \text{otherwise.} \end{cases} \tag{6.3.68}$$

10. Mimic Theorem 6.1.2, state and prove the necessary and sufficient condition that s^* is the solution of subproblem

$$\begin{array}{ll} \min_s & f + g^T s + \frac{1}{2} s^T B s \\ \text{s.t.} & \|Ds\|_2 \leq \Delta. \end{array}$$

11. Write out the program of Steihaug-CG Algorithm 6.1.12.

12. Try to state the relations among quadratic model, conic model, tensor model and collinear scaling.

13. Starting from collinear scaling $s = \frac{w}{1+h^T w}$, derive a generalized quasi-Newton equation.

14. Derive the collinear scaling BFGS formula. Try to derive other formulas of collinear scaling.

Chapter 7

Solving Nonlinear Least-Squares Problems

7.1 Introduction

This chapter is devoted to solving the following nonlinear least-squares problems:

$$\min_{x \in R^n} f(x) = \frac{1}{2} r(x)^T r(x) = \frac{1}{2} \sum_{i=1}^{m} [r_i(x)]^2, \quad m \geq n \tag{7.1.1}$$

where $r : R^n \to R^m$ is a nonlinear function of x. If $r(x)$ is a linear function, the problem (7.1.1) is the linear least-squares problem.

Nonlinear least-squares problem (7.1.1) can be regarded as a special case for unconstrained minimization with a special structure. This problem can also be interpreted as solving the system of m nonlinear equations

$$r_i(x) = 0, \ i = 1, 2, \cdots, m, \tag{7.1.2}$$

where $r_i(x)$ is called the residual function. When $m > n$, the system is called over-determined, and when $m = n$ the system is well-determined.

Nonlinear least-squares problems have wide applications in data fitting, parameter estimation, function approximation, and others. For example, suppose we are given the data $(t_1, y_1), (t_2, y_2), \cdots, (t_m, y_m)$ and want to fit a function $\phi(t, x)$ which is a nonlinear function of x. We want to choose x such that the function $\phi(t, x)$ fits the data as well as possible in the sense of

minimizing the sum of the squares of the residual,

$$\min \sum_{i=1}^{m} [r_i(x)]^2 \tag{7.1.3}$$

where

$$r_i(x) = \phi(t_i, x) - y_i, \ i = 1, \cdots, m \tag{7.1.4}$$

are the residual. Usually, $m \gg n$. So, we obtain the problem (7.1.1). For solving nonlinear least-squares problem, we usually use Newton's method to solve the resulting system of the normal equations. However, it is expensive, and the normal equations tend easily to be ill-conditioned. Note that the problem (7.1.1) has special structure which inspires some special methods. In this chapter, we shall give some effective and special methods for solving nonlinear least-squares problem (7.1.1).

Let $J(x)$ be the Jacobian of $r(x)$,

$$J(x) = \begin{bmatrix} \frac{\partial r_1}{\partial x_1}(x) & \frac{\partial r_1}{\partial x_2}(x) & \cdots & \frac{\partial r_1}{\partial x_n}(x) \\ \frac{\partial r_2}{\partial x_1}(x) & \frac{\partial r_2}{\partial x_2}(x) & \cdots & \frac{\partial r_2}{\partial x_n}(x) \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial r_m}{\partial x_1}(x) & \frac{\partial r_m}{\partial x_2}(x) & \cdots & \frac{\partial r_m}{\partial x_n}(x) \end{bmatrix}. \tag{7.1.5}$$

Then the gradient of $f(x)$ is

$$g(x) = \sum_{i=1}^{m} r_i(x) \nabla r_i(x) = J(x)^T r(x) \tag{7.1.6}$$

and the Hessian is

$$\begin{aligned} G(x) &= \sum_{i=1}^{m} (\nabla r_i(x) \nabla r_i(x)^T + r_i(x) \nabla^2 r_i(x)) \\ &= J(x)^T J(x) + S(x), \end{aligned} \tag{7.1.7}$$

where

$$S(x) = \sum_{i=1}^{m} r_i(x) \nabla^2 r_i(x). \tag{7.1.8}$$

Therefore, the quadratic model of the objective function $f(x)$ is

$$\begin{aligned} q^{(k)}(x) &= f(x_k) + g(x_k)^T (x - x_k) + \frac{1}{2}(x - x_k)^T G(x_k)(x - x_k) \\ &= \frac{1}{2} r(x_k)^T r(x_k) + (J(x_k)^T r(x_k))^T (x - x_k) \\ &\quad + \frac{1}{2}(x - x_k)^T (J(x_k)^T J(x_k) + S(x_k))(x - x_k). \end{aligned} \tag{7.1.9}$$

Then we have Newton's method for (7.1.1),

$$x_{k+1} = x_k - (J(x_k)^T J(x_k) + S(x_k))^{-1} J(x_k)^T r(x_k). \tag{7.1.10}$$

We have seen in Chapter 3 that, under standard assumptions, the iteration (7.1.10) is locally quadratically convergent. However, the main disadvantage of the above Newton's method is that the second-order term $S(x)$ in the Hessian $G(x)$ is difficult or expensive to compute. It is also not suitable to use a secant approximation of the whole of $G(x)$, because $J(x)$ and furthermore the first-order term $J(x)^T J(x)$ in $G(x)$ are available when we compute the gradient $g(x)$. Hence, for reducing the computation, it may be reasonable and effective to either neglect $S(x)$ or use first-order derivative information to approximate $S(x)$. Notice from (7.1.8) that when $r_i(x)$ approaches zero or closes to a linear function, in which case $\nabla^2 r_i(x)$ approaches zero, $S(x)$ is small and can be neglected. We call this case a small residual problem, otherwise, a large residual problem.

7.2 Gauss-Newton Method

In this section, we discuss the Gauss-Newton method, which is obtained by neglecting the second-order term $S(x)$ of $G(x)$ in the quadratic model (7.1.9). So, (7.1.9) becomes

$$\begin{aligned} \bar{q}^{(k)}(x) &= \frac{1}{2} r(x_k)^T r(x_k) + (J(x_k)^T r(x_k))^T (x - x_k) \\ &\quad + \frac{1}{2} (x - x_k)^T (J(x_k)^T J(x_k))(x - x_k). \end{aligned} \tag{7.2.1}$$

Hence (7.1.10) becomes

$$x_{k+1} = x_k + s_k = x_k - (J(x_k)^T J(x_k))^{-1} J(x_k)^T r(x_k). \tag{7.2.2}$$

To make the iteration well-defined, it is required that Jacobian matrix $J(x_k)$ has full column rank. The following is the Gauss-Newton algorithm.

Algorithm 7.2.1 *(Gauss-Newton Method)*

Step 0. Given $x_0, \epsilon > 0, k := 0$.

Step 1. If $\|g_k\| \leq \epsilon$, *stop.*

Step 2 Solve

$$J(x_k)^T J(x_k)s = -J(x_k)^T r(x_k) \quad \textit{for } s_k. \tag{7.2.3}$$

Step 3. Set $x_{k+1} = x_k + s_k$, $k := k + 1$. *Go to Step 1.* □

Obviously, whenever $J(x_k)$ has full rank and the gradient $g(x_k)$ is nonzero, the search direction s_k is a descent direction for f, because

$$s_k^T \nabla f(x_k) = s_k^T J(x_k)^T r(x_k) = -s_k^T J(x_k)^T J(x_k) s_k \leq 0.$$

The final inequality is strict unless $J(x_k)^T s_k = 0$, which is equivalent to $J(x_k)^T r(x_k) = g(x_k) = 0$.

Equation (7.2.3) is said to be the Gauss-Newton equation. Obviously, by comparing (7.2.3) and (7.1.10), we find that the difference between Gauss-Newton method and Newton method is that the first-order term $J(x_k)^T J(x_k)$ is used to replace the Hessian $G(x_k)$.

Note that Step 2 in Algorithm 7.2.1 is just analogous to the normal equations of linear least-squares problem. Besides, the model (7.2.1) is equivalent to considering the affine model of $r(x)$ near x_k,

$$\bar{M}_k = r(x_k) + J(x_k)(x - x_k), \tag{7.2.4}$$

and solve the linear least-squares problem

$$\min \frac{1}{2}\|\bar{M}_k(x)\|^2. \tag{7.2.5}$$

These two observations expose that Gauss-Newton method, in fact, is a linearization method for nonlinear least-squares problem. From (7.2.2), we see that Gauss-Newton method has some advantages in that it only requires the first-order derivative information of the residual function $r(x)$, and that $J(x)^T J(x)$ is at least positive semi-definite.

Since Newton's method, under the standard assumptions, is locally and quadratically convergent, the success of Gauss-Newton method will depend on the importance of the neglected second-order term $S(x)$ in $G(x)$. The following theorem shows:

1. if $S(x^*) = 0$, the Gauss-Newton method is quadratically convergent;
2. if $S(x^*)$ is small relative to $J(x^*)^T J(x^*)$, the Gauss-Newton method is locally Q-linearly convergent;

3. if $S(x^*)$ is too large, the Gauss-Newton method will not be convergent.

The proofs of the following theorem are similar to that of Theorem 3.2.2 for Newton's method. The different proofs given in Theorem 7.2.2 and Theorem 7.2.3 are helpful to study and understand the convergence theorems of various iterative methods.

Theorem 7.2.2 *Let $f : R^n \to R$ and $f \in C^2$. Assume that x^* is the local minimizer of the nonlinear least-squares problem (7.1.1) and $J(x^*)^T J(x^*)$ is positive definite. Assume also that the sequence $\{x_k\}$ generated by Algorithm 7.2.1 converges to x^*. Then, if $G(x)$ and $(J(x)^T J(x))^{-1}$ are Lipschitz continuous in the neighborhood of x^*, we have*

$$\|x_{k+1} - x^*\| \leq \|(J(x^*)^T J(x^*))^{-1}\| \|S(x^*)\| \|x_k - x^*\| + O(\|x_k - x^*\|^2). \quad (7.2.6)$$

Proof. Since $G(x)$ is Lipschitz continuous, $J(x)^T J(x)$ and $S(x)$ are also Lipschitz continuous. Hence, there exist $\alpha, \beta, \gamma > 0$, such that for any x, y in the neighborhood of x^*, we have

$$\|J(x)^T J(x) - J(y)^T J(y)\| \leq \alpha \|x - y\|, \quad (7.2.7)$$
$$\|S(x) - S(y)\| \leq \beta \|x - y\|, \quad (7.2.8)$$
$$\|(J(x)^T J(x))^{-1} - (J(y)^T J(y))^{-1}\| \leq \gamma \|x - y\|, \quad (7.2.9)$$

(see Exercise).

Since $f \in C^2$ and $G(x)$ is Lipschitz continuous, then we have

$$g(x_k + s) = g(x_k) + G(x_k)s + O(\|s\|^2). \quad (7.2.10)$$

Let $h_k = x_k - x^*$ and $s = -h_k$. We can deduce that

$$0 = g(x^*) = g(x_k) - G(x_k)h_k + O(\|h_k\|^2). \quad (7.2.11)$$

Substituting (7.1.6) and (7.1.7) into (7.2.11) gives

$$J(x_k)^T r(x_k) - (J(x_k)^T J(x_k) + S(x_k))h_k + O(\|h_k\|^2) = 0. \quad (7.2.12)$$

Assume that x_k is in a neighborhood of x^*. From Theorem 1.2.5, it follows that for k sufficiently large, $J(x_k)^T J(x_k)$ is positive definite, and hence $(J(x_k)^T J(x_k))^{-1}$ is bounded above and

$$\|(J(x_k)^T J(x_k))^{-1}\| \leq 2\|(J(x^*)^T J(x^*))^{-1}\|. \quad (7.2.13)$$

Then, multiplying (7.2.12) by $(J(x_k)^T J(x_k))^{-1}$ yields that

$$-s_k - h_k - (J(x_k)^T J(x_k))^{-1} S(x_k) h_k + O(\|h_k\|^2) = 0. \tag{7.2.14}$$

Note that $s_k + h_k = x_{k+1} - x^* = h_{k+1}$, the above equality can be written as

$$\begin{aligned} &-h_{k+1} - (J(x^*)^T J(x^*))^{-1} S(x^*) - (J(x_k)^T J(x_k))^{-1} (S(x_k) - S(x^*)) h_k \\ &-[(J(x_k)^T J(x_k))^{-1} - (J(x^*)^T J(x^*))^{-1}] S(x^*) h_k + O(\|h_k\|^2) \\ &= 0. \end{aligned} \tag{7.2.15}$$

Taking the norm and using (7.2.8)–(7.2.9) and (7.2.13) give the result (7.2.6). □

Theorem 7.2.3 *Let $f : D \subset R^n \to R$ and $f \in C^2(D)$, where D is an open convex set. Let $J(x)$ be Lipschitz continuous on D, i.e.,*

$$\|J(x) - J(y)\|_2 \le \gamma \|x - y\|_2, \ \forall x, y \in D, \tag{7.2.16}$$

and $\|J(x)\|_2 \le \alpha, \forall x \in D$. Assume that there exist $x^* \in D$ and $\lambda, \sigma \ge 0$ such that $J(x^*)^T r(x^*) = 0$, λ is the smallest eigenvalue of $J(x^*)^T J(x^*)$, and

$$\|(J(x) - J(x^*))^T r(x^*)\|_2 \le \sigma \|x - x^*\|_2, \ \forall x \in D. \tag{7.2.17}$$

If $\sigma < \lambda$, then, for any $c \in (1, \lambda/\sigma)$, there exists $\epsilon > 0$ such that for all $x_0 \in N(x^*, \epsilon)$, the sequence generated by Gauss-Newton Algorithm 7.2.1 is well-defined, converges to x^*, and satisfies

$$\|x_{k+1} - x^*\|_2 \le \frac{c\sigma}{\lambda} \|x_k - x^*\|_2 + \frac{c\alpha\gamma}{2\lambda} \|x_k - x^*\|_2^2 \tag{7.2.18}$$

and

$$\|x_{k+1} - x^*\|_2 \le \frac{c\sigma + \lambda}{2\lambda} \|x_k - x^*\|_2 < \|x_k - x^*\|_2. \tag{7.2.19}$$

Proof. By induction. For convenience, let J_0, r_0, r^* denote $J(x_0), r(x_0)$ and $r(x^*)$. From Theorem 1.2.5, it follows that there exists $\epsilon_1 > 0$ such that $J_0^T J_0$ is nonsingular and satisfies

$$\|(J_0^T J_0)^{-1}\| \le c/\lambda, \ \text{ for } x_0 \in N(x^*, \epsilon_1). \tag{7.2.20}$$

Let

$$\epsilon = \min \left\{ \epsilon_1, \frac{\lambda - c\sigma}{c\alpha\gamma} \right\}, \tag{7.2.21}$$

where γ is the Lipschitz constant defined in (7.2.16). Then, x_1 is well-defined at the first iteration, and we have

$$\begin{aligned} x_1 - x^* &= x_0 - x^* - (J_0^T J_0)^{-1} J_0^T r_0 \\ &= -(J_0^T J_0)^{-1}[J_0^T r_0 + J_0^T J_0(x^* - x_0)] \\ &= -(J_0^T J_0)^{-1}[J_0^T r^* - J_0^T(r^* - r_0 - J_0(x^* - x_0))]. \end{aligned} \tag{7.2.22}$$

By Theorem 1.2.22, we have

$$\|r^* - r_0 - J_0(x^* - x_0)\| \leq \frac{\gamma}{2}\|x_0 - x^*\|^2. \tag{7.2.23}$$

Noting that $J(x^*)^T r(x^*) = 0$ and using (7.2.17), we get

$$\|J_0^T r^*\| = \|(J_0 - J(x^*))^T r^*\| \leq \sigma\|x - x^*\|. \tag{7.2.24}$$

By using (7.2.20), (7.2.24), (7.2.23) and $\|J_0\| \leq \alpha$, it follows from (7.2.22) that

$$\begin{aligned} \|x_1 - x^*\| &\leq \|(J_0^T J_0)^{-1}\|(\|J_0^T r^*\| + \|J_0\|\|r^* - r_0 - J_0(x^* - x_0)\|) \\ &\leq \frac{c}{\lambda}\left(\sigma\|x_0 - x^*\| + \frac{\alpha\gamma}{2}\|x_0 - x^*\|^2\right). \end{aligned} \tag{7.2.25}$$

This proves that (7.2.18) holds at $k = 0$.

Furthermore, from (7.2.25) and (7.2.21), we deduce that

$$\begin{aligned} \|x_1 - x^*\| &\leq \|x_0 - x^*\|\left(\frac{c\sigma}{\lambda} + \frac{c\alpha\gamma}{2\lambda}\|x_0 - x^*\|\right) \\ &\leq \|x_0 - x^*\|\left(\frac{c\sigma}{\lambda} + \frac{\lambda - c\sigma}{2\lambda}\right) \\ &= \frac{c\sigma + \lambda}{2\lambda}\|x_0 - x^*\| \\ &< \|x_0 - x^*\|, \end{aligned} \tag{7.2.26}$$

which shows that (7.2.19) holds at $k = 0$.

For the general case of k, the proof is the same completely as the above. Hence, we complete the proof by induction. □

Theorem 7.2.4 *Assume that the assumptions of Theorem 7.2.2 or Theorem 7.2.3 are satisfied. If $r(x^*) = 0$, then there exists $\epsilon > 0$ such that for any $x_0 \in N(x^*, \epsilon)$, the sequence $\{x_k\}$ generated by Gauss-Newton method converges to x^* with quadratic convergence rate.*

Proof. For Theorem 7.2.2, if $r(x^*) = 0$, then $S(x^*) = 0$. So, the quadratic convergence rate is obtained immediately from (7.2.6).

For Theorem 7.2.3, if $r(x^*) = 0$, then the σ in (7.2.17) can be taken as $\sigma = 0$. Hence, it follows from (7.2.19) that $\{x_k\}$ converges to x^*, and from (7.2.18) that the rate is quadratic. □

Gauss-Newton method now is the most basic method for solving nonlinear least-squares problems. The following example demonstrates that it works well with small residual problems.

Example 7.2.5 *Let* $r_1(x) = x + 1, r_2(x) = \lambda x^2 + x - 1$. *Consider*

$$\min f(x) = \sum_{i=1}^{2} r_i(x)^2 = (x+1)^2 + (\lambda x^2 + x - 1)^2,$$

where $n = 1, m = 2$, *and* $x^* = 0$. *For* $\lambda = 0.1$, *the Gauss-Newton iteration has the following result:*

k	*1*	*2*	*3*	*4*	*5*	*6*
x_k	*1.000000*	*0.131148*	*0.013635*	*0.001369*	*0.000137*	*0.000014*

You can see that, when $\lambda = 0.1$, the degree of nonlinearity in $r(x)$ is small, and the Gauss-Newton method works well. In this case, from (7.2.2), the Gauss-Newton iteration is

$$x_{k+1} = \frac{2\lambda^2 x_k^3 + \lambda x_k^2 + 2\lambda x_k}{1 + (2\lambda x_k + 1)^2}.$$

When $\lambda = 0$, in which case $r(x)$ is linear, then $x_1 = 0 = x^*$. This indicates that Gauss-Newton method gets its minimizer in one iteration. When $\lambda \neq 0$, we have

$$x_{k+1} = \lambda x_k + O(\|x_k\|^2).$$

When λ is small enough, the convergence rate is linear. When $|\lambda| > 1$, the Gauss-Newton method fails to converge. This example shows that Gauss-Newton method is valuable only when both x_0 closes to x^* and the matrix $S(x^*)$ is small.

Remark: In practice, we usually use Gauss-Newton method with line search

$$x_{k+1} = x_k - \alpha_k (J(x_k)^T J(x_k))^{-1} J(x_k)^T r(x_k), \tag{7.2.27}$$

which is called the damped Gauss-Newton method, where α_k is a step size. As we have seen, this method guarantees the descent of the objective function in each step and therefore global convergence.

To conclude this section, we mention some numerical aspects on Gauss-Newton method. It should be pointed out that for the problem to solve Gauss-Newton equations

$$J(x_k)^T J(x_k)s = -J(x_k)^T r(x_k), \tag{7.2.28}$$

usually, we employ matrix factorization instead of solving (7.2.28) directly. Then the solution is found by back-substitution technique. So, we can substantially improve the numerical precision. To see this, it follows from the error analysis that

$$\frac{\|\delta s\|}{\|s\|} \leq \kappa(J(x_k)^T J(x_k)) \frac{\|E\|}{\|J(x_k)^T J(x_k)\|}, \tag{7.2.29}$$

where

$$\kappa(J(x_k)^T J(x_k)) = \sigma_1^2/\sigma_n^2, \tag{7.2.30}$$

δs and E denote the errors of s and $J(x_k)^T J(x_k)$ respectively, and σ_1 and σ_n are the largest and smallest singular values of $J(x_k)$ respectively.

If we employ QR factorization for the augmented matrix, we have

$$[J(x_k) \quad r_k] = Q[R \quad Q^T r_k], \tag{7.2.31}$$

where Q is an orthogonal matrix,

$$R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix},$$

and R_1 is a nonsingular upper triangular matrix. Then, we obtain

$$J(x_k)^T J(x_k) = R^T R = R_1^T R_1. \tag{7.2.32}$$

The solution of (7.2.28) can be found by solving

$$R_1 s = -(Q^T r_k)_n, \tag{7.2.33}$$

where $(\cdot)_n$ denotes the first n element partition.

It can be shown that

$$\kappa(R_1) = \frac{\sigma_1}{\sigma_n} \tag{7.2.34}$$

and therefore the errors generated by the orthogonal factorization approach are magnified much less than that directly solve (7.2.28).

7.3 Levenberg-Marquardt Method

7.3.1 Motivation and Properties

Usually, Gauss-Newton method with line search is employed in practice. However, if $J(x)$ is rank-deficient, then either the Gauss-Newton method cannot work well, or the algorithm converges to a non-stationary point.

To overcome the difficulty, we consider employing trust-region technique (for details, see §6.1). In fact, we have seen that, in Gauss-Newton method, a linearized model (7.2.4) is used to replace nonlinear function $r(x)$, and that a linear least-squares problem (7.2.5) is obtained. Unfortunately, this linearization is not effective for all $(x - x_k)$. Therefore, we put a constraint of trust-region on it, and consider the following trust-region model:

$$\min \quad \frac{1}{2}\|J(x_k)(x - x_k) + r(x_k)\|_2^2 \tag{7.3.1}$$

$$\text{s.t.} \quad \|x - x_k\|_2 \le \Delta_k, \tag{7.3.2}$$

which is a constrained linear least-squares problem. Model (7.3.1)-(7.3.2) can be written as

$$\begin{aligned} \min \quad & q_k(x) = \frac{1}{2}\|r_k\|^2 + r_k^T J(x_k)(x - x_k) + \frac{1}{2}(x - x_k)^T J(x_k)^T J(x_k)(x - x_k) \\ \text{s.t.} \quad & \|x - x_k\|_2 \le \Delta_k. \end{aligned} \tag{7.3.3}$$

Set $s = x - x_k$. The solution of the subproblem (7.3.1)-(7.3.2) is characterized by solving the system

$$(J(x_k)^T J(x_k) + \mu_k I)s = -J(x_k)^T r(x_k). \tag{7.3.4}$$

Hence,

$$x_{k+1} = x_k - (J(x_k)^T J(x_k) + \mu_k I)^{-1} J(x_k)^T r(x_k). \tag{7.3.5}$$

When $\|(J(x_k)^T J(x_k))^{-1} J(x_k)^T r(x_k)\| \le \Delta_k$, then $\mu_k = 0$ and the subproblem is solved by s_k. Otherwise, there exists $\mu_k > 0$ such that the solution s_k satisfying $\|s_k\| = \Delta_k$ and

$$(J(x_k)^T J(x_k) + \mu_k I)s_k = -J(x_k)^T r(x_k). \tag{7.3.6}$$

Since $(J(x_k)^T J(x_k) + \mu_k I)$ is positive definite, the direction s produced by (7.3.4) is a descent direction. This method is called the Levenberg-Marquardt method, since it was proposed by Levenberg [199] and Marquardt [210]. The

above discussion exposes that the Levenberg-Marquardt method is just a Gauss-Newton method by replacing the line search with a trust region strategy.

Another perspective about Levenberg-Marquardt method is as follows. This method is just a switch rule between Gauss-Newton method and the steepest descent method. This implies that this method allows choosing any direction between these two directions to be a search direction. When $\mu_k = 0$, it reduces to the Gauss-Newton direction. While μ_k is very large, (7.3.4) is approximate to

$$\mu_k I s = -J(x_k)^T r(x_k). \tag{7.3.7}$$

The produced direction is close to the steepest descent direction.

Furthermore, if, instead of I, we employ some positive definite and diagonal matrix D_k, then (7.3.4) becomes

$$(J(x_k)^T J(x_k) + \mu_k D_k)s = -J(x_k)^T r(x_k). \tag{7.3.8}$$

In this case, the produced direction is a combination of Gauss-Newton direction and the steepest descent direction with respect to a metric matrix D_k.

Next, we will describe some properties of Levenberg-Marquardt method.

Let $s = s(\mu)$ be a solution of

$$(J^T J + \mu I)s = -J^T r, \tag{7.3.9}$$

where, for convenience, $J = J(x), r = r(x)$, and $g = g(x) = J^T r$.

Theorem 7.3.1 *When μ increases monotonically from zero, $\|s(\mu)\|$ will decrease strictly monotonically.*

Proof.

$$\frac{\mathrm{d}}{\mathrm{d}\mu}\|s\| = \frac{\mathrm{d}}{\mathrm{d}\mu}(s^T s)^{\frac{1}{2}} = \frac{s^T \frac{\mathrm{d}s}{\mathrm{d}\mu}}{\|s\|}. \tag{7.3.10}$$

Differentiating (7.3.9) gives

$$(J^T J + \mu I)\frac{\mathrm{d}s}{\mathrm{d}\mu} = -s. \tag{7.3.11}$$

It follows from (7.3.11) and (7.3.9) that

$$\frac{\mathrm{d}s}{\mathrm{d}\mu} = (J^T J + \mu I)^{-2} g. \tag{7.3.12}$$

By substituting (7.3.12) into (7.3.10) and using (7.3.9), we obtain

$$\frac{\mathrm{d}}{\mathrm{d}\mu}\|s\| = -\frac{g^T(J^TJ+\mu I)^{-3}g}{\|s\|}. \tag{7.3.13}$$

When $\mu \geq 0$, $J^TJ + \mu I$ is positive definite. Therefore (7.3.13) shows that $\|s(\mu)\|$ decreases strictly monotonically. □

Theorem 7.3.2 *The angle ψ between s and $-g$ does not increase monotonically as μ increases.*

Proof. Since

$$\cos\psi = -\frac{g^Ts}{\|g\|\|s\|}, \tag{7.3.14}$$

then it is enough to prove

$$\frac{\mathrm{d}}{\mathrm{d}\mu}\cos\psi \geq 0.$$

By using (7.3.9)-(7.3.13), we deduce

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\mu}(\cos\psi) &= \frac{-g^T\frac{\mathrm{d}s}{\mathrm{d}\mu}}{\|g\|\|s\|} + \frac{g^Ts}{\|g\|\|s\|}\frac{\frac{\mathrm{d}\|s\|}{\mathrm{d}\mu}}{\|s\|} \\ &= \frac{1}{\|g\|\|s\|^3}\{-(g^T(J^TJ+\mu I)^{-2}g)^2 \\ &\quad +(g^T(J^TJ+\mu I)^{-1}g)(g^T(J^TJ+\mu I)^{-3}g)\}. \end{aligned} \tag{7.3.15}$$

So, it is enough to prove that the part in braces is greater than or equal to zero.

Note that J^TJ is symmetric, then there is an orthogonal matrix Q such that

$$J^TJ = Q^TDQ,$$

where $D = \mathrm{diag}(\lambda_1, \cdots, \lambda_n)$. Set $v = Qg$, then the part in braces on the right-hand-side of (7.3.15) can be written as

$$\sum_{j=1}^{n}\sum_{k=1}^{n}\left\{-\frac{v_j^2v_k^2}{(\lambda_j+\mu)^2(\lambda_k+\mu)^2} + \frac{v_j^2v_k^2}{(\lambda_j+\mu)(\lambda_k+\mu)^3}\right\}. \tag{7.3.16}$$

If let

$$a = \left(\frac{v_1}{(\lambda_1+\mu)^{3/2}}, \cdots, \frac{v_n}{(\lambda_n+\mu)^{3/2}}\right)^T,$$
$$b = \left(\frac{v_1}{(\lambda_1+\mu)^{1/2}}, \cdots, \frac{v_n}{(\lambda_n+\mu)^{1/2}}\right)^T,$$

then (7.3.16) becomes

$$\|a\|^2\|b\|^2 - \langle a, b\rangle^2,$$

and it follows from Schwartz inequality that the above expression is greater than or equal to zero. Therefore ψ is not increasing. We complete the proof. □

Theorem 7.3.3 *Let $\mu_k > 0$ and s_k be a solution of (7.3.4). Then s_k is a global solution of the subproblem*

$$\min \quad q^{(k)}(s) = \frac{1}{2}\|J_k s + r_k\|_2^2 \tag{7.3.17}$$
$$\text{s.t.} \quad \|s\| \le \|s_k\|. \tag{7.3.18}$$

Proof. Since s_k is a solution of (7.3.4), then

$$\begin{aligned} q^{(k)}(s_k) &= \frac{1}{2}r_k^T r_k + r_k^T J_k s_k + \frac{1}{2}s_k^T J_k^T J_k s_k \\ &= \frac{1}{2}r_k^T r_k - s_k^T(J_k^T J_k + \mu_k I)s_k + \frac{1}{2}s_k^T J_k^T J_k s_k \\ &= \frac{1}{2}r_k^T r_k - \mu_k s_k^T s_k - \frac{1}{2}s_k^T J_k^T J_k s_k. \end{aligned} \tag{7.3.19}$$

On the other hand, for any s, we have

$$\begin{aligned} q^{(k)}(s) &= \frac{1}{2}r_k^T r_k + s^T J_k^T r_k + \frac{1}{2}s^T J_k^T J_k s \\ &= \frac{1}{2}r_k^T r_k - s^T(J_k^T J_k + \mu_k I)s_k + \frac{1}{2}s^T J_k^T J_k s \\ &= \frac{1}{2}r_k^T r_k - \mu_k s^T s_k - s^T J_k^T J_k s_k + \frac{1}{2}s^T J_k^T J_k s. \end{aligned} \tag{7.3.20}$$

Then, for any s satisfying $\|s\| \le \|s_k\|$, we deduce that

$$\begin{aligned} q^{(k)}(s) - q^{(k)}(s_k) &= \frac{1}{2}(s_k - s)^T J_k^T J_k(s_k - s) + \mu_k(s_k^T s_k - s^T s_k) \\ &\ge \frac{1}{2}(s_k - s)^T J_k^T J_k(s_k - s) + \mu_k\|s_k\|(\|s_k\| - \|s\|) \\ &\ge 0, \end{aligned} \tag{7.3.21}$$

which shows that s_k is a global optimal solution of problem (7.3.17)-(7.3.18). □

Theorem 7.3.4 *The vector s_k is a solution of problem (7.3.1)-(7.3.2), i.e.,*

$$\min \quad \frac{1}{2}\|J_k s + r_k\|_2^2 \tag{7.3.22}$$

$$\text{s.t.} \quad \|s\| \leq \Delta_k \tag{7.3.23}$$

for some $\Delta_k > 0$ if and only if there exists $\mu \geq 0$ such that

$$(J_k^T J_k + \mu I)s_k = -J_k^T r_k, \tag{7.3.24}$$

$$\mu(\Delta_k - \|s_k\|) = 0, \tag{7.3.25}$$

$$\|s_k\| \leq \Delta_k. \tag{7.3.26}$$

Proof. It is obtained directly from Theorem 6.1.2. □

Usually, Levenberg-Marquardt method is characterized by the equation

$$(J(x_k)^T J(x_k) + \mu_k D(x_k))s = -J(x_k)^T r(x_k), \tag{7.3.27}$$

where $D(x_k)$ is a diagonal and positive definite matrix. The steplength factor α_k satisfies Armijio rule (2.5.3):

$$f(x_k + \alpha_k s_k) \leq f(x_k) + \sigma\alpha_k g_k^T s_k, \quad \sigma \in \left(0, \frac{1}{2}\right). \tag{7.3.28}$$

Theorem 7.3.5 *For (7.3.27), the condition number of $J(x)^T J(x) + \mu D(x)$ is a non-increasing function of μ.*

Proof. Let β_1 and β_n be the largest and smallest eigenvalues of $D(x)$ respectively. Let λ_1 and λ_n be the largest and smallest eigenvalues of $J(x)^T J(x) + \mu D(x)$ respectively. Let also $\mu_1 > \mu_2 \geq 0$. Since the range of a normal matrix is a convex hull of its spectrum, we have

$$\begin{aligned}
\frac{\lambda_1(\mu_1)}{\lambda_n(\mu_1)} &\leq \frac{\lambda_1(\mu_2) + (\mu_1 - \mu_2)\beta_1}{\lambda_n(\mu_2) + (\mu_1 - \mu_2)\beta_n} \\
&\leq \frac{\lambda_1(\mu_2) + (\mu_1 - \mu_2)(1 + \mu_2)^{-1}\lambda_1(\mu_2)}{\lambda_n(\mu_2) + (\mu_1 - \mu_2)(1 + \mu_2)^{-1}\lambda_n(\mu_2)} \\
&= \frac{\lambda_1(\mu_2)}{\lambda_n(\mu_2)}.
\end{aligned}$$

Therefore, the conclusion is obtained. □

This property indicates that the Levenberg-Marquardt method improves the condition of the equations solved.

7.3.2 Convergence of Levenberg-Marquardt Method

In this subsection we establish convergence of the Levenberg-Marquardt method.

Theorem 7.3.6 *Let $\{x_k\}$ be a sequence produced by Levenberg-Marquardt method (7.3.27). Suppose that the step lengths α_k are determined by Armijo rule (7.3.28). If there is a subsequence $\{x_{k_i}\}$ that converges to x^*, and if the corresponding subsequence $\{J_{k_i}^T J_{k_i} + \mu_{k_i} D_{k_i}\}$ converges to some positive definite matrix P, where $J_{k_i} = J(x_{k_i})$ and $D_{k_i} = D(x_{k_i})$ denoting a diagonal positive definite matrix, then $g(x^*) = 0$.*

Proof. (By contradiction) Suppose that $g(x^*) \neq 0$. Let

$$s_{k_i} = -(J_{k_i}^T J_{k_i} + \mu_{k_i} D_{k_i})^{-1} J_{k_i}^T r_{k_i},$$
$$s^* = \lim s_{k_i} = -P^{-1} J(x^*)^T r(x^*),$$

where $r_{k_i} = r(x_{k_i})$. Obviously, $g(x^*)^T s^* < 0$. Let $\beta \in (0,1)$, $\sigma \in (0, \frac{1}{2})$. Let also m^* be the least non-negative integer m such that

$$f(x^* + \beta^m s^*) < f(x^*) + \sigma \beta^m g(x^*)^T s(x^*).$$

By continuity, for k sufficiently large, we have

$$f(x_{k_i} + \beta^{m^*} s_{k_i}) \leq f(x_{k_i}) + \sigma \beta^{m^*} g(x_{k_i})^T s_{k_i}.$$

Hence

$$f(x_{k_i+1}) = f(x_{k_i} + \beta^{m_{k_i}} s_{k_i}) \leq f(x_{k_i}) + \sigma \beta^{m^*} g(x_{k_i})^T s_{k_i}. \tag{7.3.29}$$

By the monotone descent of the method, we have

$$\lim f(x_{k_i+1}) = \lim f(x_{k_i}) = f(x^*).$$

Therefore, taking limits on both sides of (7.3.29) yields

$$f(x^*) \leq f(x^*) + \sigma \beta^{m^*} g(x^*)^T s^* < 0.$$

This is impossible because $\sigma \beta^{m^*} g(x^*)^T s^* < 0$. So we complete the proof. □

The above theorem states the convergence of a subsequence. Below, we give convergence of the whole sequence.

Theorem 7.3.7 *Suppose that the following assumptions hold:*

(a) *the level set*

$$L(\bar{x}) = \{x \mid f(x) \leq f(\bar{x})\}$$

is bounded and closed for any $\bar{x} \in R^n$;

(b) *the number of stationary points at which the function values of $f(x)$ are the same is finite;*

(c) *$J(x)^T J(x)$ is positive definite $\forall\, x$;*

(d) *$\mu_k \leq M < \infty$, $\forall\, k$, that is, M is an upper bound of μ_k.*

Then for any initial point x_0, the sequence $\{x_k\}$ generated from Levenberg-Marquardt method converges to a stationary point of $f(x)$.

Proof. From (a) and the monotone property of iterative function, we know that the sequence $\{x_k\}$ is in compact set $L(\bar{x})$. This shows that $\{x_k\}$ must have accumulation points. To prove the theorem, we only need to prove the accumulation points are unique.

By (c), (d) and Theorem 7.3.6, we have that each accumulation point of $\{x_k\}$ is unique. Since $\{f(x)\}$ is a monotone descent sequence, $f(x)$ has the same values at accumulation points of $\{x_k\}$. Also, from (b), it follows that the number of stationary points of f on $L(\bar{x})$ are finite. Therefore, there are only finitely many accumulation points.

Notice that, for some subsequence $\{x_{k_i}\}$, we have $x_{k_i} \to \hat{x}_k$ and

$$\lim_{k \to \infty} g(x_{k_i}) = g(\hat{x}_k) = 0.$$

Notice also that

$$s(\mu_{k_i}) = -(J(x_{k_i})^T J(x_{k_i}) + \mu_{k_i} D(x_{k_i}))^{-1} g(x_{k_i}).$$

Then it follows from (c) and (d) that $s(\mu_{k_i}) \to 0$. Therefore, for sequence $\{s(\mu_k)\}$, we have $s(\mu_k) \to 0$.

Assume for the moment that there are more than one accumulation point of $\{x_k\}$. Let ϵ^* be the smallest distance between any two accumulation points. Since $\{x_k\}$ is in a compact set, there exists a positive integer N, such that for all $k \geq N$, x_k is contained in a closed ball with some accumulation point

as center and $\epsilon^*/4$ as radius. On the other hand, there is an integer $N' \geq N$ such that

$$\|s(\mu_k)\| < \epsilon^*/4, \ \forall k \geq N'.$$

Therefore, when $k \geq N'$, all x_k are in the closed ball mentioned above with that accumulation point as center and $\epsilon^*/4$ as radius. Then we have a contradiction which proves the theorem. □

The above theorem establishes global convergence of the Levenberg-Marquardt method. In the following, similar to Theorem 7.2.2, we discuss the convergence rate of the Levenberg-Marquardt method.

Theorem 7.3.8 *Suppose that the iterates x_k generated by Levenberg-Marquardt method converge to a stationary point x^*. Let l be the smallest eigenvalue of $J(x^*)^T J(x^*)$, M the maximum of absolute values of eigenvalues of $S(x^*) = \sum_{i=1}^m r_i(x^*)\nabla^2 r_i(x^*)$. If*

$$\tau = M/l < 1, \ 0 < \beta < (1-\tau)/2, \ \mu_k \to 0, \tag{7.3.30}$$

then, for all k sufficiently large, the stepsize $\alpha_k = 1$,

$$\limsup \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \leq \tau, \tag{7.3.31}$$

and x^* is a strict local minimizer of $f(x)$.

Proof. We first prove $\alpha_k = 1$ for k large enough. Notice that

$$f(x_k + s_k) - f(x_k) = g_k^T s_k + \frac{1}{2} s_k^T G(x_k + \theta s_k) s_k, \tag{7.3.32}$$

where $\theta \in (0,1)$. By means of Armijo rule (7.3.28), to prove $\alpha_k = 1$ for k large enough, we need to show

$$\beta g_k^T s_k - [f(x_k + s_k) - f(x_k)] \geq 0. \tag{7.3.33}$$

By use of $g_k = -(J_k^T J_k + \mu_k D_k) s_k$ and (7.3.32), the left-hand side of (7.3.33) can be written as

$$\begin{aligned} & (1-\beta) s_k^T (J_k^T J_k + \mu_k D_k) s_k - \frac{1}{2} s_k^T G(x_k + \theta s_k) s_k \\ = \ & s_k^T \left[(1-\beta) J_k^T J_k - \frac{1}{2} G(x_k) + (1-\beta)\mu_k D_k \right. \\ & \left. - \frac{1}{2}(G(x_k + \theta s_k) - G(x_k)) \right] s_k \\ = \ & s_k^T \left[\left(\frac{1}{2} - \beta \right) J_k^T J_k - \frac{1}{2} S(x_k) + V_k \right] s_k, \end{aligned}$$

where $V_k = (1-\beta)\mu_k D_k - \frac{1}{2}(G(x_k+\theta s_k) - G(x_k))$, $S(x_k)$ is defined by (7.1.7). Since $V_k \to 0$, to prove (7.3.33) holds for k large enough, we show $(\frac{1}{2}-\beta)J_k^T J_k - \frac{1}{2}S(x_k)$ converges to a positive definite matrix. Note that the smallest eigenvalue of

$$\left(\frac{1}{2}-\beta\right) J(x^*)^T J(x^*) - \frac{1}{2}S(x^*)$$

is bounded below and that the lower bound is

$$\left(\frac{1}{2}-\beta\right) l - \frac{1}{2}M = l\left[\frac{1}{2}-\beta-\frac{1}{2}\tau\right] > 0,$$

which holds because the second inequality in (7.3.30) is met for β. So we obtain $\alpha_k = 1$ for sufficiently large k.

Second, we prove (7.3.31). By (7.3.27) and (7.1.7), we have

$$\begin{aligned} x_{k+1} - x^* &= x_k - x^* - (J_k^T J_k + \mu_k D_k)^{-1} g_k \\ &= x_k - x^* - (J_k^T J_k + \mu_k D_k)^{-1}[G_k(x_k - x^*) \\ &\quad + g_k + G_k(x^* - x_k)] \\ &= -(J_k^T J_k + \mu_k D_k)^{-1}[S(x_k)(x_k - x^*) \\ &\quad - \mu_k D_k(x_k - x^*) + g_k + G_k(x^* - x_k)]. \end{aligned} \tag{7.3.34}$$

Taking norm gives

$$\begin{aligned} \|x_{k+1} - x^*\| &\le \|(J_k^T J_k)^{-1}\|(\|S(x_k)\|\|x_k - x^*\| \\ &\quad + \mu_k\|D_k\|\|x_k - x^*\| + \|g_k + G_k(x^* - x_k)\|). \end{aligned} \tag{7.3.35}$$

Since

$$\begin{aligned} \|g_k + G_k(x^* - x_k)\| &= \|g_k - g(x^*) - G_k(x_k - x^*)\| \\ &\le \epsilon_k\|x_k - x^*\|, \end{aligned} \tag{7.3.36}$$

where $\epsilon_k \to 0$, then dividing the both sides of (7.3.35) by $\|x_k - x^*\|$ deduces

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \le \|(J_k^T J_k)^{-1}\|(\|S(x_k)\| + \mu_k\|D_k\| + \epsilon_k). \tag{7.3.37}$$

Note that $\mu_k \to 0$ and that $\epsilon_k \to 0$, and it follows immediately that

$$\limsup \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \le \frac{M}{l} = \tau,$$

which proves (7.3.31).

Finally, since $g(x^*) = 0$ and $G(x^*) = J(x^*)^T J(x^*) + S(x^*)$ with the lower bound $l - M > 0$ of the smallest eigenvalue, then $G(x^*)$ is positive definite. Hence x^* is a strict local minimizer of $f(x)$. □

As mentioned above, the Levenberg-Marquardt method can be described and analyzed by use of the framework of trust region method (7.3.1)–(7.3.2) or (7.3.3). So, following the discussions of Section 6.1, we immediately have the following algorithm and theorem which are straightforward consequences of Algorithm 6.1.1 and Theorem 6.1.9, respectively.

Algorithm 7.3.9 *(Trust-Region Type Levenberg-Marquardt Algorithm)*

Step 1. Given initial point $x_0, \bar{\Delta}, \Delta_0 \in (0, \bar{\Delta}), \epsilon \geq 0, 0 < \eta_1 \leq \eta_2 < 1$ *and* $0 < \gamma_1 < 1 < \gamma_2$, $k := 0$.

Step 2. If $\|g_k\| = \|J_k^T r_k\| \leq \epsilon$, *stop.*

Step 3. Approximately solve the subproblem (7.3.1)–(7.3.2) for s_k.

Step 4. Compute

$$\begin{aligned} Pred_k &= f(x_k) - q_k(s_k), \\ Ared_k &= f(x_k) - f(x_k + s_k), \\ r_k &= \frac{Ared_k}{Pred_k}. \end{aligned}$$

Step 5. If $r_k < \eta_1$, *set* $\Delta_k = \gamma_1 \Delta_k$ *and go to Step 3.*

Step 6. Set $x_{k+1} = x_k + s_k$. *Set*

$$\Delta_{k+1} = \begin{cases} \min\{\gamma_2 \Delta_k, \bar{\Delta}\}, & \text{if } r_k \geq \eta_2 \text{ and } \|s_k\| = \Delta_k, \\ \Delta_k, & \text{otherwise.} \end{cases}$$

Step 7. Set $k := k + 1$, *go to Step 2.* □

From Step 3 of the above algorithm, s_k is the approximate solution of subproblem (7.3.1)–(7.3.2). It follows from Lemma 6.1.3 that

$$q_k(0) - q_k(s_k) \geq c_1 \|J_k^T r_k\| \min\left(\Delta_k, \frac{\|J_k^T r_k\|}{\|J_k^T J_k\|}\right) \tag{7.3.38}$$

for some constant $c_1 > 0$.

Now we can state the convergence result which is a straightforward consequence of Theorem 6.1.9.

Theorem 7.3.10 *Suppose that the function* $f(x) = \frac{1}{2}\sum_{i=1}^{m}[r_i(x)]^2$ *is twice continuously differentiable, that the level set*

$$\mathcal{L}(x_0) = \{x \mid \ f(x) \leq f(x_0)\}$$

is bounded, and that there are constants $M_1 > 0, M_2 > 0$ *such that*

$$\|\nabla^2 f(x)\| \leq M_1, \quad \forall x \in \mathcal{L}(x_0),$$
$$\|J(x)^T J(x)\| \leq M_2, \quad \forall x \in \mathcal{L}(x_0).$$

Then we have that

$$\lim_{k\to\infty} \nabla f(x_k) = \lim_{k\to\infty} J_k^T r_k = 0. \tag{7.3.39}$$

7.4 Implementation of L-M Method

There are various implementations of the Levenberg-Marquardt method. Moré [218] gave an efficient and reliable implementation, which is contained in the MINPACK software package.

The Levenberg-Marquardt method Moré [218] considered is to find s by means of solving equations

$$s(\mu) = -(J_k^T J_k + \mu_k D_k^T D_k)^{-1} J_k^T r_k, \tag{7.4.1}$$

which correspond to a trust region subproblem (or a constrained linear least-squares problem)

$$\min \ \|r_k + J_k s\| \ \text{ s.t. } \|D_k s\| \leq \Delta_k, \tag{7.4.2}$$

where $\Delta_k > 0$ is the trust-region radius. If J_k is singular and $\mu_k = 0$, the solution of (7.4.2) can be defined by a limit

$$D_k s(0) = \lim_{\mu_k \to 0^+} D_k s(\mu_k) = -(J_k D_k^{-1})^+ r_k. \tag{7.4.3}$$

There are two possibilities: either $\mu_k = 0$ and $\|D_k s(0)\| \leq \Delta_k$, in which case $s(0)$ is the solution of (7.4.2); or $\mu_k > 0$ and $\|D_k s(\mu_k)\| = \Delta_k$, in which case $s(\mu_k)$ is a unique solution of (7.4.2). Hence we can describe the following algorithm.

Algorithm 7.4.1 *(Levenberg-Marquardt Algorithm)*

(a) Given $\Delta_k > 0$, find $\mu_k \geq 0$ such that

$$(J_k^T J_k + \mu_k D_k)s_k = -J_k^T r_k.$$

Then either $\mu_k = 0$ and $\|D_k s_k\| \leq \Delta_k$, or $\mu_k > 0$ and $\|D_k s_k\| = \Delta_k$.

(b) If $\|r(x_k + s_k)\| \leq \|r(x_k)\|$, set $x_{k+1} = x_k + s_k$ and compute J_{k+1}; otherwise set $x_{k+1} = x_k$ and $J_{k+1} = J_k$.

(c) Choose Δ_{k+1} and D_{k+1}. □

In the following, we discuss how to perform the above algorithm efficiently and reliably.

(1) How to solve the trust-region subproblem (i.e., constrained linear least-squares problem).

For equations

$$(J_k^T J_k + \mu_k D_k^T D_k)s = -J_k^T r_k, \tag{7.4.4}$$

the simplest way is using Cholesky factorization. However, because of the special structure of the coefficient matrix in (7.4.4), it is easy to use QR factorization.

Note that (7.4.4) are just the normal equations for linear least-squares problem

$$\begin{bmatrix} J_k \\ \mu_k^{1/2} D_k \end{bmatrix} s \cong - \begin{bmatrix} r \\ 0 \end{bmatrix}. \tag{7.4.5}$$

For the structure of (7.4.5), instead of computing $J_k^T J_k$ and $D_k^T D_k$ and their Cholesky factorization, we can use column pivoting QR factorization.

Now we describe the two-step QR factorization to find the solution of the linear least-squares problem (7.4.5).

First Step: Calculate the QR factorization of J_k and obtain

$$QJ_k\pi = \begin{bmatrix} T & W \\ 0 & 0 \end{bmatrix}, \tag{7.4.6}$$

where Q is orthogonal, T is a nonsingular upper triangular matrix with rank(T) =rank(J_k), and π is a permutation matrix. If $\mu_k = 0$, then the

solution of (7.4.5) is

$$s = \pi \begin{bmatrix} T^{-1} & 0 \\ 0 & 0 \end{bmatrix} Qr_k \equiv J_k^- r_k, \tag{7.4.7}$$

where J_k^- denotes $\{1,3\}$-inverse satisfying

$$J_k J_k^- J_k = J_k,\ J_k J_k^- = (J_k J_k^-)^T$$

(see X.He and W. Sun [172], Ben-Israel and Greville [12]). If $\mu_k > 0$, since (7.4.6) becomes

$$\begin{bmatrix} Q & 0 \\ 0 & \pi^T \end{bmatrix} \begin{bmatrix} J_k \\ \mu_k^{1/2} D_k \end{bmatrix} \pi = \begin{bmatrix} R \\ 0 \\ D_\mu \end{bmatrix}, \tag{7.4.8}$$

where $D_\mu = \mu_k^{1/2} \pi^T D_k \pi$ and R is an upper trapezoid matrix, it follows from (7.4.8) that (7.4.5) becomes

$$\begin{bmatrix} R \\ 0 \\ D_\mu \end{bmatrix} \pi^T s = - \begin{bmatrix} Qr \\ 0 \end{bmatrix}. \tag{7.4.9}$$

Second Step: It is easy to eliminate D_μ in (7.4.9) by a sequence of $n(n+1)/2$ Givens rotations and obtain

$$W \begin{bmatrix} R \\ 0 \\ D_\mu \end{bmatrix} = \begin{bmatrix} R_\mu \\ 0 \\ 0 \end{bmatrix}, \tag{7.4.10}$$

where R_μ is a nonsingular upper triangular matrix and W a product of a sequence of rotations. Then (7.4.9) becomes

$$\begin{bmatrix} R_\mu \\ 0 \end{bmatrix} \pi^T s = -W \begin{bmatrix} Qr \\ 0 \end{bmatrix} \triangleq \begin{bmatrix} u \\ v \end{bmatrix}, \tag{7.4.11}$$

and we obtain

$$s = \pi R_\mu^{-1} u. \tag{7.4.12}$$

(2) How to update the trust-region radius Δ_k.

As described in §6.1, the choice of Δ_k depends on the ratio between actual reduction and predicted reduction of the objective function. In the nonlinear least-squares case, the ratio is

$$\rho = \frac{\|r(x_k)\|^2 - \|r(x_k + s_k)\|^2}{\|r(x_k)\|^2 - \|r(x_k) + J(x_k)s_k\|^2}, \tag{7.4.13}$$

which measures the agreement between the linearized model and the nonlinear function. For example, if $r(x)$ is linear, then $\rho = 1$. If $J(x_k)^T r(x_k) \neq 0$, then $\rho \to 1$ when $\|s_k\| \to 0$. If $\|r(x_k + s_k)\| \geq \|r(x_k)\|$, then $\rho \leq 0$.

Because of roundoff error, there may be overflow when we compute ρ by (7.4.13). So we write (7.4.13) in a safeguard form. Multiplying both sides of (7.4.4) by $2s^T$ yields

$$-2r_k^T J_k^T s = 2s^T J_k^T J_k s + 2\mu_k s D_k^T D_k s,$$

which is

$$r_k^T r_k - r_k^T r_k - 2r_k^T J_k^T s - s^T J_k^T J_k s = s^T J_k^T J_k s + 2\mu_k s^T D_k^T D_k s.$$

Then we obtain

$$\|r_k\|^2 - \|r_k + J_k s\|^2 = \|J_k s\|^2 + 2\mu_k \|D_k s\|^2. \tag{7.4.14}$$

Substituting the above into (7.4.13) gives

$$\rho = \frac{1 - \left[\dfrac{\|r(x_k + s_k)\|}{\|r(x_k)\|}\right]^2}{\left[\dfrac{\|J_k s\|}{\|r(x_k)\|}\right]^2 + 2\left[\dfrac{\mu_k^{1/2}\|D_k s\|}{\|r(x_k)\|}\right]^2}. \tag{7.4.15}$$

It is easy to see from (7.4.14) that

$$\|J_k s\| \leq \|r(x_k)\|, \ \mu_k^{1/2}\|D_k s\| \leq \|r(x_k)\|.$$

Hence the computation in (7.4.15) will not lead to overflow. Also, regardless of roundoff error, the denominator in (7.4.15) is always nonnegative. It should be mentioned that when $\|r(x_k + s_k)\| \gg \|r(x_k)\|$, the numerator in (7.4.15) may be overflown. However, since we are only interested in $\rho \geq 0$, then when $\|r(x_k + s_k)\| > \|r(x_k)\|$, we set $\rho = 0$ without needing to compute ρ by (7.4.15).

(3) How to find a Levenberg-Marquardt parameter.

In the Moré algorithm, if

$$|\phi(\mu)| \leq \sigma\Delta,\ \sigma \in (0,1), \tag{7.4.16}$$

where

$$\phi(\mu) = \|D(J^TJ + \mu D^TD)^{-1}J^Tr\| - \Delta, \tag{7.4.17}$$

then $\mu > 0$ is accepted as a Levenberg-Marquardt parameter, where σ indicates the related error in $\|D_k s(\mu)\|$. If $\phi(0) \leq 0$, then $\mu = 0$ is a required parameter. Therefore we only need to discuss the case of $\phi(0) > 0$. Since ϕ is a continuous and strictly descending function, then, when $\mu \to \infty$, we have $\phi(\mu) \to -\Delta$. Thus, there exists a unique $\mu^* > 0$ such that $\phi(\mu^*) = 0$. To determine the Levenberg-Marquardt parameter, we start from $\mu_0 > 0$ and generate a sequence $\{\mu_k\} \to \mu^*$.

From (7.4.17), we have

$$\phi(\mu) = \|(\tilde{J}^T\tilde{J} + \mu I)^{-1}\tilde{J}^Tr\| - \Delta, \tag{7.4.18}$$

where $\tilde{J} = JD^{-1}$. Let $\tilde{J} = U\Sigma V^T$ be the singular value decomposition of $\tilde{J}$, then

$$\phi(\mu) = \sum_{i=1}^{n}\left[\frac{\sigma_i^2 z_i^2}{(\sigma_i^2 + \mu)^2}\right]^{1/2} - \Delta, \tag{7.4.19}$$

where $z = U^Tr$, $\sigma_1, \cdots, \sigma_n$ are singular values of $\tilde{J}$. Hence we assume

$$\phi(\mu) \doteq \frac{a}{b + \mu} \equiv \tilde{\phi}(\mu) \tag{7.4.20}$$

and choose a and b such that $\tilde{\phi}(\mu_k) = \phi(\mu_k), \tilde{\phi}'(\mu_k) = \phi'(\mu_k)$. Then $\tilde{\phi}(\mu_{k+1}) = 0$ if

$$\mu_{k+1} = \mu_k - \left[\frac{\phi(\mu_k) + \Delta}{\Delta}\right]\left[\frac{\phi(\mu_k)}{\phi'(\mu_k)}\right]. \tag{7.4.21}$$

To make computation of μ_{k+1} safe and reliable, the Moré algorithm designs the following technique for computing μ_{k+1}.

Let

$$u_0 = \frac{\|(JD^{-1})^Tr\|}{\Delta},$$

$$l_0 = \begin{cases} -\phi(0)/\phi'(0), & \text{if } J \text{ is nonsingular,} \\ 0, & \text{otherwise,} \end{cases}$$

(a) If $\mu_k \notin (l_k, u_k)$, set $\mu_k = \max\{0.001u_k, (l_k u_k)^{1/2}\}$.

(b) Compute $\phi(\mu_k)$ and $\phi'(\mu_k)$. Update u_k:

$$u_{k+1} = \begin{cases} \mu_k, & \text{if } \phi(\mu_k) < 0, \\ u_k, & \text{otherwise.} \end{cases}$$

Update l_k:

$$l_{k+1} = \max\left\{l_k, \mu_k - \frac{\phi(\mu_k)}{\phi'(\mu_k)}\right\}.$$

(c) Compute μ_{k+1} by (7.4.21).

The above algorithm gives upper and lower bounds of μ_k. In (a), it shows that if μ_k is not in (l_k, u_k), a point in (l_k, u_k) inclining to l_k will replace μ_k. In (b), the convexity of ϕ guarantees that the Newton's iteration can be used to update l_k. The sequence $\{\mu_k\}$ generated by the algorithm will converge to μ^*. In fact, if we set $\sigma = 0.1$, it takes no more than two steps on average to satisfy (7.4.16).

As to computing $\phi'(\mu)$, we have from (7.4.17) that

$$\phi'(\mu) = -\frac{(D^T q(\mu))^T (J^T J + \mu D^T D)^{-1} (D^T q(\mu))}{\|q(\mu)\|}, \tag{7.4.22}$$

where $q(\mu) = Ds(\mu)$. By (7.4.8) and (7.4.10) we get

$$\pi^T (J^T J + \mu D^T D)\pi = R_\mu^T R_\mu.$$

Then

$$\phi'(\mu) = -\|q(\mu)\| \left\| R_\mu^{-T} \left(\frac{\pi^T D^T q(\mu)}{\|q(\mu)\|} \right) \right\|^2. \tag{7.4.23}$$

(4) How to update the scaling matrix.

In the Levenberg-Marquardt method, D_k is a diagonal matrix which reduces the effects of poor scaling of the problems. In the algorithm, we choose

$$D_k = \text{diag}\,(d_1^{(k)}, \cdots, d_n^{(k)}), \tag{7.4.24}$$

where

$$\begin{aligned} d_i^{(0)} &= \|\partial_i r(x_0)\|, \\ d_i^{(k)} &= \max\{d_i^{(k-1)},\ \|\partial_i r(x_k)\|\},\ k \geq 1. \end{aligned}$$

We should point out that the above scaling is invariant under scaling, that is, if D is a diagonal and positive definite matrix, then for function $r(x)$ with starting point x_0 and for function $\tilde{r}(x) = r(D^{-1}x)$ with starting point $\tilde{x}_0 = Dx_0$, Algorithm 7.4.1 generates the same sequence of iterates.

Finally, we give the Moré version of the Levenberg-Marquardt algorithm and the convergence theorem.

Algorithm 7.4.2 *(Moré's Version)*

(a) Let $\sigma \in (0,1)$. If $\|D_k J_k^- r_k\| \leq (1+\sigma)\Delta_k$, set $\mu_k = 0$ and $s_k = -J_k^- r_k$; otherwise, determine $\mu_k > 0$ such that if

$$\begin{bmatrix} J_k \\ \mu_k^{1/2} D_k \end{bmatrix} s_k \cong - \begin{bmatrix} r_k \\ 0 \end{bmatrix}$$

(i.e., s_k is the solution of the above least-squares problem), then

$$(1-\sigma)\Delta_k \leq \|D_k s_k\| \leq (1+\sigma)\Delta_k.$$

(b) Compute the ratio ρ_k between the actual reduction and the predicted reduction of the objective function.

(c) If $\rho_k \leq 0.0001$, set $x_{k+1} = x_k$ and $J_{k+1} = J_k$.
If $\rho_k > 0.0001$, set $x_{k+1} = x_k + s_k$, and compute J_{k+1}.

(d) If $\rho_k \leq \frac{1}{4}$, set $\Delta_{k+1} \in \left[\frac{1}{10}\Delta_k, \frac{1}{2}\Delta_k\right]$. If either $\rho_k \in \left[\frac{1}{4}, \frac{1}{3}\right]$ and $\mu_k = 0$, or $\rho_k \geq \frac{3}{4}$, then set $\Delta_{k+1} = 2\|D_k s_k\|$.

(e) Update D_k by (7.4.24). □

For the above algorithm, the convergence theorem is stated as follows without proof. The interested readers may consult Moré [218].

Theorem 7.4.3 *Let $r : R^n \to R^m$ be continuously differentiable. Let $\{x_k\}$ be a sequence generated by Algorithm 7.4.2. Then*

$$\liminf_{k \to +\infty} \|(J_k D_k^{-1})^T r_k\| = 0. \tag{7.4.25}$$

This result indicates that the scaled gradient is, at last, sufficiently small. If $\{J_k\}$ is bounded, then (7.4.25) implies

$$\liminf_{k \to +\infty} \|J_k^T r_k\| = 0. \tag{7.4.26}$$

Further, if $\nabla r(x)$ is uniformly continuous, then

$$\lim_{k\to+\infty} \|J_k^T r_k\| = 0. \tag{7.4.27}$$

7.5 Quasi-Newton Method

We have seen from the above sections that, for large-residual problems (i.e., $r(x)$ is large or $r(x)$ is severely nonlinear), the performance of the Gauss-Newton method and Levenberg-Marquardt method is usually poor. The convergence is slow and only linear. This is because we don't use the second-order information $S(x)$ in Hessian $G(x) = J(x)^T J(x) + S(x)$ which is significant. As mentioned in §7.1, in fact, computation of $S(x)$ is either difficult or expensive. It is also not suitable to use the secant approximation of the whole Hessian $G(x)$. So, it may be a good idea to use a secant approximation of the second information $S(x) = \sum_{i=1}^m r_i(x)\nabla^2 r_i(x)$ in $G(x)$.

Let B_k be a secant approximation of $S(x_k)$, then the iteration (7.1.10) becomes

$$(J(x_k)^T J(x_k) + B_k)d_k = -J(x_k)^T r(x_k). \tag{7.5.1}$$

Since

$$S(x_{k+1}) = \sum_{i=1}^m r_i(x_{k+1})\nabla^2 r_i(x_{k+1}), \tag{7.5.2}$$

then we use

$$B_{k+1} = \sum_{i=1}^m r_i(x_{k+1})(H_i)_{k+1} \tag{7.5.3}$$

to approximate $S(x_{k+1})$, where $(H_i)_{k+1}$ is a secant approximation of $\nabla^2 r_i(x_{k+1})$. Then we have that

$$(H_i)_{k+1}(x_{k+1} - x_k) = \nabla r_i(x_{k+1}) - \nabla r_i(x_k). \tag{7.5.4}$$

Hence, we get

$$\begin{aligned} B_{k+1}(x_{k+1} - x_k) &= \sum_{i=1}^m r_i(x_{k+1})(H_i)_{k+1}(x_{k+1} - x_k) \\ &= \sum_{i=1}^m r_i(x_{k+1})(\nabla r_i(x_{k+1}) - \nabla r_i(x_k)) \\ &= (J(x_{k+1}) - J(x_k))^T r(x_{k+1}) \triangleq y_k \end{aligned} \tag{7.5.5}$$

which is a quasi-Newton condition imposed on B_k.

Similarly, if we ask

$$(J(x_{k+1})^T J(x_{k+1}) + B_{k+1})s_k = J(x_{k+1})^T r(x_{k+1}) - J(x_k)^T r(x_k) \quad (7.5.6)$$

to hold, then B_{k+1} should satisfy

$$B_{k+1}s_k = \tilde{y}_k, \quad (7.5.7)$$

where

$$\tilde{y}_k = J(x_{k+1})^T r(x_{k+1}) - J(x_k)^T r(x_k) - J(x_{k+1})^T J(x_{k+1})s_k. \quad (7.5.8)$$

Now, we give an update formula for B_k by weighted Frobenius norm. The following theorem is a restatement of Theorem 5.1.10 in Chapter 5.

Theorem 7.5.1 *Let $v_k^T s_k > 0$ and $T \in R^{n\times n}$ be a symmetric and positive definite matrix such that*

$$TT^T s_k = v_k, \quad (7.5.9)$$

where

$$\begin{aligned} v_k &\triangleq \nabla f(x_{k+1}) - \nabla f(x_k) \\ &= J(x_{k+1})^T r(x_{k+1}) - J(x_k)^T r(x_k). \end{aligned} \quad (7.5.10)$$

Then the update

$$\begin{aligned} B_{k+1} &= B_k + \frac{(y_k - B_k s_k)v_k^T + v_k(y_k - B_k s_k)^T}{s_k^T v_k} \\ &\quad - \frac{s_k^T(y_k - B_k s_k)}{(s_k^T v_k)^2} v_k v_k^T \end{aligned} \quad (7.5.11)$$

is a unique solution of the minimization problem

$$\begin{aligned} \min \quad & \|T^{-T}(B_{k+1} - B_k)T^{-1}\|_F \\ \text{s.t.} \quad & (B_{k+1} - B_k) \text{ is symmetric, } B_{k+1}s_k = y_k. \end{aligned} \quad (7.5.12)$$

Dennis, Gay and Welsch [88] use the quasi-Newton condition (7.5.5) and (7.5.11), and present a quasi-Newton algorithm NL2SOL with trust region strategy. At each step, it is required to solve the subproblem

$$\begin{aligned} \min \quad & \frac{1}{2} r(x_k)^T r(x_k) + (x - x_k)^T J(x_k)^T r(x_k) \\ & + \frac{1}{2}(x - x_k)^T (J(x_k)^T J(x_k) + B_k)(x - x_k) \\ \text{s.t.} \quad & \|x - x_k\| \le \Delta_k. \end{aligned} \quad (7.5.13)$$

In this algorithm, a deficiency of the update (7.5.11) for B_k is that this matrix is not guaranteed to vanish when the iterates approach to a zero-residual solution, so it can interfere with superlinear convergence. This problem can be avoided by a strategy of scaling B_k, that is, we choose a scaling factor

$$\gamma_k = \min\left\{\frac{s_k^T y_k}{s_k^T B_k s_k}, 1\right\}, \tag{7.5.14}$$

multiply B_k by γ_k, and then use (7.5.11) to update it.

Numerical experiments show that, for large-residual problems, quasi-Newton algorithm NL2SOL is significantly advantageous; for small-residual problems, the performance of NL2SOL and Moré's Levenberg-Marquardt algorithm is similar; for zero-residual problems we prefer the Gauss-Newton method. Therefore, the Gauss-Newton method, Levenberg-Marquardt method, and quasi-Newton method introduced in this chapter are very important to solve nonlinear least-squares problems. Now, Moré's Levenberg-Marquardt algorithm and quasi-Newton algorithm NL2SOL are very popular.

Similar to the above discussion, Bartholomew-Biggs [15] uses the quasi-Newton condition (7.5.5) and the following rank-one updating formula

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}, \tag{7.5.15}$$

and gives a quasi-Newton method for nonlinear least-squares problems. In Bartholomew-Biggs' algorithm, the scaling factor is

$$\gamma_k = r_{k+1}^T r_{k+1} / r_k^T r_k. \tag{7.5.16}$$

Fletcher and Xu [139] presented a hybrid algorithm which combines Gauss-Newton method and quasi-Newton method. If the current Gauss-Newton step reduces the function f by a certain fixed amount, i.e.,

$$f(x_k) - f(x_{k+1}) \geq \tau f(x_k),\ \tau \in (0,1), \tag{7.5.17}$$

we use the Gauss-Newton step. Otherwise, we use quasi-Newton update (for example, BFGS update). In the zero-residual case, the method eventually takes Gauss-Newton steps and gives quadratic convergence; while in the nonzero-residual case, the method eventually reduces to BFGS formula. The theoretical analysis shows that Fletcher-Xu method is superlinearly convergent. Normally, we take $\tau = 0.2$ in (7.5.17).

Exercises

1. Solve the least-squares problem

$$\min f(x) = \frac{1}{2}[(x_2 - x_1^2)^2 + (1 - x_1)^2], \quad x^{(0)} = (0, 0)^T$$

by Gauss-Newton method and Levenberg-Marquardt method.

2. Consider nonlinear least-squares problems:

$$\min_{x \in R^n} f(x) = \frac{1}{2} r(x)^T r(x) = \frac{1}{2} \sum_{i=1}^{m} [r_i(x)]^2, \quad m \geq n$$

where

$$\begin{aligned}
r_1(x) &= x_1^2 + x_2^2 + x_3^2 - 1, \\
r_2(x) &= x_1^2 + x_2^2 + (x_3 - 2)^2 - 1, \\
r_3(x) &= x_1 + x_2 + x_3 - 1, \\
r_4(x) &= x_1 + x_2 - x_3 + 1, \\
r_5(x) &= x_1^3 + 3x_2^2 + (5x_3 - x_1 + 1)^2 - 36.
\end{aligned}$$

(1) Compute $\nabla f(x), J(x)^T J(x)$, and $\nabla^2 f(x)$.
(2) Please answer whether $J(x)^T J(x) = \nabla^2 f(x)$ holds for $x = (0, 0)^T$, and why?

3. Prove (7.2.7)–(7.2.9).

4. Suppose that the function $f(x) = \frac{1}{2} \sum_{i=1}^{m} [r_i(x)]^2$ is twice continuously differentiable, and that the level set

$$\mathcal{L}(x_0) = \{x \mid \ f(x) \leq f(x_0)\}$$

is bounded. Let the sequence $\{x_k\}$ generated by trust-region type Levenberg-Marquardt Algorithm 7.3.9 converge to x^* with positive definite $\nabla^2 f(x^*)$ and

$$S(x^*) = \sum_{i=1}^{m} r_i(x^*) \nabla^2 r_i(x^*) = 0.$$

Prove that $\{x_k\}$ converges to x^* with quadratic rate.

5. Let $r \in R^m,\ J \in R^{m\times n},\ \mu > 0$. Prove that $s = -(J^TJ + \mu I)^{-1}J^Tr$ is the solution of the least squares problem

$$\min \ \|Ws + y\|^2,$$

where

$$W = \begin{bmatrix} J \\ \mu^{\frac{1}{2}} I \end{bmatrix}, \quad y = \begin{bmatrix} r \\ 0 \end{bmatrix}.$$

6. Consider nonlinear least-squares problems

$$\min_{x\in R^n} f(x) = \frac{1}{2} r(x)^T r(x) = \frac{1}{2}\sum_{i=1}^{m} [r_i(x)]^2, \ \ m \geq n$$

where $r : R^n \to R^m$ is a nonlinear function of x and its Jacobian matrix $J(x)$ is full-rank for all $x \in R^n$. Denote the Gauss-Newton direction, the Levenberg-Marquardt direction, and the steepest descent direction respectively by s^{GN}, s^{LM}, and s^C:

$$\begin{aligned} s^{GN} &= -(J^TJ)^{-1}J^Tr, \\ s^{LM} &= -(J^TJ + \mu I)^{-1}J^Tr, \\ s^C &= -J^Tr. \end{aligned}$$

Prove that

$$\begin{aligned} &\lim_{\mu\to 0} s^{LM}(\mu) = s^{GN}, \\ &\lim_{\mu\to\infty} \frac{s^{LM}(\mu)}{\|s^{LM}(\mu)\|} = \frac{s^C}{\|s^C\|}. \end{aligned}$$

Chapter 8

Theory of Constrained Optimization

8.1 Constrained Optimization Problems

A general formulation for nonlinear constrained optimization is

$$\min_{x \in R^n} \quad f(x) \tag{8.1.1}$$

$$\text{s.t.} \quad c_i(x) = 0,\ i = 1, \cdots, m_e; \tag{8.1.2}$$

$$c_i(x) \geq 0,\ i = m_e + 1, \cdots, m, \tag{8.1.3}$$

where the objective function $f(x)$ and the constrained functions $c_i(x), (i = 1, \cdots, m)$ are all smooth, real-valued functions on R^n, and at least one is nonlinear, and m_e and m are nonnegative integers with $0 \leq m_e \leq m$. Sometimes, we set

$$E = \{1, \cdots, m_e\} \text{ and } I = \{m_e + 1, \cdots, m\}$$

as index sets of equality constraints and inequality constraints, respectively. If $m = 0$, the problem (8.1.1)-(8.1.3) is an unconstrained optimization problem; if $m_e = m \neq 0$, the problem is called an equality constrained optimization problem; if all $c_i(x)(i = 1, \cdots, m)$ are linear functions, the problem (8.1.1)-(8.1.3) is called a linearly constrained optimization problem. A linearly constrained optimization problem with quadratic objective function $f(x)$ is said to be a quadratic programming problem which will be discussed in Chapter 9.

Definition 8.1.1 *The point $x \in R^n$ is said to be a feasible point if and only if (8.1.2)-(8.1.3) hold. The set of all feasible points is said to be a feasible set.*

In problem (8.1.1)–(8.1.3), (8.1.2)-(8.1.3) are constrained conditions. From Definition 8.1.1, the feasible point is the point satisfying all constraints. We write the feasible set X as

$$X = \left\{ x \;\middle|\; \begin{array}{ll} c_i(x) = 0, & i = 1, \cdots, m_e; \\ c_i(x) \geq 0, & i = m_e + 1, \cdots, m \end{array} \right\}. \tag{8.1.4}$$

or

$$X = \{x \mid c_i(x) = 0,\ i \in E;\ c_i(x) \geq 0,\ i \in I\}. \tag{8.1.5}$$

So, we can rewrite problem (8.1.1)-(8.1.3) as

$$\min_{x \in X} f(x) \tag{8.1.6}$$

which means that solution of constrained optimization problem (8.1.1)-(8.1.3) is just to find a point x on the feasible set X, such that the objective function $f(x)$ is minimized.

In the following, we give some definitions about local and global minimizers.

Definition 8.1.2 *If $x^* \in X$ and if*

$$f(x) \geq f(x^*),\ \forall x \in X, \tag{8.1.7}$$

then x^* is said to be a global minimizer of the problem (8.1.1)–(8.1.3). If $x^* \in X$ and if

$$f(x) > f(x^*),\ \forall x \in X, x \neq x^*, \tag{8.1.8}$$

then x^* is said to be a strict global minimizer.

Definition 8.1.3 *If $x^* \in X$ and if there is a neighborhood $B(x^*, \delta)$ of x^* such that*

$$f(x) \geq f(x^*),\ \forall x \in X \cap B(x^*, \delta), \tag{8.1.9}$$

then x^* is said to be a local minimizer of problem (8.1.1)–(8.1.3), where

$$B(x^*, \delta) = \{x \mid \|x - x^*\|_2 \leq \delta\} \tag{8.1.10}$$

and $\delta > 0$.

If $x^* \in X$ *and if there is a neighborhood* $B(x^*, \delta)$ *of* x^* *such that*

$$f(x) > f(x^*),\ \forall x \in X \cap B(x^*, \delta), x \neq x^*, \tag{8.1.11}$$

then x^* *is said to be a strict local minimizer.*

Definition 8.1.4 *If* $x^* \in X$ *and if there is a neighborhood* $B(x^*, \delta)$ *of* x^* *such that* x^* *is the only local minimizer in* $X \cap B(x^*, \delta)$*, then* x^* *is an isolated local minimizer.*

Obviously, a global minimizer is also a local minimizer.

Assume that x^* is a local minimizer of problem (8.1.1)–(8.1.3), if there is an index $i_0 \in I = [m_e + 1, m]$ such that

$$c_{i_0}(x^*) > 0, \tag{8.1.12}$$

then, if we delete the i_0-th constraint, x^* is still the local minimizer of the problem obtained by deleting i_0-th constraint. Thus, we say that the i_0-th constraint is inactive at x^*. Now, we give the definitions of active constraint and inactive constraint. First, write

$$I(x) = \{i \mid c_i(x) = 0, i \in I\}. \tag{8.1.13}$$

Definition 8.1.5 *For any* $x \in R^n$*, the set*

$$\mathcal{A}(x) = E \cup I(x) \tag{8.1.14}$$

is an index set of active constraints at x*,* $c_i(x)\,(i \in \mathcal{A}(x))$ *is an active constraint at* x*,* $c_i(x)\,(i \notin \mathcal{A})$ *is an inactive constraint at* x*.*

Assume that $\mathcal{A}(x^*)$ is an index set of the active constraints of problem (8.1.1)–(8.1.3) at x^*, then, from the observation about inactive constraints, it is enough for us to solve the constrained optimization problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & c_i(x) = 0,\ i \in \mathcal{A}(x^*). \end{aligned} \tag{8.1.15}$$

In general, it is easier to solve equality constraint problem (8.1.15) than the original problem (8.1.1)–(8.1.3).

8.2 First-Order Optimality Conditions

In this section we discuss the first-order optimality conditions. Since the feasible directions play a very important role in deriving the optimality conditions, we first give some definitions of several feasible directions.

Definition 8.2.1 *Let $x^* \in X$, $0 \neq d \in R^n$. If there exists $\delta > 0$ such that*

$$x^* + td \in X, \ \forall t \in [0, \delta],$$

then d is said to be a feasible direction of X at x^*. The set of all feasible directions of X at x^* is

$$FD(x^*, X) = \{d \,|\, x^* + td \in X, \ \forall t \in [0, \delta]\}. \tag{8.2.1}$$

Definition 8.2.2 *Let $x^* \in X$ and $d \in R^n$. If*

$$\begin{aligned} d^T \nabla c_i(x^*) &= 0, \qquad i \in E, \\ d^T \nabla c_i(x^*) &\geq 0, \qquad i \in I(x^*), \end{aligned}$$

then d is said to be a linearized feasible direction of X at x^*. The set of all linearized feasible directions of X at x^* is

$$LFD(x^*, X) = \left\{ d \;\middle|\; \begin{array}{ll} d^T \nabla c_i(x^*) = 0, & i \in E \\ d^T \nabla c_i(x^*) \geq 0, & i \in I(x^*) \end{array} \right\}. \tag{8.2.2}$$

Definition 8.2.3 *Let $x^* \in X$ and $d \in R^n$. If there exist sequences $d_k (k = 1, 2, \cdots)$ and $\delta_k > 0, (k = 1, 2, \cdots)$ such that $x^* + \delta_k d_k \in X$, $\forall k$ and $d_k \to d, \delta_k \to 0$, then the limiting direction d is called the sequential feasible direction of X at x^*. The set of all sequential feasible directions of X at x^* is*

$$SFD(x^*, X) = \left\{ d \;\middle|\; \begin{array}{l} x^* + \delta_k d_k \in X, \forall k \\ d_k \to d, \delta_k \to 0 \end{array} \right\}. \tag{8.2.3}$$

In the definition above, if set $x_k = x^* + \delta_k d_k$, then $\{x_k\}$ is a feasible point sequence that satisfies:
(1) $x_k \neq x^*, \forall k$;
(2) $\lim_{k \to \infty} x_k = x^*$;

(3) $x_k \in X$ for all k sufficiently large.
If set $\delta_k = \|x_k - x^*\|$, then we have

$$d_k = \frac{x_k - x^*}{\|x_k - x^*\|} \to d,$$

which means that $x_k = x^* + \delta_k d_k$ is a feasible point sequence with the feasible direction d.

Note that if $SFD(x^*, X)$ includes the zero vector, it is referred to as the tangent cone of X at x^*, i.e.,

$$\mathcal{T}_X(x^*) = SFD(x^*, X) \cup \{0\}.$$

Obviously, by use of the above definitions of some feasible directions, we have the following lemma which indicates the relations of the above sets of feasible directions $FD(x^*, X), SFD(x^*, X)$ and $LFD(x^*, X)$.

Lemma 8.2.4 *Let $x^* \in X$. If all constraint functions are differentiable at x^*, then*

$$FD(x^*, X) \subseteq SFD(x^*, X) \subseteq LFD(x^*, X). \tag{8.2.4}$$

Proof. For any $d \in FD(x^*, X)$, it follows from Definition 8.2.1 that there is a $\delta > 0$ such that (8.2.1) holds. Set $d_k = d$ and $\delta_k = \delta/2^k$, then (8.2.3) holds and clearly $d_k \to d$ and $\delta_k \to 0$. Thus $d \in SFD(x^*, X)$. Since d is arbitrary, then

$$FD(x^*, X) \subseteq SFD(x^*, X). \tag{8.2.5}$$

Next, for any $d \in SFD(x^*, X)$, if $d = 0$, then $d \in LFD(x^*, X)$. Assume that $d \neq 0$. By Definition 8.2.3, there exist sequences $d_k\,(k = 1, 2, \cdots)$ and $\delta_k > 0\,(k = 1, 2, \cdots)$ such that (8.2.3) holds, and $d_k \to d \neq 0$ and $\delta_k \to 0$. By use of (8.2.3), we see that $x^* + \delta_k d_k \in X$, i.e.,

$$0 = c_i(x^* + \delta_k d_k) = \delta_k d_k^T \nabla c_i(x^*) + o(\|\delta_k d_k\|),\ i \in E; \tag{8.2.6}$$

$$0 \leq c_i(x^* + \delta_k d_k) = \delta_k d_k^T \nabla c_i(x^*) + o(\|\delta_k d_k\|),\ i \in I(x^*). \tag{8.2.7}$$

Dividing the above two equations by $\delta_k > 0$ and setting $k \to \infty$, we obtain (8.2.2). Thus we also have

$$SFD(x^*, X) \subseteq LFD(x^*, X). \tag{8.2.8}$$

Both (8.2.5) and (8.2.8) give the result of (8.2.4). □

In order to describe clearly necessary conditions for a local solution, it is convenient to introduce the set

$$\mathcal{D}(x') = \mathcal{D}' = \{d \mid d^T \nabla f(x') < 0\}, \tag{8.2.9}$$

which is called a set of descent direction at x'.

Now we describe the most basic necessary condition – geometry optimality condition as follows.

Theorem 8.2.5 *(Geometry optimality condition) Let $x^* \in X$ be a local minimizer of problem (8.1.1)-(8.1.3). If $f(x)$ and $c_i(x)\,(i = 1, 2, \cdots, m)$ are differentiable at x^*, then*

$$d^T \nabla f(x^*) \geq 0,\ \forall d \in SFD(x^*, X), \tag{8.2.10}$$

which means

$$SFD(x^*, X) \cap \mathcal{D}(x^*) = \phi, \tag{8.2.11}$$

where ϕ is an empty set.

Proof. For any $d \in SFD(x^*, X)$, there exist $\delta_k > 0\,(k = 1, 2, \cdots)$ and $d_k\,(k = 1, 2, \cdots)$ such that $x^* + \delta_k d_k \in X$ with $\delta_k \to 0$ and $d_k \to d$. Since $x^* + \delta_k d_k \to x^*$ and x^* is a local minimizer, then for k sufficiently large, we have

$$f(x^*) \leq f(x^* + \delta_k d_k) = f(x^*) + \delta_k d_k^T \nabla f(x^*) + o(\delta_k) \tag{8.2.12}$$

which implies

$$d^T \nabla f(x^*) \geq 0. \tag{8.2.13}$$

Since d is arbitrary, we obtain (8.2.10).

Furthermore, (8.2.13) also implies $d \notin \mathcal{D}(x^*)$, and hence $SFD(x^*, X) \cap \mathcal{D}(x^*) = \phi$. □

If we use terminology of the tangent cone to represent (8.2.10), we have

$$d^T \nabla f(x^*) \geq 0,\ \forall d \in \mathcal{T}_X(x^*),$$

i.e.,

$$-\nabla f(x^*)^T d \leq 0,\ \forall d \in \mathcal{T}_X(x^*). \tag{8.2.14}$$

This implies that

$$-\nabla f(x^*) \in \mathcal{N}_X(x^*), \tag{8.2.15}$$

where $\mathcal{N}_X(x^*)$ is the normal cone of X at x^*.

Theorem 8.2.5 shows that there is no sequential feasible direction at a local minimizer x^*. Unfortunately, it is not possible to proceed further without constraint qualification. In the following, by means of Farkas' Lemma 1.3.22 and the constraint qualification, we can get the first-order optimality condition — the famous Karush-Kuhn-Tucker Theorem.

Farkas' Lemma 1.3.22 gives the following form.

Lemma 8.2.6 *The set*

$$S = \left\{ d \;\middle|\; \begin{array}{l} d^T \nabla f(x^*) < 0, \\ d^T \nabla c_i(x^*) = 0,\ i \in E, \\ d^T \nabla c_i(x^*) \geq 0,\ i \in I \end{array} \right\} \tag{8.2.16}$$

is empty if and only if there exist real numbers $\lambda_i, i \in E$ and nonnegative real numbers $\lambda_i \geq 0, i \in I$ such that

$$\nabla f(x^*) = \sum_{i \in E} \lambda_i \nabla c_i(x^*) + \sum_{i \in I} \lambda_i \nabla c_i(x^*). \tag{8.2.17}$$

In fact, set

$$d = -x,\ \nabla f(x^*) = c,\ A = \begin{bmatrix} \nabla c_1^T(x^*) \\ \vdots \\ \nabla c_m^T(x^*) \end{bmatrix},\ \lambda = y,$$

we immediately have that (8.2.16) is just (1.3.49), and that (8.2.17) and $\lambda_i \geq 0, i \in I$ are just (1.3.50). This implies that Lemma 8.2.6 is a direct consequence of Farkas' Lemma 1.3.22 and also called Farkas' Lemma.

It is convenient to state the optimality condition by introducing the Lagrangian function

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i=1}^{m} \lambda_i c_i(x), \tag{8.2.18}$$

where $\lambda = (\lambda_1, \cdots, \lambda_m)^T \in R^m$ is a Lagrange multiplier vector.

Now, we are in a position to state the first-order necessary condition of a local minimizer by use of Farkas' Lemma and Theorem 8.2.5.

Theorem 8.2.7 *(Karush-Kuhn-Tucker Theorem)*

Let x^* be a local minimizer of problem (8.1.1)–(8.1.3). If the constraint qualification (CQ)

$$SFD(x^*, X) = LFD(x^*, X) \tag{8.2.19}$$

holds, then there exist Lagrange multipliers λ_i^* such that the following conditions are satisfied at (x^*, λ^*):

$$\nabla f(x^*) - \sum_{i=1}^{m} \lambda_i^* \nabla c_i(x^*) = 0, \tag{8.2.20}$$

$$c_i(x^*) = 0, \qquad \forall i \in E, \tag{8.2.21}$$

$$c_i(x^*) \geq 0, \qquad \forall i \in I, \tag{8.2.22}$$

$$\lambda_i^* \geq 0, \qquad \forall i \in I, \tag{8.2.23}$$

$$\lambda_i^* c_i(x^*) = 0, \qquad \forall i \in I. \tag{8.2.24}$$

Proof. Since x^* is a local minimizer, x^* is feasible and the conditions (8.2.21) and (8.2.22) are satisfied.

Let $d \in SFD(x^*, X)$; since x^* is a local minimizer, it follows from Theorem 8.2.5 that $d^T \nabla f(x^*) \geq 0$. By constraint qualification (8.2.19), we have $d \in LFD(x^*, X)$. Thus the system

$$d^T \nabla c_i(x^*) = 0, \qquad i \in E, \tag{8.2.25}$$

$$d^T \nabla c_i(x^*) \geq 0, \qquad i \in I(x^*), \tag{8.2.26}$$

$$d^T \nabla f(x^*) < 0 \tag{8.2.27}$$

has no solution. By Farkas' Lemma, we immediately obtain that

$$\nabla f(x^*) = \sum_{i \in E} \lambda_i^* \nabla c_i(x^*) + \sum_{i \in I(x^*)} \lambda_i^* \nabla c_i(x^*), \tag{8.2.28}$$

where $\lambda_i^* \in R\,(i \in E)$ and $\lambda_i^* \geq 0\,(i \in I(x^*))$. Setting $\lambda_i^* = 0\,(i \in I \backslash I(x^*))$, it follows that

$$\nabla f(x^*) = \sum_{i=1}^{m} \lambda_i^* \nabla c_i(x^*),$$

which is (8.2.20). It is obvious that $\lambda_i^* \geq 0, \forall i \in I$.

Finally, note that:

when $i \in I(x^*)$, $c_i(x^*) = 0$ and $\lambda_i^* \geq 0$, therefore $\lambda_i^* c_i(x^*) = 0$;

when $i \in I \setminus I(x^*)$, $c_i(x^*) > 0$ but $\lambda_i^* = 0$, therefore we also have $\lambda_i^* c_i(x^*) = 0$.

Thus we obtain that $\lambda_i^* c_i(x^*) = 0, \forall i \in I$. $\square$

Theorem 8.2.7 was presented by Kuhn and Tucker [193], and is known as the Kuhn-Tucker Theorem. Since Karush [185] also considered similarly the optimality condition for constrained optimization, the conditions (8.2.20)–(8.2.24) are often known as the Karush-Kuhn-Tucker conditions, or KKT conditions for short. A point that satisfies the conditions is referred to as a KKT point.

In KKT conditions, (8.2.20) is called a stationary point condition, because it can be rewritten as

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = \nabla f(x^*) - \sum_{i=1}^{m} \lambda_i^* \nabla c_i(x^*) = 0. \tag{8.2.29}$$

Conditions (8.2.21) and (8.2.22) are called the feasibility conditions, (8.2.23) is the nonnegativity condition for multipliers, and (8.2.24) is referred to as the complementarity condition which states that both λ_i^* and $c_i(x^*)$ cannot be nonzero, or equivalently that Lagrange multipliers corresponding to inactive constraints are zero.

We say that the strict complementarity condition holds if exactly one of λ_i^* and $c_i(x^*)$ is zero for each $i \in I$, i.e., we have that $\lambda_i^* > 0$ for each $i \in I \cap \mathcal{A}(x^*)$.

An inequality constraint c_i is strongly active if $i \in I \cap \mathcal{A}(x^*)$ and $\lambda_i^* > 0$, i.e., $\lambda_i^* > 0$ and $c_i(x^*) = 0$. An inequality constraint c_i is weakly active if $i \in I \cap \mathcal{A}(x^*)$ and $\lambda_i^* = 0$, i.e., $\lambda_i^* = c_i(x^*) = 0$.

The condition (8.2.19) is called the constraint qualification (CQ). The constraint qualification is important for KKT conditions. As an example given by Fletcher [133], it indicates that if constraint qualification (8.2.19) does not hold, then the local minimizer of problem (8.1.1)–(8.1.3) may not be a KKT point.

Example:

$$\min_{(x_1,x_2)\in R^2} \quad x_1 \tag{8.2.30}$$

$$\text{s.t.} \quad x_1^3 - x_2 \geq 0, \tag{8.2.31}$$

$$x_2 \geq 0. \tag{8.2.32}$$

It is not difficult to see that $x^* = (0,0)^T$ is the global minimizer of (8.2.30)–(8.2.32). At x^*, we have

$$SFD(x^*, X) = \left\{ d \;\middle|\; d = \begin{pmatrix} \alpha \\ 0 \end{pmatrix}, \alpha \geq 0 \right\} \tag{8.2.33}$$

and

$$LFD(x^*, X) = \left\{ d \;\middle|\; d = \begin{pmatrix} \alpha \\ 0 \end{pmatrix}, \alpha \in R^1 \right\}. \tag{8.2.34}$$

Therefore, (8.2.19) does not hold. By direct computing, we have

$$\nabla f(x^*) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \nabla c_1(x^*) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \quad \nabla c_2(x^*) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{8.2.35}$$

which show that there does not exist λ_1^* and λ_2^* such that

$$\nabla f(x^*) = \lambda_1^* \nabla c_1(x^*) + \lambda_2^* \nabla c_2(x^*). \tag{8.2.36}$$

This simple example indicates the importance of constraint qualification. However, it is not easy to examine whether or not the CQ condition (8.2.19) holds. In the following, we give some concrete constraint qualifications which are easy to examine and frequently used.

The most simple and obvious constraint qualification is linear function constraint qualification.

Definition 8.2.8 *If all constraints $c_i(x^*)\,(i \in \mathcal{A}(x^*) = E \cup I(x^*))$ are linear functions, we say that linear function constraint qualification (LFCQ) holds.*

By the definition, if $c_i(x^*)\,(i \in \mathcal{A}(x^*))$ are linear functions, then CQ condition (8.2.19) holds and we have the following corollary.

Corollary 8.2.9 *Let x^* be a local minimizer of problem (8.1.1)-(8.1.3). If the linear function constraint qualification holds at x^*, then x^* is a KKT point.*

The most important and frequently used constraint qualification is the following linear independence constraint qualification (LICQ).

Definition 8.2.10 *If active constraint gradients $\nabla c_i(x^*), i \in \mathcal{A}(x^*)$ are linearly independent, we say that the linear independence constraint qualification (LICQ) holds.*

Theorem 8.2.11 *Let x^* be a feasible point and $\mathcal{A}(x^*)$ an index set of active constraints at x^*. If $\nabla c_i(x^*), i \in \mathcal{A}(x^*)$, are linearly independent, then the constraint qualification (8.2.19) holds.*

Proof. Since $SFD(x^*, X) \subseteq LFD(x^*, X)$, it is enough that we only need to prove $LFD(x^*, X) \subseteq SFD(x^*, X)$. Let $d \in LFD(x^*, X)$ be arbitrary. Now let

$$\mathcal{A}(x^*) = E \cup I(x^*) = \{1, \cdots, l\},\ m_e \leq l \leq n.$$

Since $\nabla c_1(x^*), \cdots, \nabla c_l(x^*)$ are linearly independent, there are $b_{l+1}, \cdots, b_n$ such that $\nabla c_1(x^*), \cdots, \nabla c_l(x^*), b_{l+1}, \cdots, b_n$ are linearly independent.

Consider the nonlinear system

$$r(x, \theta) = 0, \tag{8.2.37}$$

whose components are defined as

$$r_i(x, \theta) = c_i(x) - \theta d^T \nabla c_i(x^*), \qquad i = 1, \cdots, l, \tag{8.2.38}$$

$$r_i(x, \theta) = (x - x^*)^T b_i - \theta d^T b_i, \qquad i = l + 1, \cdots, n. \tag{8.2.39}$$

When $\theta = 0$, the system (8.2.37) is solved by x^*, and when $\theta \geq 0$ is sufficiently small, any solution x is also a feasible point in (8.1.1)–(8.1.3).

Let us write

$$A = [\nabla c_1(x), \cdots, \nabla c_l(x)],\ B = [b_{l+1}, \cdots, b_n].$$

Then the Jacobian matrix $J(x, \theta) = \nabla_x r^T(x, \theta) = [A : B]$. Obviously, $J(x^*) = [A(x^*) : B]$ is nonsingular. Hence by the implicit function theorem there exist open neighborhoods Ω_x about x^* and Ω_θ about $\theta = 0$ such that for any $\theta \in \Omega_\theta$, a unique solution $x(\theta) \in \Omega_x$ exists, and $x(\theta)$ is feasible and continuously differentiable with respect to θ. From (8.2.37) and using the chain rule,

$$0 = \frac{\mathrm{d}r_i}{\mathrm{d}\theta} = \sum_j \frac{\partial r_i}{\partial x_j} \frac{\mathrm{d}x_j}{\mathrm{d}\theta} + \frac{\partial r_i}{\partial \theta},\ i = 1, \cdots, n,$$

that is

$$\nabla c_i(x)^T \frac{\mathrm{d}x}{\mathrm{d}\theta} - \nabla c_i(x^*)^T d = 0,\ i = 1, \cdots, l, \tag{8.2.40}$$

$$b_i^T \frac{\mathrm{d}x}{\mathrm{d}\theta} - b_i^T d = 0,\ i = l + 1, \cdots, n. \tag{8.2.41}$$

The above system is

$$J^T \frac{\mathrm{d}x}{\mathrm{d}\theta} - J(x^*)^T d = 0.$$

Since $x = x^*$ at $\theta = 0$, we have $J = J(x^*)$ at $\theta = 0$. Thus the above equation becomes

$$J(x^*)[\frac{\mathrm{d}x}{\mathrm{d}\theta}|_{\theta=0} - d] = 0.$$

Since the coefficient matrix is nonsingular, we obtain

$$\frac{\mathrm{d}x}{\mathrm{d}\theta} = d \ \text{ at } \theta = 0,$$

which implies that if $\theta_k \downarrow 0$ is any sequence, then $x(\theta_k)$ is a feasible sequence with the feasible direction d, i.e.,

$$\frac{x(\theta_k) - x^*}{\theta_k} \to d.$$

This shows that $d \in SFD(x^*, X)$. Since $d \in LFD(x^*, X)$ is arbitrary, we get $LFD(x^*, X) \subseteq SFD(x^*, X)$. $\square$

By the above theorem and Theorem 8.2.7, we immediately obtain the following theorem.

Theorem 8.2.12 *Let x^* be a local minimizer of problem (8.1.1)–(8.1.3). If LICQ holds, i.e., $\nabla c_i(x^*), i \in \mathcal{A}(x^*) = E \cup I(x^*)$, are linearly independent, then there are Lagrange multipliers λ_i^* $(i = 1, \cdots, m)$ such that (8.2.20)–(8.2.24) hold.*

We want to mention that sometimes we use the regularity assumption

$$SFD(x^*, X) \cap \mathcal{D}(x^*) = LFD(x^*, X) \cap \mathcal{D}(x^*). \tag{8.2.42}$$

Since both sides are subsets of $SFD(x^*, X)$ and $LFD(x^*, X)$ respectively, this assumption is clearly implied by the CQ (8.2.19). However, the converse does not hold.

With the regularity assumption (8.2.42), the necessary condition (8.2.14) in Theorem 8.2.5 (no feasible descent directions: $SFD(x^*, X) \cap \mathcal{D}(x^*) = \phi$) becomes

$$LFD(x^*, X) \cap \mathcal{D}(x^*) = \phi,$$

i.e., there are no linearized feasible descent directions. Furthermore, as a corollary of KKT Theorem 8.2.7, we have

Theorem 8.2.13 *Let x^* be a local minimizer of problem (8.1.1)–(8.1.3). If the regularity assumption (8.2.42) holds, then x^* is a KKT point.*

Next, we discuss the first-order sufficiency condition.

Theorem 8.2.14 *Let $x^* \in X$. Let $f(x)$ and $c_i(x)\,(i = 1, \cdots, m)$ be differentiable at x^*. If*

$$d^T \nabla f(x^*) > 0,\ \forall 0 \neq d \in SFD(x^*, X), \tag{8.2.43}$$

then x^* is a strict local minimizer of problem (8.1.1)–(8.1.3).

Proof. Suppose, by contradiction, that x^* is not a strict local minimizer, then there exist $x_k \in X (k = 1, 2, \cdots)$ such that

$$f(x_k) \leq f(x^*), \tag{8.2.44}$$

and $x_k \to x^*, x_k \neq x^*\,(k = 1, 2, \cdots)$. Without loss of generality, we assume that

$$\frac{x_k - x^*}{\|x_k - x^*\|_2} \to d. \tag{8.2.45}$$

Set $d_k = (x_k - x^*)/\|x_k - x^*\|_2, \delta_k = \|x_k - x^*\|_2$. By Definition 8.2.3, we have

$$d \in SFD(x^*, X). \tag{8.2.46}$$

By use of (8.2.44), (8.2.45) and $f(x_k) = f(x^*) + (x_k - x^*)^T \nabla f(x^*) + o(\|x_k - x^*\|_2)$, by dividing $\|x_k - x^*\|_2$ and then taking the limit as $k \to \infty$, we obtain

$$d^T \nabla f(x^*) \leq 0, \tag{8.2.47}$$

which, together with (8.2.46), contradicts (8.2.43). This completes the proof. □

Since $SFD(x^*, X) \subseteq LFD(x^*, X)$, we also have the following corollary.

Corollary 8.2.15 *Let $x^* \in X$. Let $f(x)$ and $c_i(x)\,(i = 1, \cdots, m)$ be differentiable at x^*. If*

$$d^T \nabla f(x^*) > 0,\ \forall 0 \neq d \in LFD(x^*, X), \tag{8.2.48}$$

then x^* is a strict local minimizer of problem (8.1.1)–(8.1.3).

The other important optimality condition, which is credited to Fritz John [183], is the Fritz John optimality condition, which needs no the constraint qualification.

Theorem 8.2.16 *Let $f(x)$ and $c_i(x)\,(i = 1, \cdots, m)$ be continuously differentiable on a nonempty open set containing the feasible set X. If x^* is a local minimizer of problem (8.1.1)–(8.1.3), then there exist a scalar $\lambda_0^* \geq 0$ and a vector $\lambda^* \in R^m$ such that*

$$\lambda_0^* \nabla f(x^*) - \sum_{i=1}^{m} \lambda_i \nabla c_i(x^*) = 0, \tag{8.2.49}$$

$$c_i(x^*) = 0,\ i \in E, \tag{8.2.50}$$

$$c_i(x^*) \geq 0,\ i \in I, \tag{8.2.51}$$

$$\lambda_i^* \geq 0,\ i \in I, \tag{8.2.52}$$

$$\lambda_i^* c_i(x^*) = 0,\ \forall i, \tag{8.2.53}$$

$$\sum_{i=0}^{m} (\lambda_i^*)^2 > 0. \tag{8.2.54}$$

Proof. If $\nabla c_i(x^*)\,(i \in \mathcal{A}(x^*))$ are linearly dependent, then there are λ_i^* $(i \in \mathcal{A}(x^*))$ not all zero, such that

$$\sum_{i \in \mathcal{A}(x^*)} \lambda_i^* \nabla c_i(x^*) = 0. \tag{8.2.55}$$

Set $\lambda_0^* = 0$ and $\lambda_i^* = 0$, $(i \in I \backslash I(x^*))$, we obtain (8.2.49)–(8.2.54).

If $\nabla c_i(x^*)$ $(i \in \mathcal{A}(x^*))$ are linearly independent, we can obtain immediately (8.2.49)–(8.2.54) with $\lambda_0 = 1$ by means of Theorem 8.2.12. □

The point satisfying (8.2.49)–(8.2.54) is said to be the Fritz John point. The following weighted Lagrangian function

$$\tilde{L}(x, \lambda_0, \lambda) = \lambda_0 f(x) - \sum_{i=1}^{m} \lambda_i c_i(x) \tag{8.2.56}$$

is said to be the Fritz John function. Obviously, the Fritz John point is the stationary point of the Fritz John function. Note that $\lambda_0 \geq 0$. If $\lambda_0 > 0$, the Fritz John function can be regarded as a λ_0 multiple of the Lagrangian function. However, if $\lambda_0 = 0$, the Fritz John function only describes the constraint functions and is independent of the objective function. In such a case, Fritz John optimality conditions do not represent actually the optimality conditions of the original constrained optimization problem. This disadvantage makes the Fritz John conditions unfavorable.

We conclude this section with an optimality condition of convex programming.

As we know, the problem of minimizing a convex function on a convex set Ω is said to be a convex programming problem. Such a problem has the form

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \Omega, \end{aligned} \tag{8.2.57}$$

where $f(x)$ is a convex function on a convex set Ω. Typically, in nonlinear programming

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & c_i(x) = 0,\ i \in E, \\ & c_i(x) \geq 0,\ i \in I, \end{aligned} \tag{8.2.58}$$

if $f(x)$ is convex, $c_i(x)$, $(i \in E)$ are linear functions, and $c_i(x)$, $(i \in I)$ are concave, then the constrained set $\Omega = \{x \mid c_i(x) = 0, i \in E;\ c_i(x) \geq 0, i \in I\}$ is a convex set, and hence the problem (8.2.58) is convex programming.

As Theorem 1.4.9 in the unconstrained case, the following theorem indicates that the local minimizer of convex programming is also its global minimizer.

Theorem 8.2.17 *Each local minimizer of convex programming problem (8.2.57) is also the global minimizer, and the set S of global minimizers is convex.*

Proof. Suppose, by contradiction, that x^* is a local but not global minimizer. Then there exists $x_1 \in \Omega$ such that $f(x_1) < f(x^*)$. Consider

$$x_\theta = (1-\theta)x^* + \theta x_1,\ \theta \in [0,1].$$

By convexity of Ω, $x_\theta \in \Omega$. Also, by convexity of f,

$$\begin{aligned} f(x_\theta) &\leq (1-\theta)f(x^*) + \theta f(x_1) \\ &= f(x^*) + \theta(f(x_1) - f(x^*)) \\ &< f(x^*). \end{aligned}$$

For sufficiently small θ, $x_\theta) \in N(x^*, \epsilon) \cap \Omega$. So, it follows from assumption that x^* is a local minimizer that $f(x_\theta) \geq f(x^*)$. We get a contradiction which means that local minimizers are global.

Let $x_0, x_1 \in S$. Define $x_\theta = (1-\theta)x_0 + \theta x_1, \theta \in [0,1]$. By the global property, $f(x_\theta) \geq f(x_0) = f(x_1)$. However, by convexity of f, $f(x_\theta) \leq (1-\theta)f(x_0) + \theta f(x_1) = f(x_0) = f(x_1)$. Therefore $f(x_\theta) = f(x_0) = f(x_1)$ and so $x_\theta \in S$, which means that S is convex. □

Theorem 8.2.18 *The KKT point of convex programming must be its minimizer.*

Proof. Let (x^*, λ^*) be any KKT pair of convex programming. Obviously, the Lagrangian function

$$L(x, \lambda^*) = f(x) - \sum_{i \in E} \lambda_i^* c_i(x) - \sum_{i \in I} \lambda_i^* c_i(x) \tag{8.2.59}$$

is convex for x. By use of properties of convex function and KKT conditions, we have for any feasible x,

$$\begin{aligned} L(x, \lambda^*) &\geq L(x^*, \lambda^*) + (x - x^*)^T \nabla_x L(x^*, \lambda^*) \\ &= L(x^*, \lambda^*) \\ &= f(x^*) - \sum_{i=1}^{m} \lambda_i^* c_i(x^*) \\ &= f(x^*). \end{aligned} \tag{8.2.60}$$

Note that x is a feasible point and $\lambda_i^* \geq 0, i \in I$, so we have

$$\lambda_i^* c_i(x) = 0, i \in E;\ \lambda_i^* c_i(x) \geq 0, i \in I.$$

Hence

$$L(x, \lambda^*) \leq f(x). \tag{8.2.61}$$

By (8.2.60) and (8.2.61) we obtain

$$f(x) \geq f(x^*), \tag{8.2.62}$$

that is, KKT point x^* is a minimizer. □

Theorem 8.2.19 *The convex programming with strictly convex objective function has unique minimizer.*

The proof of this theorem is as an exercise.

8.3 Second-Order Optimality Conditions

We have seen in unconstrained optimization that the second-order derivative information has significant implications in optimality conditions. Let $x^* \in X$. If

$$d^T \nabla f(x^*) > 0, \, \forall 0 \neq d \in SFD(x^*, X), \tag{8.3.1}$$

then x^* is a strict local minimizer of problem (8.1.1)–(8.1.3). If

$$\text{there exists } d \in SFD(x^*, X) \text{ such that } d^T \nabla f(x^*) < 0, \tag{8.3.2}$$

then from Theorem 8.2.5 it follows that x^* must not be a local minimizer of problem (8.1.1)–(8.1.3). These results tell us that, provided either (8.3.1) or (8.3.2) holds, the first-order optimality condition can be used to determine whether x^* is a local minimizer. However, we cannot determine whether x^* is a local minimizer by the first derivative information alone, if both (8.3.1) and (8.3.2) do not hold, i.e.,

$$d^T \nabla f(x^*) \geq 0, \, \forall d \in SFD(x^*, X); \tag{8.3.3}$$

$$d^T \nabla f(x^*) = 0, \, \exists \, 0 \neq d \in SFD(x^*, X). \tag{8.3.4}$$

In these cases, the second-order derivative information is needed.

Assume that the constraint qualification (8.2.19) holds. It follows from (8.3.3), (8.2.19) and Farkas' Lemma 8.2.6 that x^* is a KKT point. By (8.3.4) and the definition of Lagrange multipliers, there exists $0 \neq d \in SFD(x^*, X)$ such that

$$d^T \nabla f(x^*) = \sum_{i=1}^{m} \lambda_i^* d^T \nabla c_i(x^*) = 0. \tag{8.3.5}$$

Since $SFD(x^*, X) \subseteq LFD(x^*, X)$, by use of Definition 8.2.2, we have that (8.3.5) is equivalent to

$$\lambda_i^* d^T \nabla c_i(x^*) = 0, \, \forall i \in I(x^*). \tag{8.3.6}$$

So, we give the following definitions. Let x^* be a KKT point of (8.1.1)–(8.1.3), and λ^* a corresponding Lagrange multiplier vector. Define a set of strong active constraints as

$$I_+(x^*) = \{i \mid i \in I(x^*) \text{ with } \lambda_i^* > 0\}. \tag{8.3.7}$$

Obviously, $I_+(x^*) \subseteq I(x^*)$.

Definition 8.3.1 *Let x^* be a KKT point of (8.1.1)–(8.1.3), and λ^* a corresponding Lagrange multiplier vector. If there exist sequences $d_k\,(k = 1, 2, \cdots)$ and $\delta_k > 0\,(k = 1, 2, \cdots)$ such that*

$$x^* + \delta_k d_k \in X \tag{8.3.8}$$

satisfy

$$c_i(x_k) = 0, \quad i \in E \cup I_+(x^*), \tag{8.3.9}$$
$$c_i(x_k) \geq 0, \quad i \in I(x^*) \setminus I_+(x^*), \tag{8.3.10}$$

and $d_k \to d$ and $\delta_k \to 0$, then d is said to be a sequential null constraint direction at x^*. The set of all sequential null constraint directions is written as $S(x^*, \lambda^*)$,

$$S(x^*, \lambda^*) = \left\{ d \;\middle|\; \begin{array}{l} x_k = x^* + \delta_k d_k \in X, \delta_k > 0, \delta_k \to 0, d_k \to d, \\ c_i(x_k) = 0,\ i \in E \cup I_+(x^*), \\ c_i(x_k) \geq 0,\ i \in I(x^*) - I_+(x^*). \end{array} \right\}. \tag{8.3.11}$$

It is easy to see that (8.3.9)–(8.3.10) imply that

$$\sum_{i=1}^{m} \lambda_i^* c_i(x^* + \delta_k d_k) = 0. \tag{8.3.12}$$

So, equivalently,

$$S(x^*, \lambda^*) = \left\{ d \;\middle|\; \begin{array}{l} d \in SFD(x^*, X); \\ \sum_{i=1}^{m} \lambda_i^* c_i(x_k) = 0. \end{array} \right\}. \tag{8.3.13}$$

Obviously, $S(x^*, \lambda^*) \subseteq SFD(x^*, X)$.

Similar to the linearized feasible direction, we have the following definition.

Definition 8.3.2 *Let x^* be a KKT point of (8.1.1)–(8.1.3), and λ^* a corresponding Lagrange multiplier vector. If d is a linearized feasible direction at x^* and (8.3.6) holds, then d is said to be a linearized null constraint direction. The set of all linearized null constraint directions is written as $G(x^*, \lambda^*)$,*

$$G(x^*, \lambda^*) = \left\{ d \;\middle|\; \begin{array}{l} d \neq 0, \\ d^T \nabla c_i(x^*) = 0, \quad i \in E \cup I_+(x^*), \\ d^T \nabla c_i(x^*) \geq 0, \quad i \in I(x^*) \setminus I_+(x^*). \end{array} \right\}. \tag{8.3.14}$$

Equivalently,

$$G(x^*, \lambda^*) = \left\{ d \;\middle|\; \begin{array}{l} d \in LFD(x^*, \lambda^*); \\ d^T \nabla c_i(x^*) = 0, i \in I_+(x^*). \end{array} \right\} \tag{8.3.15}$$

If the Lagrange multiplier at x^* is unique, $G(x^*, \lambda^*)$ can be denoted by $G(x^*)$.

By the above definitions, we have

$$S(x^*, \lambda^*) \subseteq SFD(x^*, X), \tag{8.3.16}$$
$$G(x^*, \lambda^*) \subseteq LFD(x^*, X). \tag{8.3.17}$$

Similar to $SFD(x^*, X) \subseteq LFD(x^*, X)$, we also can prove

$$S(x^*, \lambda^*) \subseteq G(x^*, \lambda^*), \tag{8.3.18}$$

which is an exercise left to readers.

Now, we state the main results of this section.

Theorem 8.3.3 *(Second-order necessary conditions)*
Let x^* be a local minimizer of (8.1.1)–(8.1.3). If the constraint qualification (8.2.19) holds, then we have

$$d^T \nabla^2_{xx} \mathcal{L}(x^*, \lambda^*) d \geq 0, \ \forall d \in S(x^*, \lambda^*), \tag{8.3.19}$$

where $\mathcal{L}(x, \lambda)$ is a Lagrangian function.

Furthermore, if

$$S(x^*, \lambda^*) = G(x^*, \lambda^*), \tag{8.3.20}$$

then

$$d^T \nabla^2_{xx} \mathcal{L}(x^*, \lambda^*) d \geq 0, \ \forall d \in G(x^*, \lambda^*). \tag{8.3.21}$$

Proof. For any $d \in S(x^*, \lambda^*)$, if $d = 0$, it is obvious that $d^T \nabla^2_{xx} \mathcal{L}(x^*, \lambda^*) d = 0$. Now we consider $d \neq 0$. From the definition of $S(x^*, \lambda^*)$, there exist $\{d_k\}$ and $\{\delta_k\}$ such that (8.3.8)–(8.3.12) hold. Therefore, by (8.3.12) and KKT conditions, we have

$$\begin{aligned} f(x^* + \delta_k d_k) &= \mathcal{L}(x^* + \delta_k d_k, \lambda^*) \\ &= \mathcal{L}(x^*, \lambda^*) + \frac{1}{2} \delta_k^2 d_k^T \nabla^2_{xx} \mathcal{L}(x^*, \lambda^*) d_k + o(\delta_k^2) \\ &= f(x^*) + \frac{1}{2} \delta_k^2 d_k^T \nabla^2_{xx} \mathcal{L}(x^*, \lambda^*) d_k + o(\delta_k^2). \end{aligned} \tag{8.3.22}$$

Since x^* is a local minimizer, it follows for all k sufficiently large that

$$f(x^* + \delta_k d_k) \geq f(x^*). \tag{8.3.23}$$

Using (8.3.22)–(8.3.23) and taking limits give

$$d^T \nabla^2_{xx} \mathcal{L}(x^*, \lambda^*) d \geq 0.$$

Since $d \in S(x^*, \lambda^*)$ is arbitrary, then (8.3.19) follows.

By (8.3.20), we immediately obtain (8.3.21) from (8.3.19). □

Theorem 8.3.4 *(Second-order sufficient conditions)*
Let x^* be a KKT point of (8.1.1)–(8.1.3). If

$$d^T \nabla^2_{xx} \mathcal{L}(x^*, \lambda^*) d > 0, \ \forall d \in G(x^*, \lambda^*), \tag{8.3.24}$$

then x^* is a strict local minimizer.

Proof. Assume that x^* is not a strict local minimizer, then there exists a sequence $\{x_k\} \subset X$ such that

$$f(x_k) \leq f(x^*), \tag{8.3.25}$$

with $x_k \to x^*$ and $x_k \neq x^* (k = 1, 2, \cdots)$. Without loss of generality, we assume that

$$\frac{x_k - x^*}{\|x_k - x^*\|_2} \to d.$$

By a similar argument to (8.2.45)–(8.2.47), we have

$$d^T \nabla f(x^*) \leq 0 \tag{8.3.26}$$

and

$$d \in SFD(x^*, X) \subseteq LFD(x^*, X). \tag{8.3.27}$$

It follows by KKT conditions and (8.2.4) that

$$d^T \nabla f(x^*) = \sum_{i=1}^{m} \lambda_i d^T \nabla c_i(x^*) \geq 0. \tag{8.3.28}$$

Note that (8.3.26) and (8.3.28) give

$$d^T \nabla f(x^*) = 0 \tag{8.3.29}$$

which implies from (8.3.28) and Definition 8.2.2 that

$$\lambda_i d^T \nabla c_i(x^*) = 0, \ \forall i \in I(x^*). \tag{8.3.30}$$

So, it follows from (8.3.27), (8.3.30) and Definition 8.3.2 that

$$d \in G(x^*, \lambda^*). \tag{8.3.31}$$

From (8.3.25), we have

$$\begin{aligned} \mathcal{L}(x^*, \lambda^*) &\geq \mathcal{L}(x_k, \lambda^*) \\ &= \mathcal{L}(x^*, \lambda^*) + \frac{1}{2}\delta_k^2 d_k^T \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*) d_k + o(\delta_k^2). \end{aligned} \tag{8.3.32}$$

Dividing by δ_k^2 and taking the limit give

$$d^T \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*) d \leq 0 \tag{8.3.33}$$

which contradicts (8.3.24). We complete the proof. □

Notice that a sufficient condition for (8.3.24) is that

$$d^T \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*) d > 0$$

for all $d \neq 0$ such that $d^T \nabla c_i(x^*) = 0, \ i \in \mathcal{A}_+(x^*, \lambda^*)$, where

$$\mathcal{A}_+(x^*, \lambda^*) = E \cup \{i \,|\, i \in I(x^*), \lambda_i^* > 0\}, \tag{8.3.34}$$

which is obtained by deleting indices for which $\lambda_i^* = 0, \ i \in I(x^*)$ from $\mathcal{A}(x^*)$. The $\mathcal{A}_+(x^*, \lambda^*)$ is said to be an index set of strong active constraints, which is a union of the index sets of equality constraints and strongly active inequality constraints. So, we immediately obtain the following corollary which is also a second-order sufficient condition and more convenient to verify in practice.

Corollary 8.3.5 *Let x^* be a KKT point of (8.1.1)–(8.1.3). If*

$$d^T \nabla_{xx}^2 \mathcal{L}(x^*, d^*) d > 0 \tag{8.3.35}$$

for all d satisfying

$$d^T \nabla c_i(x^*) = 0, \ \forall i \in \mathcal{A}_+(x^*, \lambda^*), \tag{8.3.36}$$

then x^* is a strict local minimizer.

Proof. It is enough to prove that (8.3.35)–(8.3.36) are the sufficient conditions of (8.3.24). In fact, if x^* is a KKT point, then $\forall d \in SFD(x^*, X) \subseteq LFD(x^*, X)$,

$$d^T \nabla c_i(x^*) = 0, \qquad i \in E \tag{8.3.37}$$

$$d^T \nabla c_i(x^*) \geq 0, \qquad i \in I(x^*). \tag{8.3.38}$$

So, (8.3.6) holds, which implies $d \in G(x^*, \lambda^*)$ from Definition 8.3.2. Therefore, by means of Theorem 8.3.4, it follows that (8.3.35)–(8.3.36) implies (8.3.24). □

8.4 Duality

We conclude this chapter with a brief discussion of duality. The concept of duality occurs widely in the mathematical programming literature. The aim is to provide an alternative formulation of a mathematical programming problem which is more convenient computationally or has some theoretical significance.

The original problem is referred to as the primal, and the transformed problem is referred to as the dual.

In this section, we give an introduction of duality theory which is associated with the convex programming problem. We will introduce the Lagrangian dual problem, and prove the duality theorem and the weak duality theorem. Now we first state the duality theorem.

Theorem 8.4.1 *Let x^* be a minimizer of convex primal problem (P)*

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & c_i(x) \geq 0,\ i = 1, \cdots, m. \end{aligned} \tag{8.4.1}$$

If $f(x)$ and $c_i(x)$, $(i = 1, \cdots, m)$ are continuously differentiable and the regularity condition (8.2.42) holds, then x^* and λ^* solve the dual problem

$$\begin{aligned} \max_{x,\lambda} \quad & \mathcal{L}(x, \lambda) \\ \text{s.t.} \quad & \nabla_x \mathcal{L}(x, \lambda) = 0, \\ & \lambda \geq 0. \end{aligned} \tag{8.4.2}$$

Furthermore, the minimum of the primal and the maximum of the dual are equal, i.e.,

$$f(x^*) = \mathcal{L}(x^*, \lambda^*). \tag{8.4.3}$$

Proof. By the assumption and KKT Theorem 8.2.7, there exist Lagrange multipliers $\lambda^* \geq 0$ such that $\nabla_x \mathcal{L}(x^*, \lambda^*) = 0$ and $\lambda_i^* c_i(x^*) = 0$, $i = 1, \cdots, m$. Thus, $f(x^*) = \mathcal{L}(x^*, \lambda^*)$.

Let x, λ be dual feasible. Using $\lambda \geq 0$, convexity of $\mathcal{L}$, and $\nabla_x \mathcal{L}(x, \lambda) = 0$ gives

$$\begin{aligned} \mathcal{L}(x^*, \lambda^*) &= f(x^*) \geq f(x^*) - \sum_{i=1}^{m} \lambda_i c_i(x^*) \\ &= \mathcal{L}(x^*, \lambda) \\ &\geq \mathcal{L}(x, \lambda) + (x^* - x)^T \nabla_x \mathcal{L}(x, \lambda) \\ &= \mathcal{L}(x, \lambda) \end{aligned} \tag{8.4.4}$$

which means that (x^*, λ^*) solves the dual problem. □

Usually, (8.4.3) is said to be the strong duality. Now, we give some examples of dual problems. Let the primal problem in linear programming be

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & A^T x \geq b. \end{aligned} \tag{8.4.5}$$

By Theorem 8.4.1, we immediately have the dual:

$$\begin{aligned} \max \quad & b^T \lambda \\ \text{s.t.} \quad & A\lambda = c, \\ & \lambda \geq 0. \end{aligned} \tag{8.4.6}$$

Normally, linear programs have standard form:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0. \end{aligned} \tag{8.4.7}$$

The corresponding dual problem is

$$\begin{aligned} \max \quad & b^T \lambda \\ \text{s.t.} \quad & A^T \lambda \leq c. \end{aligned} \tag{8.4.8}$$

It is easy to examine that the optimality conditions of (8.4.7) and (8.4.8) are identical.

For convex quadratic programming, the primal problem is

$$\begin{aligned} \min_{x} \quad & \frac{1}{2}x^T Gx + h^T x \\ \text{s.t.} \quad & A^T x \geq b, \end{aligned} \tag{8.4.9}$$

where G is positive definite. The dual problem is

$$\begin{aligned} \max_{x,\lambda} \quad & \frac{1}{2}x^T Gx + h^T x - \lambda^T (A^T x - b) \\ \text{s.t.} \quad & Gx + h - A\lambda = 0, \qquad (8.4.10) \\ & \lambda \geq 0. \qquad (8.4.11) \end{aligned}$$

By eliminating x, we obtain the following problem:

$$\begin{aligned} \max_{\lambda} \quad & -\frac{1}{2}\lambda^T (A^T G^{-1} A)\lambda + \lambda^T (b - A^T G^{-1} h) - \frac{1}{2} h^T G^{-1} h \\ \text{s.t.} \quad & \lambda \geq 0. \end{aligned} \tag{8.4.12}$$

This is a quadratic programming problem in λ with bounded-constraints $\lambda \geq 0$.

The following theorem, referred to as the weak duality theorem, shows that the objective value of any feasible point of the primal problem is larger than or equal to the objective value of any feasible point of the dual problem.

Theorem 8.4.2 *Let x' be any feasible point in primal problem (8.4.1). Let (x, λ) be any feasible point in dual problem (8.4.2). Then*

$$f(x') \geq \mathcal{L}(x, \lambda). \tag{8.4.13}$$

Proof. Let x' be primal feasible and (x, λ) dual feasible. Then by convexity of f, dual feasibility, concavity of c_i, and nonnegativity of $c_i(x')$ and λ_i in turn, it follows that

$$\begin{aligned} f(x') - f(x) &\geq \nabla f(x)^T (x' - x) \\ &= \sum_{i=1}^{m} \lambda_i \nabla c_i(x)^T (x' - x) \\ &\geq \sum_{i=1}^{m} \lambda_i (c_i(x') - c_i(x)) \\ &\geq -\sum_{i=1}^{m} \lambda_i c_i(x). \end{aligned}$$

Hence

$$f(x') \geq f(x) - \sum_{i=1}^{m} \lambda_i c_i(x) = \mathcal{L}(x, \lambda). \quad \square$$

From the above theorem, we immediately have

$$\inf_x f(x) \geq \sup_{x,\lambda} \mathcal{L}(x, \lambda). \tag{8.4.14}$$

This implies that if the primal problem is unbounded, it follows that $\inf_x f(x) = \sup_{x,\lambda} \mathcal{L}(x, \lambda) = -\infty$, and this is not possible if (x, λ) is feasible. Therefore, an unbounded primal implies an inconsistent dual.

Exercises

1. Assume that $f(x)$ is a convex function, $c_i(x)$ $(1 \leq i \leq m_e)$ are linear functions and $c_i(x)$ $(m_e + 1 \leq i \leq m)$ are concave functions. Show that x^* is a global minimizer of (8.1.1)–(8.1.3) if it is a local minimizer of (8.1.1)–(8.1.3).

2. Define the ϵ-active set by

$$I_\epsilon(x) = \{i \mid c_i(x) \leq \epsilon, \quad i \in I(x)\}.$$

Prove that, for any given $x \in \Re^n$,

$$\lim_{\epsilon \to 0_+} I_\epsilon(x) = I(x).$$

3. Prove: if $c_i(x)\,(i \in \mathcal{A}(x^*))$ are linear functions, then CQ condition (8.2.19) holds.

4. Prove (8.3.16) and (8.3.17).

5. Prove (8.3.18).

6. Let $0 \neq c \in \Re^n$. Consider the problem

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & \|x\|_2^2 \leq 1. \end{array}$$

Prove that $x^* = c/\|c\|_2$ satisfies the second-order sufficient condition.

7. Form the KKT conditions for

$$\begin{array}{ll} \max & (x+1)^2 + (y+1)^2 \\ \text{s.t.} & x^2 + y^2 \leq 2, \\ & 1 - y \geq 0 \end{array}$$

and then determine the solution.

8. Give an example in which the second-order necessary condition holds while the second-order sufficient condition fails.

9. By solving the KKT equation, find the point on the ellipse defined by the intersection of the surface $x + y = 1$ and $x^2 + 2y^2 + z^2 = 1$ which is nearest to the origin.

10. Show that the dual of problem

$$\begin{array}{ll} \min & \dfrac{1}{2}\sigma x_1^2 + \dfrac{1}{2}x_2^2 + x_1 \\ \text{s.t.} & x_1 \geq 0 \end{array}$$

is a maximization problem in terms of a Largrange multiplier λ. For the case $\sigma = +1$ and $\sigma = -1$, investigate whether the local solution of the dual gives the multiplier λ^* which exists at the local solution to the primal.

Chapter 9

Quadratic Programming

9.1 Optimality Conditions for Quadratic Programming

Quadratic programming is the simplest constrained nonlinear optimization problem. It is a special class of optimization problem (8.1.1)–(8.1.3) with a quadratic objective function $f(x)$ and linear constraints $c_i(x)$ $(i = 1, \cdots, m)$. The general quadratic programming (QP) has the following form:

$$\min \quad Q(x) = \frac{1}{2}x^T Gx + g^T x \tag{9.1.1}$$

$$\text{s.t.} \quad a_i^T x = b_i,\ i \in E, \tag{9.1.2}$$

$$a_i^T x \geq b_i,\ i \in I, \tag{9.1.3}$$

where G is a symmetric $n \times n$ matrix, E and I are finite sets of indices, $E = \{1, \cdots, m_e\}$ and $I = \{m_e+1, \cdots, m\}$. If the Hessian matrix G is positive semi-definite, then (9.1.1)–(9.1.3) is a convex quadratic programming problem and the local solution x^* is a global solution. If G is positive definite, then (9.1.1)–(9.1.3) is a strict convex QP and x^* is a unique global solution. If G is indefinite, then (9.1.1)–(9.1.3) is a nonconvex QP which is more important and worth emphasizing.

From Theorem 8.2.7, Theorem 8.3.3 and Theorem 8.3.4, we immediately get the following theorems:

Theorem 9.1.1 *(Necessary conditions)*

Let x^* be a local minimizer of quadratic programming problem (9.1.1)–(9.1.3). Then there exist multipliers $\lambda_i^*\,(i = 1, \cdots, m)$ such that

$$g + Gx^* = \sum_{i=1}^{m} \lambda_i^* a_i, \tag{9.1.4}$$

$$a_i^T x^* = b_i,\ i \in E, \tag{9.1.5}$$

$$a_i^T x^* \geq b_i,\ i \in I, \tag{9.1.6}$$

$$\lambda_i^*(a_i^T x^* - b_i) = 0,\ i \in I, \tag{9.1.7}$$

$$\lambda_i^* \geq 0,\ i \in I. \tag{9.1.8}$$

Furthermore,

$$d^T G d \geq 0,\ \forall d \in G(x^*, \lambda^*), \tag{9.1.9}$$

where

$$G(x^*, \lambda^*) = \left\{ d \neq 0 \;\middle|\; \begin{array}{l} d^T a_i = 0,\ i \in E \\ d^T a_i \geq 0,\ i \in I(x^*) \\ d^T a_i = 0,\ i \in I(x^*) \text{ and } \lambda_i^* > 0 \end{array} \right\}. \tag{9.1.10}$$

Theorem 9.1.2 *(Sufficient conditions)*

Let x^* be a KKT point and λ^* a corresponding Lagrange multiplier vector. If $d^T G d > 0\ \forall 0 \neq d \in G(x^*, \lambda^*)$, then x^* is a strict local minimizer to (9.1.1)–(9.1.3).

Next, we give a sufficient and necessary optimality condition for (9.1.1)–(9.1.3).

Theorem 9.1.3 *(Necessary and sufficient conditions)*

Let x^* be a feasible point of quadratic programming problem (9.1.1)–(9.1.3), then x^* is a local minimizer if and only if (x^*, λ^*) is a KKT pair such that (9.1.4)–(9.1.8) hold, and

$$d^T G d \geq 0,\ \forall d \in G(x^*, \lambda^*). \tag{9.1.11}$$

Proof. Let x^* be a local minimizer, it follows from Theorem 9.1.1 that there exists multiplier vector λ^* such that (9.1.4)–(9.1.8) hold. Let $0 \neq d \in G(x^*, \lambda^*)$. Obviously, for sufficiently small $t > 0$, we have

$$x^* + td \in X. \tag{9.1.12}$$

Then, by the definition of d, for sufficiently small $t > 0$, we have

$$\begin{aligned} Q(x^*) &\leq Q(x^* + td) = Q(x^*) + td^T(Gx^* + g) + \frac{1}{2}t^2 d^T G d \\ &= Q(x^*) + t\sum_{i=1}^{m} \lambda_i^* a_i^T d + \frac{1}{2}t^2 d^T G d \\ &= Q(x^*) + \frac{1}{2}t^2 d^T G d, \end{aligned} \tag{9.1.13}$$

which, together with the arbitrariness of d, means (9.1.11) holds.

Second, we prove the sufficiency. Suppose, by contradiction, that x^* is not a local minimizer, so that there exists $x_k = x^* + \delta_k d_k \in X$ such that

$$Q(x_k) = Q(x^* + \delta_k d_k) < Q(x^*), \tag{9.1.14}$$

where $\delta_k > 0, \delta_k \to 0, d_k \to \bar{d}$. Completely similar to the proof of Theorem 8.3.4, we know that

$$\bar{d} \in G(x^*, \lambda^*). \tag{9.1.15}$$

Thus, it follows from (9.1.14) and KKT conditions that

$$\begin{aligned} \mathcal{L}(x^*, \lambda^*) &> \mathcal{L}(x_k, \lambda^*) \\ &= \mathcal{L}(x^*, \lambda^*) + \frac{1}{2}\delta_k^2 d_k^T G d_k + o(\delta_k^2). \end{aligned} \tag{9.1.16}$$

Dividing both sides by δ_k^2 and taking the limit, we obtain

$$\bar{d}^T G \bar{d} < 0. \tag{9.1.17}$$

Noting that $\bar{d} \in G(x^*, \lambda^*)$, it follows that (9.1.17) contradicts the assumption (9.1.11). Then we complete the proof. □

Obviously, finding the KKT point of a quadratic programming problem is equivalent to finding $x^* \in R^n, \lambda^* \in R^m$ such that (9.1.4)–(9.1.8) hold.

9.2 Duality for Quadratic Programming

In this section we give more detailed discussion on the duality of convex quadratic programming, because in some classes of practical problems we can take advantage of the special structure of the dual to solve the problems more efficiently.

Assume that G is a positive definite matrix. From the results in §9.1, we have known that solving the quadratic programming problem (9.1.1)–(9.1.3) is equivalent to solving (9.1.4)–(9.1.8). Write

$$y = A\lambda - g, \tag{9.2.1}$$

and

$$t_i = a_i^T x - b_i,\ i \in I, \tag{9.2.2}$$

where $A = [a_1, \cdots, a_m] \in R^{n\times m}, \lambda = [\lambda_1, \cdots, \lambda_m]^T \in R^m$. Note that (9.1.4) is just $y = Gx$ and that (9.1.5)–(9.1.6) become

$$A^T x - b = (0, \cdots, 0, t_{m_e+1}, \cdots, t_m)^T,$$

then (9.1.4)–(9.1.8) can be written as

$$\begin{bmatrix} -b \\ G^{-1}y \end{bmatrix} = \begin{bmatrix} -A^T \\ I \end{bmatrix} x + (0, \cdots, 0, t_{m_e+1}, \cdots, t_m, 0, \cdots, 0)^T \tag{9.2.3}$$

$$A\lambda - y = g, \tag{9.2.4}$$

$$\lambda_i \geq 0,\ i \in I, \tag{9.2.5}$$

$$t_i\lambda_i = 0,\ i \in I, \tag{9.2.6}$$

$$t_i \geq 0,\ i \in I. \tag{9.2.7}$$

By KKT conditions, it follows that (9.2.3)–(9.2.7) are equivalent to solving the problem

$$\max_{\lambda,y} \quad b^T\lambda - \frac{1}{2}y^T G^{-1} y \stackrel{Def}{=} \bar{Q}(\lambda, y) \tag{9.2.8}$$

$$\text{s.t.} \quad A\lambda - y = g, \tag{9.2.9}$$

$$\lambda_i \geq 0,\ i \in I, \tag{9.2.10}$$

which is the dual of the primal (9.1.1)–(9.1.3). As an exercise, please prove that problem (9.2.8)–(9.2.10) just is

$$\max_{x,\lambda} \quad L(x, \lambda) = \frac{1}{2}x^T Gx + g^T x - \lambda^T(A^T x - b) \tag{9.2.11}$$

$$\text{s.t.} \quad \nabla_x L(x, \lambda) = 0 \tag{9.2.12}$$

$$\lambda_i \geq 0,\ i \in I. \tag{9.2.13}$$

Eliminating y in (9.2.9) by use of (9.2.1), we get that (9.2.8)–(9.2.10) can be reduced to

$$\min_{\lambda \in R^m} \quad -(b + A^T G^{-1} g)^T \lambda + \frac{1}{2} \lambda^T (A^T G^{-1} A) \lambda \tag{9.2.14}$$

$$\text{s.t.} \quad \lambda_i \geq 0, \ i \in I. \tag{9.2.15}$$

Assume that x and (λ, y) are feasible points of the primal problem (9.1.1)–(9.1.3) and the dual problem (9.2.8)–(9.2.10) respectively, then we have

$$\begin{aligned} Q(x) - \bar{Q}(\lambda, y) &= x^T (A\lambda - y) + \frac{1}{2} x^T G x \\ &\quad -(\lambda^T A x - \sum_{i \in I} \lambda_i t_i - \frac{1}{2} y^T G^{-1} y) \\ &= \sum_{i \in I} \lambda_i t_i + \frac{1}{2} (x^T G x + y^T G^{-1} y - 2 x^T y), \end{aligned} \tag{9.2.16}$$

where t_i is defined in (9.2.2). Then, the positive definiteness of G gives

$$Q(x) \geq \bar{Q}(\lambda, y), \tag{9.2.17}$$

which is what we showed in Theorem 8.4.2.

It also follows from (9.2.16) that both sides of (9.2.17) are equal if and only if

$$\sum_{i \in I} \lambda_i (a_i^T x - b_i) = 0 \tag{9.2.18}$$

and

$$x = G^{-1} y. \tag{9.2.19}$$

It is not difficult to see that (9.2.19) and (9.2.18) are equivalent to (9.1.4) and (9.1.7) respectively. So, with the assumptions of feasibility, we have the following theorem.

Theorem 9.2.1 *Let G be positive definite. If the primal problem is feasible, then $x^* \in X$ is a solution of primal problem (9.1.1)–(9.1.3) if and only if (λ^*, y^*) is the solution of dual problem (9.2.8)–(9.2.10).*

In §8.4, we have shown that an unbounded primal implies an infeasible dual. We would like to know whether or not an infeasible primal implies an unbounded dual. This guess does not always hold. However, it is true for linearly constrained problems.

Theorem 9.2.2 *Let G be positive definite. Then the primal problem (9.1.1)–(9.1.3) is infeasible if and only if the dual (9.2.8)–(9.2.10) is unbounded.*

Proof. From (9.2.17), if the primal problem is feasible, the objective function of the dual problem on set satisfying constraints (9.2.9)–(9.2.10) is uniformly bounded above.

Now suppose that the primal problem is infeasible, then the system

$$(a_i^T, b_i)\tilde{x} = 0,\ i \in E, \tag{9.2.20}$$

$$(a_i^T, b_i)\tilde{x} \geq 0,\ i \in I, \tag{9.2.21}$$

$$(0, \cdots, 0, 1)\tilde{x} < 0 \tag{9.2.22}$$

has no solution for $\tilde{x} \in R^{n+1}$. By Corollary 8.2.6 of Farkas' Lemma, it follows that there exist $\bar{\lambda}_i\,(i = 1, \cdots, m)$ such that

$$(0, \cdots, 0, 1) = \sum_{i \in E} \bar{\lambda}_i (a_i^T, b_i) + \sum_{i \in I} \bar{\lambda}_i (a_i^T, b_i), \tag{9.2.23}$$

i.e.,

$$\sum_{i=1}^{m} \bar{\lambda}_i a_i = 0, \tag{9.2.24}$$

$$\sum_{i=1}^{m} \bar{\lambda}_i b_i = 1, \tag{9.2.25}$$

$$\bar{\lambda}_i \geq 0,\ i \in I. \tag{9.2.26}$$

Set $\lambda_i = t\bar{\lambda}_i$, then (9.2.24) gives $A\lambda = tA\bar{\lambda} = 0$. It follows from (9.2.9) that $y = -g$. Therefore, when $t \to +\infty$, it follows from (9.2.8) and (9.2.25) that

$$\bar{Q}(\lambda, y) = t \to +\infty.$$

Also, for all $t > 0$, we have that $\lambda = (t\bar{\lambda}_1, \cdots, t\bar{\lambda}_m)^T$ and $y = -g$ satisfy constraints (9.2.9)–(9.2.10) of the dual problem. This shows that the dual problem is unbounded. □

There is a closed connection between Lagrangian function

$$\mathcal{L}(x, \lambda) = Q(x) - \sum_{i=1}^{m} \lambda_i (a_i^T x - b_i) \tag{9.2.27}$$

of the primal problem and duality. It is not difficult to see that solving KKT conditions is equivalent to finding a stationary point of $\mathcal{L}(x, \lambda)$ on the area $\{(x, \lambda) \mid \lambda_i \geq 0,\ i \in I\}$. Since the Hessian matrix of $\mathcal{L}(x, \lambda)$ is

$$\nabla^2 \mathcal{L}(x, \lambda) = \begin{bmatrix} G & -A \\ -A^T & 0 \end{bmatrix}, \tag{9.2.28}$$

by use of the identity

$$\begin{bmatrix} I & 0 \\ A^T G^{-1} & I \end{bmatrix} \nabla^2 \mathcal{L}(x, \lambda) \begin{bmatrix} I & G^{-1}A \\ 0 & I \end{bmatrix} = \begin{bmatrix} G & 0 \\ 0 & -A^T G^{-1} A \end{bmatrix}, \tag{9.2.29}$$

we know that $\nabla^2 \mathcal{L}(x, \lambda)$ has just n positive eigenvalues, and that the number of negative eigenvalues equals rank(A). Thus, in general, the stationary point of $\mathcal{L}(x, \lambda)$ is a saddle point, i.e., there is $\lambda^* \in \Lambda$,

$$\Lambda = \{\lambda \in R^m \mid \lambda_i \geq 0,\ i \in I\},$$

such that (x^*, λ^*) satisfies

$$\mathcal{L}(x^*, \lambda) \leq \mathcal{L}(x^*, \lambda^*) \leq \mathcal{L}(x, \lambda^*) \tag{9.2.30}$$

for all $x \in X$ and $\lambda \in \Lambda$.

In fact, for all $x \in X$, we have

$$\max_{\lambda \in \Lambda} \mathcal{L}(x, \lambda) = Q(x). \tag{9.2.31}$$

For any $\lambda \in \Lambda$, set

$$y = A\lambda - g, \tag{9.2.32}$$

then (λ, y) is a feasible point of dual problem (9.2.8)–(9.2.10). This means that such a feasible (λ, y) satisfies (9.2.9), which is $\nabla_x \mathcal{L}(x, \lambda) = 0$, i.e., such a feasible (λ, y) such that $\min_{x \in R^n} \mathcal{L}(x, \lambda)$. Therefore,

$$\min_{x \in R^n} \mathcal{L}(x, \lambda) = b^T \lambda - \frac{1}{2} y^T G^{-1} y = \bar{Q}(\lambda, y). \tag{9.2.33}$$

Let (x^*, λ^*) be a solution of (9.1.4)–(9.1.8). Let $y^* = A\lambda^* - g$. It follows that (λ^*, y^*) is a feasible point of (9.2.8)–(9.2.10). Then, for any $x \in X$ and any $\lambda \in \Lambda$, we have

$$\begin{aligned} \mathcal{L}(x, \lambda^*) &\geq \bar{Q}(\lambda^*, y^*) \\ &= \mathcal{L}(x^*, \lambda^*) = Q(x^*) \geq \mathcal{L}(x^*, \lambda), \end{aligned} \tag{9.2.34}$$

which means that (x^*, λ^*) is a saddle point of $\mathcal{L}(x, \lambda)$.

Conversely, if

$$\mathcal{L}(x^*, \lambda) \leq \mathcal{L}(x^*, \lambda^*) \leq \mathcal{L}(x, \lambda^*) \tag{9.2.35}$$

holds for all $x \in X$ and $\lambda \in \Lambda$, then

$$\begin{aligned} Q(x^*) - \sum_{i=1}^{m} \lambda_i (a_i^T x^* - b_i) &\leq Q(x^*) - \sum_{i=1}^{m} \lambda_i^* (a_i^T x^* - b_i) \\ &\leq Q(x) - \sum_{i=1}^{m} \lambda_i^* (a_i^T x - b_i). \end{aligned} \tag{9.2.36}$$

Rearranging the first inequality gives

$$\sum_{i=1}^{m} (\lambda_i - \lambda_i^*)(a_i^T x^* - b_i) \geq 0, \tag{9.2.37}$$

which is

$$\sum_{i \in E} (\lambda_i - \lambda_i^*)(a_i^T x^* - b_i) + \sum_{i \in I} (\lambda_i - \lambda_i^*)(a_i^T x^* - b_i) \geq 0. \tag{9.2.38}$$

Now we prove

$$a_i^T x^* = b_i, \qquad i \in E \tag{9.2.39}$$

$$a_i^T x^* \geq b_i, \qquad i \in I \tag{9.2.40}$$

by contradiction. Suppose that $a_k^T x^* > b_k$ for some $k \in E$. Set $\lambda_i = \lambda_i^*$ for $i \neq k$ and $\lambda_k = \lambda_k^* - 1$, then we get a contradiction from (9.2.38) to the assumption $a_k^T x^* > b_k$. If we suppose $a_k^T x^* < b_k$ for some $k \in E$, we can get a similar contradiction. Therefore, we have that $a_i^T x^* = b_i$, $\forall i \in E$.

Now assume, for some $k \in I$, that

$$\lambda_k = \lambda_k^* + 1 \text{ and } \lambda_i = \lambda_i^* \text{ for } i \neq k. \tag{9.2.41}$$

Obviously, it follows that

$$a_k^T x^* - b_k \geq 0, \ k \in I. \tag{9.2.42}$$

Repeating the process for all $k \in I$, we obtain

$$a_i^T x^* - b_i \geq 0, \ \forall i \in I. \tag{9.2.43}$$

Then, x^* is a feasible point to the primal problem.

Set $\lambda = 0$, it follows from (9.2.35) that $\mathcal{L}(x^*, \lambda^*) \geq \mathcal{L}(x^*, 0)$, which is

$$\sum_{i=1}^{m} \lambda_i^* (a_i^T x^* - b_i) \leq 0. \tag{9.2.44}$$

By use of (9.2.44), (9.2.35) and $\lambda^* \in \Lambda$, we have, for all $x \in X$,

$$\begin{aligned} Q(x^*) &\leq Q(x^*) - \sum_{i=1}^{m} \lambda_i^* (a_i^T x^* - b_i) \\ &= \mathcal{L}(x^*, \lambda^*) \\ &\leq \mathcal{L}(x, \lambda^*) \\ &\leq \mathcal{L}(x, \lambda^*) + \sum_{i=1}^{m} \lambda_i^* (a_i^T x - b_i) \\ &= Q(x), \end{aligned} \tag{9.2.45}$$

which shows that x^* is a minimizer of the primal problem.

Therefore, we get the following theorem which is a famous saddle point theorem on the relationship between the saddle point of a Lagrangian function and the minimizer of the primal problem.

Theorem 9.2.3 *(Saddle point theorem for quadratic programming)*
Let G be positive definite. Then $x^* \in X$ is a minimizer of the primal problem (9.1.1)–(9.1.3) if and only if there exists $\lambda^* \in \Lambda$ such that (x^*, λ^*) is a saddle point of Lagrangian function $\mathcal{L}(x, \lambda)$, i.e., the saddle point conditions

$$\mathcal{L}(x^*, \lambda) \leq \mathcal{L}(x^*, \lambda^*) \leq \mathcal{L}(x, \lambda^*) \tag{9.2.46}$$

hold for all $x \in X$ and $\lambda \in \Lambda$.

9.3 Equality-Constrained Quadratic Programming

The equality-constrained quadratic programming problem can be written as

$$\min_{x \in R^n} \quad Q(x) = g^T x + \frac{1}{2} x^T G x \tag{9.3.1}$$

$$\text{s.t.} \quad A^T x = b, \tag{9.3.2}$$

where $g \in R^n$, $b \in R^m$, $A = [a_1, \cdots, a_m] \in R^{n\times m}, G \in R^{n\times n}$ and G is symmetric. Without loss of generality, we assume that rank $(A) = m$, i.e., A has full column rank.

First, we introduce the variable elimination method. Assume that the partitions are as follows:

$$x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}, \ A = \begin{bmatrix} A_B \\ A_N \end{bmatrix}, \ g = \begin{pmatrix} g_B \\ g_N \end{pmatrix}, \ G = \begin{bmatrix} G_{BB} & G_{BN} \\ G_{NB} & G_{NN} \end{bmatrix}, \quad (9.3.3)$$

where $x_B \in R^m, x_N \in R^{n-m}$, and A_B is invertible. By these partitions, the constraint condition (9.3.2) can be written as

$$A_B^T x_B + A_N^T x_N = b. \quad (9.3.4)$$

Since A_B^{-1} exists, then

$$x_B = (A_B^{-1})^T (b - A_N^T x_N). \quad (9.3.5)$$

Substituting it into (9.3.1) gives the following form

$$\min_{x_N \in R^{n-m}} \frac{1}{2} x_N^T \hat{G}_N x_N + \hat{g}_N^T x_N + \hat{c}, \quad (9.3.6)$$

which is equivalent to (9.3.1), where

$$\begin{aligned} \hat{g}_N &= g_N - A_N A_B^{-1} g_B + [G_{NB} - A_N A_B^{-1} G_{BB}](A_B^{-1})^T b, & (9.3.7) \\ \hat{G}_N &= G_{NN} - G_{NB}(A_B^{-1})^T A_N^T \\ &\quad -A_N A_B^{-1} G_{BN} + A_N A_B^{-1} G_{BB}(A_B^{-1})^T A_N^T, & (9.3.8) \\ \hat{c} &= \frac{1}{2} b^T A_B^{-1} G_{BB} A_B^{-T} b + g_B^T A_B^{-T} b. & (9.3.9) \end{aligned}$$

If $\hat{G}_N$ is positive definite, the solution of (9.3.6) is

$$x_N^* = -\hat{G}_N^{-1} \hat{g}_N \quad (9.3.10)$$

which is unique. So the solution of problem (9.3.1)–(9.3.2) is

$$x^* = \begin{bmatrix} x_B^* \\ x_N^* \end{bmatrix} = \begin{bmatrix} (A_B^{-1})^T b \\ 0 \end{bmatrix} + \begin{bmatrix} (A_B^{-1})^T A_N^T \\ -I \end{bmatrix} \hat{G}_N^{-1} \hat{g}_N. \quad (9.3.11)$$

Let λ^* be the Lagrange multiplier vector at x^*, then

$$g + Gx^* = A\lambda^*, \quad (9.3.12)$$

and thus

$$\lambda^* = A_B^{-1}(g_B + G_{BB}x_B^* + G_{BN}x_N^*). \tag{9.3.13}$$

If $\hat{G}_N$ in (9.3.6) is positive semi-definite, then when

$$(I - \hat{G}_N\hat{G}_N^+)\hat{g}_N = 0, \tag{9.3.14}$$

i.e., $\hat{g}_N \in R(\hat{G}_N)$, the minimization problem (9.3.6) is bounded, and its solution is

$$x_N^* = -\hat{G}_N^+\hat{g}_N + (I - \hat{G}_N^+\hat{G}_N)\tilde{x}, \tag{9.3.15}$$

where $\tilde{x} \in R^{n-m}$ is any vector, $\hat{G}_N^+$ denotes the generalized inverse matrix of $\hat{G}_N$. In this case, the solution of problem (9.3.1)–(9.3.2) can be represented by (9.3.15) and (9.3.5). If (9.3.14) does not hold, the problem (9.3.6) has no lower bound, and thus the original problem (9.3.1)–(9.3.4) also has no lower bound, that is, the original problem has no finite solution.

If $\hat{G}_N$ has negative eigenvalue, it is obvious that the minimization problem (9.3.6) has not lower bound, and thus the problem (9.3.1)–(9.3.2) has not finite solution.

Example 9.3.1

$$\begin{aligned} \min \quad & Q(x) = x_1^2 - x_2^2 - x_3^2 && (9.3.16)\\ \text{s.t.} \quad & x_1 + x_2 + x_3 = 1, && (9.3.17)\\ & x_2 - x_3 = 1. && (9.3.18) \end{aligned}$$

From (9.3.18), we have

$$x_2 = x_3 + 1. \tag{9.3.19}$$

Substituting it into (9.3.17) yields

$$x_1 = -2x_3. \tag{9.3.20}$$

In fact, here $x_B = (x_1, x_2)^T, x_N = x_3$. By substituting (9.3.19)–(9.3.20) into (9.3.16), we obtain

$$\min_{x_3 \in R} 4x_3^2 - (x_3 + 1)^2 - x_3^2. \tag{9.3.21}$$

Solving (9.3.21) gives $x_3 = \frac{1}{2}$. By substituting $x_3 = \frac{1}{2}$ into (9.3.19)–(9.3.20), we get

$$x^* = (-1, \frac{3}{2}, \frac{1}{2})^T,$$

which is the solution of (9.3.16)–(9.3.18).

By use of $g^* = A\lambda^*$, it follows that

$$\begin{pmatrix} -2 \\ -3 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \lambda_1^* \\ \lambda_2^* \end{pmatrix} \tag{9.3.22}$$

which gives Lagrange multipliers $\lambda_1^* = -2$ and $\lambda_2^* = -1$. □

The idea of variable elimination method is simple and clear. However, when A_B closes to singular, computing the solution by (9.3.11) will lead to a numerically instable case.

A direct generalization of the variable elimination method is the generalized elimination method. We partition R^n into two complementary subspaces, i.e., $R^n = R(A) \oplus N(A^T)$. Let $y_1, \cdots, y_m$ be a set of linearly independent vectors in $R(A)$, the range of A, and let $z_1, \cdots, z_{n-m}$ be a set of linearly independent vectors in $N(A^T)$, the null space of A^T. Write

$$Y = [y_1, \cdots, y_m],\ Z = [z_1, \cdots, z_{n-m}],$$

which are $n \times m$ and $n \times (n-m)$ matrices respectively. Obviously, $R(Y) = R(A)$, $R(Z) = N(A^T)$, and $[Y : Z]$ is nonsingular. In addition, $A^T Y$ is nonsingular and $A^T Z = 0$. Set

$$x = Y\bar{x} + Z\hat{x}, \tag{9.3.23}$$

where $\bar{x} \in R^m, \hat{x} \in R^{n-m}$, it follows from the constraint condition (9.3.2) that

$$b = A^T x = A^T Y \bar{x}. \tag{9.3.24}$$

Then the feasible point of (9.3.1)–(9.3.2) can be represented as

$$x = Y(A^T Y)^{-1} b + Z\hat{x}. \tag{9.3.25}$$

By substituting (9.3.25) into (9.3.1), we obtain

$$\min_{\hat{x} \in R^{n-m}} (g + GY(A^T Y)^{-1} b)^T Z\hat{x} + \frac{1}{2}\hat{x}^T Z^T G Z \hat{x}, \tag{9.3.26}$$

which is an unconstrained minimization problem in R^{n-m}. Here $Z^T GZ$ and $Z^T(g + GY(A^T Y)^{-1} b)$ are called reduced Hessian and reduced gradient, respectively. Suppose that $Z^T GZ$ is positive definite, then it follows from (9.3.26) that

$$(Z^T GZ)\hat{x} = -[Z^T GY(A^T Y)^{-1} b + Z^T g] \tag{9.3.27}$$

or

$$\hat{x}^* = -(Z^T G Z)^{-1} Z^T (g + GY(A^T Y)^{-1} b). \tag{9.3.28}$$

The system (9.3.27) can be solved by means of Cholesky factorization. Thus, the (9.3.28) and (9.3.25) give the solution of problem (9.3.1)–(9.3.2)

$$\begin{aligned} x^* &= Y(A^T Y)^{-1} b - Z(Z^T G Z)^{-1} Z^T (g + GY(A^T Y)^{-1} b) \\ &= (I - Z(Z^T G Z)^{-1} Z^T G) Y (A^T Y)^{-1} b - Z(Z^T G Z)^{-1} Z^T g. \end{aligned} \tag{9.3.29}$$

Furthermore, from the KKT condition

$$A\lambda^* = g + Gx^*,$$

by left-multiplying Y^T and noting that $A^T Y$ is nonsingular, we obtain

$$(Y^T A)\lambda^* = Y^T (g + Gx^*)$$

and

$$\begin{aligned} \lambda^* &= (A^T Y)^{-T} Y^T [g + Gx^*] \\ &= (A^T Y)^{-T} Y^T [Pg + GP^T Y (A^T Y)^{-1} b], \end{aligned} \tag{9.3.30}$$

where

$$P = I - GZ(Z^T G Z)^{-1} Z^T \tag{9.3.31}$$

is an affine mapping from R^n to $R(A)$. In particular, if we choose Y such that

$$A^T Y = I, \tag{9.3.32}$$

where Y is a left-inverse of A^T, then (9.3.25) becomes

$$x = Yb + Z\hat{x}, \tag{9.3.33}$$

where $\hat{x} \in R^{n-m}$, and further (9.3.29)–(9.3.30) become

$$x^* = Yb - Z(Z^T G Z)^{-1} Z^T (g + GYb) \tag{9.3.34}$$

$$= P^T Yb - Z(Z^T G Z)^{-1} Z^T g, \tag{9.3.35}$$

and

$$\begin{aligned} \lambda^* &= Y^T (g + Gx^*) \\ &= Y^T (Pg + GP^T Yb). \end{aligned} \tag{9.3.36}$$

From (9.3.25), we know that the feasible area of (9.3.1)–(9.3.2) is a subspace parallel to $N(A^T)$. The generalized elimination method just uses column-vectors z_i $(i = 1, \cdots, n-m)$ of Z, which form a base of the null space of A^T, as basis vectors and transforms the quadratic programming problem (9.3.1)–(9.3.2) into an unconstrained minimization (9.3.26) of quadratic function in a reduced space. Thus, this kind of method is also said to be null-space method.

The above discussions tell us that how to choose matrix Z, base matrix of the null space $N(A^T)$, is a key for this kind of methods. Different choices of Z form different null-space methods for solving quadratic programming problem (9.3.1)–(9.3.2). In the following we give some typical choices.

Clearly, the variable elimination method is a particular case of the generalized elimination method in which

$$Y = \begin{bmatrix} A_B^{-1} \\ 0 \end{bmatrix}, \tag{9.3.37}$$

$$Z = \begin{bmatrix} -A_B^{-T} A_N^T \\ I \end{bmatrix}. \tag{9.3.38}$$

Another particular case is based on QR decomposition of A. Let

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1 \; Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R, \tag{9.3.39}$$

where Q is an $n \times n$ orthogonal matrix, R is an $m \times m$ nonsingular upper triangular matrix. Therefore, we have a choice

$$Y = (A^+)^T = Q_1 R^{-T}, \; Z = Q_2. \tag{9.3.40}$$

A general scheme for choosing Y and Z is as follows. For any Y and Z with $A^T Y = I$ and $A^T Z = 0$,

$$A^T [Y \; Z] = [I \; 0]. \tag{9.3.41}$$

Since $[Y \; Z]$ is nonsingular, there exists $V \in R^{n \times (n-m)}$ such that

$$[Y \; Z] = \begin{bmatrix} A^T \\ V^T \end{bmatrix}^{-1}, \tag{9.3.42}$$

i.e.,

$$[A\ V]^{-1} = \begin{bmatrix} Y^T \\ Z^T \end{bmatrix}. \tag{9.3.43}$$

It means that the different choices of $V \in R^{n\times(n-m)}$ lead to different Y and Z, and different elimination methods. For example, if we set

$$V = \begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix},$$

we can get the variable elimination method (9.3.11). If we set

$$V = Q_2,$$

the above orthogonal decomposition choice (9.3.40) is obtained. Normally, null-space method is very useful, especially for small and medium-sized problems and when the computation of the null-space matrix Z and the factors of Z^TGZ is not very expensive.

The Lagrange method for solving equality-constrained quadratic programming is based on KKT conditions, which are

$$g + Gx = A\lambda, \tag{9.3.44}$$
$$A^Tx = b. \tag{9.3.45}$$

The above system can be written in the matrix form

$$\begin{bmatrix} G & -A \\ -A^T & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = - \begin{bmatrix} g \\ b \end{bmatrix}. \tag{9.3.46}$$

Here

$$\begin{bmatrix} G & -A \\ -A^T & 0 \end{bmatrix} \tag{9.3.47}$$

is a KKT matrix for quadratic programming (9.3.1)–(9.3.2). It is not difficult to show that if A has full column-rank and Z^TGZ is positive definite, then KKT matrix (9.3.47) is nonsingular.

Theorem 9.3.2 *Let $A \in R^{n\times m}$ be a full column-rank matrix. Assume that the reduced Hessian Z^TGZ is positive definite. Then the KKT matrix (9.3.47) is nonsingular. Furthermore, there exists a unique KKT pair (x^*, λ^*) such that equation (9.3.46) is satisfied.*

Proof. The proof is by contradiction. Suppose that KKT matrix (9.3.47) is singular, then there exists nonzero vector $(p, v) \neq 0$ such that

$$\begin{bmatrix} G & -A \\ -A^T & 0 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} = 0, \tag{9.3.48}$$

where $p \in R^n$ and $v \in R^m$. Clearly, we have $A^T p = 0$. By left-multiplying $\begin{bmatrix} p \\ v \end{bmatrix}^T$ on both sides of (9.3.48), we obtain

$$0 = \begin{bmatrix} p \\ v \end{bmatrix}^T \begin{bmatrix} G & -A \\ -A^T & 0 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} = p^T G p.$$

Since $p \in N(A^T)$ and $Z = [z_1, \cdots, z_{n-m}]$ spans $N(A^T)$, we may denote $p = Zu$ for some $u \in R^{n-m}$ and have

$$0 = p^T G p = u^T Z G Z u.$$

The assumption that $Z^T G Z$ is positive definite gives $u = 0$ and then

$$p = Zu = 0. \tag{9.3.49}$$

So, it follows from (9.3.48) that $Av = 0$. Notice that A has full column-rank, then we obtain also $v = 0$ which together with (9.3.49) contradicts the fact $(p, v) \neq 0$. We complete the proof. □

Now let KKT matrix be nonsingular. Then there exist matrices $U \in R^{n\times n}, W \in R^{n\times m}$ and $T \in R^{m\times m}$ such that

$$\begin{bmatrix} G & -A \\ -A^T & 0 \end{bmatrix}^{-1} = \begin{bmatrix} U & W \\ W^T & T \end{bmatrix}, \tag{9.3.50}$$

and the unique solution of (9.3.46) is

$$x^* = -Ug - Wb, \tag{9.3.51}$$

$$\lambda^* = -W^T g - Tb. \tag{9.3.52}$$

As long as the KKT matrix (9.3.47) is nonsingular, then (9.3.50) is determined uniquely, so the stationary point of the Lagrangian function is determined uniquely by (9.3.51)–(9.3.52). However, since there are many expressions for U, W, and T, and we can derive a different computational schemes of formula (9.3.51)–(9.3.52).

If G is invertible and A has full column-rank, then $(A^TG^{-1}A)^{-1}$ exists. It is not difficult to show that the expressions of U, W, and T in (9.3.50) are

$$U = G^{-1} - G^{-1}A(A^TG^{-1}A)^{-1}A^TG^{-1}, \tag{9.3.53}$$
$$W = -G^{-1}A(A^TG^{-1}A)^{-1}, \tag{9.3.54}$$
$$T = -(A^TG^{-1}A)^{-1}. \tag{9.3.55}$$

Then it follows from (9.3.46) that the solution for quadratic programming with equality constraints is

$$x^* = -G^{-1}g + G^{-1}A(A^TG^{-1}A)^{-1}[A^TG^{-1}g + b], \tag{9.3.56}$$
$$\lambda^* = (A^TG^{-1}A)^{-1}[A^TG^{-1}g + b]. \tag{9.3.57}$$

As we said, if A has full column-rank and Z^TGZ is positive definite, then KKT matrix is invertible. In this case, if Y and Z are defined by (9.3.42), the matrices U, W, and T in (9.3.50) can be represented as

$$U = Z(Z^TGZ)^{-1}z^T, \tag{9.3.58}$$
$$W = -P^TY, \tag{9.3.59}$$
$$T = -Y^TGP^TY, \tag{9.3.60}$$

where P is defined by (9.3.31). Substituting (9.3.58)–(9.3.60) into (9.3.51)–(9.3.52) yields the formula (9.3.35)–(9.3.36). Hence, the Lagrange method is equivalent to the generalized elimination method.

9.4 Active Set Methods

Most QP problems involve inequality constraints and so can be expressed in the form (9.1.1)–(9.1.3). In this section we describe how the methods for solving equality-constrained QP can be generalized to handle the general QP problem (9.1.1)–(9.1.3) by means of active set methods, which are, in general, the most effective methods for small and medium-sized problems. We start our discussion by considering the convex case, i.e., the matrix G in (9.1.1)–(9.1.3) is positive semi-definite. The other case in which G is indefinite will be simply discussed in the end of the section. Intuitively, inactive inequality constraints do not play any role near the solution, so they can be dropped; the active inequality constraints have zero values at solution, and so they can be replaced by equality constraints. The following lemma is a base for active set methods.

Lemma 9.4.1 *Let x^* be a local minimizer of QP problem (9.1.1)–(9.1.3). Then x^* is a local minimizer of problem*

$$\min_{x\in R^n} \quad g^T x + \frac{1}{2}x^T G x \tag{9.4.1}$$

$$\text{s.t.} \qquad a_i^T x = b_i,\ i \in E \cup I(x^*). \tag{9.4.2}$$

Conversely, if x^* is a feasible point of (9.1.1)–(9.1.3) and a KKT point of (9.4.1)–(9.4.2), and the corresponding Lagrange multiplier vector λ^* satisfies

$$\lambda_i^* \geq 0,\ i \in I(x^*), \tag{9.4.3}$$

then x^* is also the KKT point of problem (9.1.1)–(9.1.3).

Proof. Since, near x^*, the feasible point of (9.1.1)–(9.1.3) is also feasible for problem (9.4.1)–(9.4.2), then, obviously, the local minimizer of (9.1.1)–(9.1.3) is also the local minimizer of problem (9.4.1)–(9.4.2).

Now let x^* be feasible for (9.1.1)–(9.1.3) and a KKT point for (9.4.1)–(9.4.2). Let there exist λ_i^* $(i \in E \cup I(x^*))$ such that

$$Gx^* + g = \sum_{i\in I(x^*)\cup E} a_i \lambda_i^*, \tag{9.4.4}$$

$$\lambda_i^*(a_i^T x^* - b_i) = 0,\ \lambda_i^* \geq 0,\ i \in I(x^*). \tag{9.4.5}$$

Define

$$\lambda_i^* = 0,\ i \in I \setminus I(x^*). \tag{9.4.6}$$

Then we immediately have from (9.4.4)–(9.4.6) that

$$Gx^* + g = \sum_{i=1}^{m} \lambda_i^* a_i, \tag{9.4.7}$$

$$a_i^T x^* = b_i,\ i \in E, \tag{9.4.8}$$

$$a_i^T x^* \geq b_i,\ i \in I, \tag{9.4.9}$$

$$\lambda_i^* \geq 0,\ i \in I, \tag{9.4.10}$$

$$\lambda_i^*(a_i^T x^* - b_i) = 0,\ \forall i \tag{9.4.11}$$

which means that x^* is a KKT point of problem (9.1.1)–(9.1.3). □

The active set methods are a feasible point method, that is, all iterates remain feasible. In each iteration, we solve a quadratic programming subproblem with a subset of equality constraints. This subset is said to be a working set and is denoted by $\mathcal{S}_k \subset E \cup I(x^*)$.

If the solution of the equality-constrained QP subproblem on $\mathcal{S}_k$ is feasible for original problem (9.1.1)–(9.1.3), we need to examine whether (9.4.3) is satisfied or not. If (9.4.3) is satisfied, then stop and we get the solution of the original problem. Otherwise, the KKT conditions are not satisfied, and the objective function $q(\cdot)$ can be decreased by dropping this constraint. Thus, we remove the index from the working set $\mathcal{S}_k$ and solve a new subproblem. If the solution of equality-constrained QP subproblem on $\mathcal{S}_k$ is not feasible for problem (9.1.1)–(9.1.3), we need to add a constraint into the working set $\mathcal{S}_k$ and then solve a new subproblem.

At each iteration, a feasible point x_k and a working set $\mathcal{S}_k$ are known. Each iteration attempts to locate a solution of an equality-constrained subproblem on $\mathcal{S}_k$. Let d be a step from x_k. We can express the QP subproblem in terms of d. Consider the QP subproblem

$$\min_{d\in R^n} \quad \frac{1}{2}(x_k+d)^T G(x_k+d) + g^T(x_k+d), \tag{9.4.12}$$

$$\text{s.t.} \quad a_i^T d = 0,\ i \in \mathcal{S}_k \tag{9.4.13}$$

which can be written as

$$\min_{d\in R^n} \quad \frac{1}{2}d^T G d + g_k^T d \tag{9.4.14}$$

$$\text{s.t.} \quad a_i^T d = 0,\ i \in \mathcal{S}_k \tag{9.4.15}$$

where $g_k = \nabla Q(x_k) = Gx_k + g$. Denote the KKT point of (9.4.12)–(9.4.13) by d_k, the corresponding Lagrange multipliers by $\lambda_i^{(k)}\,(i \in \mathcal{S}_k)$. If $d_k = 0$, then x_k is the KKT point of subproblem

$$\min_{x\in R^n} \quad \frac{1}{2}x^T G x + g^T x \tag{9.4.16}$$

$$\text{s.t.} \quad a_i^T x = b_i,\ i \in \mathcal{S}_k. \tag{9.4.17}$$

At this time, if $\lambda_i^{(k)} \geq 0$, $\forall i \in \mathcal{S}_k \cap I$, then x_k is a KKT point of problem (9.1.1)–(9.1.3), and we terminate the iteration. Otherwise, there exists negative Lagrange multiplier, for example, $\lambda_{i_k}^{(k)} < 0$. In this case, it is possible to reduce the objective function by dropping the i_k-th constraint from current working set $\mathcal{S}_k$. Then we solve the resulting QP subproblem. Note that if there are more than one index such that $\lambda_i < 0$, it is usual to choose i_k for which

$$\lambda_{i_k} = \min_{\substack{i\in\mathcal{S}_k\cap I \\ \lambda_i^{(k)}<0}} \lambda_i^{(k)} \tag{9.4.18}$$

and set

$$\mathcal{S}_k := \mathcal{S}_k \setminus \{i_k\}. \tag{9.4.19}$$

Suppose that the solution $d_k \neq 0$. If $x_k + d_k$ is feasible with regard to all the constraints, then we set

$$x_{k+1} = x_k + d_k. \tag{9.4.20}$$

Otherwise, a line search is made along the direction d_k and we set

$$x_{k+1} = x_k + \alpha_k d_k, \tag{9.4.21}$$

where α_k is a steplength such that $x_k + \alpha_k d_k$ is the "best" feasible point on $[x_k, x_k + d_k]$ and the closest to $x_k + d_k$, i.e., take α_k as large as possible in the interval $[0, 1]$.

Now we derive the explicit formula for α_k. We ask $x_k + \alpha_k d_k$ for satisfying all constraints. Obviously, if $i \in \mathcal{S}_k$, then the corresponding constraint will be certainly feasible. Thus we only need to consider those constraints for which $i \notin \mathcal{S}_k$. There are two cases we need to consider. If $a_i^T d_k \geq 0$ for some $i \notin \mathcal{S}_k$, then we have for all $\alpha_k \geq 0$,

$$a_i^T(x_k + \alpha_k d_k) \geq a_i^T x_k \geq b_i,\ i \notin \mathcal{S}_k.$$

In this case, the constraint is satisfied. If $a_i^T d_k < 0$ for some $i \notin \mathcal{S}_k$, we have

$$a_i^T(x_k + \alpha_k d_k) \geq b_i$$

only if

$$\alpha_k \leq \frac{b_i - a_i^T x_k}{a_i^T d_k},\ i \notin \mathcal{S}_k. \tag{9.4.22}$$

Hence, we should take

$$\alpha_k = \min_{\substack{i \notin \mathcal{S}_k \\ a_i^T d_k < 0}} \frac{b_i - a_i^T x_k}{a_i^T d_k}. \tag{9.4.23}$$

Since we want α_k to be as large as possible in $[0, 1]$ subject to remaining feasibility, we have the following formula:

$$\alpha_k = \min \left\{ 1,\ \min_{\substack{i \notin \mathcal{S}_k \\ a_i^T d_k < 0}} \frac{b_i - a_i^T x_k}{a_i^T d_k} \right\}. \tag{9.4.24}$$

If $\alpha_k < 1$, i.e., (9.4.23) holds, then there exists some $j \notin \mathcal{S}_k$ such that

$$\alpha_k = \frac{b_j - a_j^T x_k}{a_j^T d_k}.$$

Thus,

$$a_j^T x_{k+1} = a_j^T x_k + \alpha_k a_j^T d_k = b_j.$$

This means that there is a new constraint indexed by $j \notin \mathcal{S}_k$ becoming an active constraint at x_{k+1}. So we put it into the working set, that is, set $\mathcal{S}_{k+1} = \mathcal{S}_k \cup \{j\}$.

If $\alpha_k = 1$, then the working set remains the same, i.e., $\mathcal{S}_{k+1} = \mathcal{S}_k$.

So, we can continue the next iteration on the new working set $\mathcal{S}_{k+1}$.

Now, we are in a position to give the algorithm of active set method as follows.

Algorithm 9.4.2 *(Active Set Methods)*

Step 1. Given x_1, set $\mathcal{S}_1 = E \cup I(x_1), k := 1$.

Step 2. Find the solution d_k for subproblem (9.4.12)–(9.4.13).
If $d_k \neq 0$, go to Step 3;
Else if $d_k = 0$, compute $\lambda_i^{(k)}$ from $Gx_k + g = \sum_{i \in \mathcal{S}_k} \lambda_i^{(k)} a_i$.
If $\lambda_i^{(k)} \geq 0 \, \forall i \in \mathcal{S}_k \cap I$, stop;
else find i_k by (9.4.18).
$\mathcal{S}_k := \mathcal{S}_k \setminus \{i_k\}, x_{k+1} = x_k$, go to Step 4.

Step 3. Find α_k by (9.4.24);
Set

$$x_{k+1} = x_k + \alpha_k d_k. \tag{9.4.25}$$

If $\alpha_k = 1$, go to Step 4;
Else find $j \notin \mathcal{S}_k$ such that

$$a_j^T (x_k + \alpha_k d_k) = b_j. \tag{9.4.26}$$

Set $\mathcal{S}_k := \mathcal{S}_k \cup \{j\}$.

Step 4. $\mathcal{S}_{k+1} := \mathcal{S}_k, k := k + 1$, go to Step 2. □

Now we give analysis to the algorithm.

From Algorithm 9.4.2, we know that all iterates are feasible, i.e.,

$$x_k \in X, \ \forall k, \tag{9.4.27}$$

and the objective function remains descent, i.e.,

$$Q(x_{k+1}) \leq Q(x_k), \ \forall k. \tag{9.4.28}$$

Further, as long as $d_k \neq 0$ (i.e., x_k is not the KKT point of (9.4.16)–(9.4.17)) and $\alpha_k > 0$, we have

$$Q(x_{k+1}) < Q(x_k). \tag{9.4.29}$$

If the algorithm terminates in finitely many steps, the obtained point is a KKT point of the original problem (9.1.1)–(9.1.3).

Suppose that the algorithm does not terminate in finitely many steps; since there is only a finite number of constraints, it is impossible that the number of elements in $\mathcal{S}_k$ increases infinitely many times and does not reduce. So there are infinitely many indices k such that $d_k = 0$. It follows from the algorithm that there are infinitely many indices k such that x_k is a KKT point of (9.4.16)–(9.4.17). Since the number of constraints is finite, $\mathcal{S}_k$ has only finitely many different combinations and so the sequence of the objective values $\{Q(x_k)\}$ has only finitely many elements. Therefore, there must exist a sufficiently large k_0 such that

$$Q(x_{k+1}) = Q(x_k), \ \forall k \geq k_0. \tag{9.4.30}$$

Then for all $k \geq k_0$, in both

$$\alpha_k = 0 \tag{9.4.31}$$

and

$$d_k = 0, \tag{9.4.32}$$

only one holds. Since there are only finitely many constraints, it is impossible that the algorithm only increases the constraint into $\mathcal{S}_k$, nor reduces the constraint from $\mathcal{S}_k$. Hence, there must be infinitely many indices k such that

$$d_k \neq 0, \tag{9.4.33}$$

and infinitely many indices k such that

$$d_k = 0. \tag{9.4.34}$$

So, there exist $k_2 > k_1 > k_0$ such that

$$d_{k_1} = 0,\ d_{k_2} = 0, \tag{9.4.35}$$

$$d_k \neq 0,\ k_1 < k < k_2, \tag{9.4.36}$$

and

$$k_2 > k_1 + 1. \tag{9.4.37}$$

Lemma 9.4.3 *Let k_0 be an index satisfying (9.4.30). If $k_2 > k_1 > k_0$ satisfy (9.4.35)–(9.4.37), then*

$$\mathcal{S}_{k_2} \neq \mathcal{S}_{k_1}. \tag{9.4.38}$$

Proof. By (9.4.35), there exist $\lambda_i^{(k_1)}$ such that

$$g + G\bar{x} = \sum_{i \in \mathcal{S}_{k_1}} a_i \lambda_i^{(k_1)}, \tag{9.4.39}$$

where $\bar{x} = x_{k_0}$. From (9.4.31)–(9.4.32), it follows that $x_k = \bar{x}$ for all $k \geq k_0$.

Since $d_{k_1+1} \neq 0, \alpha_{k_1+1} = 0$, there must be

$$j \notin \mathcal{S}_{k_1+1}, \tag{9.4.40}$$

such that $j \in \mathcal{S}_{k_1+2}$,

$$j \in I(\bar{x}) \tag{9.4.41}$$

and

$$a_j^T d_{k_1+1} < 0. \tag{9.4.42}$$

Since d_k is a solution for subproblem (9.4.12)–(9.4.13), i.e., d_k is a descent direction of the objective function, then

$$(g + G\bar{x})^T d_{k_1+1} \leq 0. \tag{9.4.43}$$

By using (9.4.39), (9.4.43) and $\mathcal{S}_{k_1+1} = \mathcal{S}_{k_1} \setminus \{i_{k_1}\}$, we get

$$\lambda_{i_{k_1}}^{(k_1)} a_{i_{k_1}}^T d_{k_1+1} \leq 0, \tag{9.4.44}$$

which means that

$$a_{i_{k_1}}^T d_{k_1+1} \geq 0 \tag{9.4.45}$$

by the definition of $\{i_k\}$. Comparing (9.4.42)–(9.4.44) gives $j \neq i_{k_1}$. Hence it follows from (9.4.40) that $j \notin \mathcal{S}_{k_1}$.

On the other hand, $j \in \mathcal{S}_{k_1+2} \subseteq \mathcal{S}_{k_2}$. Hence we have $\mathcal{S}_{k_2} \neq \mathcal{S}_{k_1}$. The proof is complete. □

Finally, we give the convergence theorem of active set methods.

Theorem 9.4.4 *If, for all k, $a_i\,(i \in E \cup I(x_k))$ are linearly independent, then either the sequence generated from Algorithm 9.4.2 converges to a KKT point of problem (9.1.1)–(9.1.3) in finite iterations, or the original problem (9.1.1)–(9.1.3) is unbounded below.*

Proof. Assume that the problem (9.1.1)–(9.1.3) is bounded below, then the sequence $\{x_k\}$ is bounded.

If the solution of subproblem (9.4.12)–(9.4.13) is $d_k = 0$, then x_k is a KKT point of (9.4.16)–(9.4.17) for the current working set $\mathcal{S}_k$. If $\lambda_i^{(k)} \geq 0$, $\forall i \in \mathcal{S}_k \cap I$, then x_k is a KKT point of the original problem (9.1.1)–(9.1.3). Otherwise, there exists $\lambda_{i_k}^{(k)} < 0$ $(i_k \in \mathcal{S}_k \cap I)$ for which we can find a feasible descent direction d_k such that

$$a_j^T d_k = 0, \quad j \in \mathcal{S}_k, j \neq i_k, \tag{9.4.46}$$

$$a_{i_k}^T d_k > 0 \tag{9.4.47}$$

and

$$g_k^T d_k = (\lambda^{(k)})^T A_k^T d_k = (a_{i_k}^T d_k)(\lambda^{(k)})^T e_{i_k} = (a_{i_k}^T d_k)\lambda_{i_k}^{(k)} < 0. \tag{9.4.48}$$

If we substitute (9.4.46) for the constraints in (9.4.13), i.e., set $\mathcal{S}_k := \mathcal{S}_k \backslash \{i_k\}$, the resulting QP subproblem will have a feasible descent direction. Since $\alpha_k > 0$, we have

$$Q(x_{k+1}) < Q(x_k),$$

and consequently, by finiteness of constraints, the algorithm never returns to the current working set $\mathcal{S}_k$, and the sequence $\{x_k\}$ is finite.

If $d_k \neq 0$ and $\alpha_k = 1$, then $\mathcal{S}_{k+1} = \mathcal{S}_k$, and the subproblem (9.4.12)–(9.4.13) is unchanged for x_{k+1} and so the x_{k+1} is the solution of (9.4.12)–(9.4.13).

Only if $d_k \neq 0$ and $\alpha_k < 1$, x_{k+1} is not the solution of (9.4.12)–(9.4.13). At this time, from (9.4.26) in Step 3 of Algorithm 9.4.2, we know that there is an index $j \notin \mathcal{S}_k$ such that the j-th constraint is feasible. So, such a constraint is added into $\mathcal{S}_{k+1}$. If this procedure occurs repeatedly, then after at most n iterations the working set $\mathcal{S}_k$ will contains n indices, which correspond to n linearly independent vectors, then it follows from (9.4.13) that $d_k = 0$. Thus such a procedure continues at most n times. So there will be a KKT point x_k of (9.4.16)–(9.4.17) at most after n iterations.

Combining the above discussion, in any case, the algorithm will converge in finite iterations to the KKT point of problem (9.1.1)–(9.1.3). □

By modifying the algorithm, the active set method for a convex QP problem can be adopted to the indefinite case in which the Hessian matrix G has some negative eigenvalues.

As we know from §9.3 that if G in $\mathcal{S}_k$ is indefinite, then the problem (9.4.13) may be unbounded. We can choose the direction d_k such that $a_i^T d_k = 0\,(\forall i \in \mathcal{S}_k)$ and either

$$d_k^T G d_k < 0 \tag{9.4.49}$$

or

$$\nabla Q(x_k)^T d_k < 0,\ d_k^T G d_k = 0 \tag{9.4.50}$$

where $\nabla Q(x_k) = g + Gx_k$. If, for all $i \notin \mathcal{S}_k$, $a_i^T d_k \geq 0$, then the original problem (9.1.1)–(9.1.3) is unbounded below. Otherwise, we can find $i \notin \mathcal{S}_k$ and $a_i^T d_k < 0$. Then, when $\alpha > 0$ is sufficiently large, $x_k + \alpha d_k$ is not a feasible point of (9.1.1)–(9.1.3). In this case we can take α_k as large as possible and make $x_k + \alpha_k d_k$ feasible.

9.5 Dual Method

For the convex QP problem

$$\min_{x \in R^n} \quad g^T x + \frac{1}{2} x^T G x \tag{9.5.1}$$

$$\text{s.t.} \quad a_i^T x = b_i,\ i \in E, \tag{9.5.2}$$

$$a_i^T x \geq b_i,\ i \in I, \tag{9.5.3}$$

where G is symmetric and positive definite. We know from §9.2 that the dual problem is

$$\min_{\lambda \in R^m} \quad -(b + AG^{-1}g)^T \lambda + \frac{1}{2} \lambda^T (A^T G^{-1} A) \lambda \tag{9.5.4}$$

$$\text{s.t.} \quad \lambda_i \geq 0,\ i \in I. \tag{9.5.5}$$

Now we adopt the active-set method to (9.5.4)–(9.5.5). The equality-constrained subproblem we solved at each iteration is

$$\min_{\lambda \in R^m} \quad -(b + A^T G^{-1} g)^T \lambda + \frac{1}{2} \lambda^T (A^T G^{-1} A) \lambda \tag{9.5.6}$$

$$\text{s.t.} \quad \lambda_i = 0,\ i \in \bar{\mathcal{S}}_k, \tag{9.5.7}$$

where $\bar{\mathcal{S}}_k \subseteq I$ is a working set for dual problem (9.5.4)–(9.5.5). Let λ_k be a KKT point of the subproblem (9.5.6)–(9.5.7). Set

$$x_k = -G^{-1}(g - A\lambda_k), \tag{9.5.8}$$

then

$$Gx_k + g = A\lambda_k, \tag{9.5.9}$$

and from

$$(b + A^T G^{-1} g - A^T G^{-1} A\lambda_k)_i = 0, \ \forall i \notin \bar{\mathcal{S}}_k, \tag{9.5.10}$$

we have

$$(A^T x_k - b)_i = 0, \ \forall i \notin \bar{\mathcal{S}}_k. \tag{9.5.11}$$

Thus, x_k is the KKT point of the subproblem

$$\min_{x \in R^n} \quad g^T x + \frac{1}{2} x^T G x \tag{9.5.12}$$

$$\text{s.t.} \qquad a_i^T x = b_i, \ i \notin \bar{\mathcal{S}}_k. \tag{9.5.13}$$

Write $\mathcal{S}_k = \{I \cup E\} \setminus \bar{\mathcal{S}}_k$. It is obvious that (9.5.12)–(9.5.13) is the same as (9.4.16)–(9.4.17). It is not difficult to see that the Lagrange multipliers of dual problem (9.5.6)-(9.5.7) satisfy

$$\begin{aligned} & (A^T G^{-1} A\lambda_k - b - A^T G^{-1} g)_i \\ = \ & (A^T x_k - b)_i = a_i^T x_k - b_i, \ i \in \bar{\mathcal{S}}_k. \end{aligned} \tag{9.5.14}$$

We ask λ_k to be a feasible point of (9.5.4)–(9.5.5). If the Lagrange multiplier (9.5.14) of the dual problem (9.5.6)–(9.5.7) is nonnegative, x_k is a KKT point of the original problem (9.5.1)–(9.5.3). Let A_k be a matrix with the columns $a_i \, (i \in \mathcal{S}_k)$, $\bar{\lambda}_k$ the vector consisting of the components of λ_k corresponding to $i \in \mathcal{S}_k$. It follows from (9.5.10) that

$$b_i + a_i^T G^{-1} g - a_i^T G^{-1} A_k \bar{\lambda}_k = 0, \ i \in \mathcal{S}_k, \tag{9.5.15}$$

i.e.,

$$b^{(k)} + A_k^T G^{-1} g - A_k^T G^{-1} A_k \bar{\lambda}_k = 0, \tag{9.5.16}$$

where $b^{(k)}$ consists of the components of b corresponding to $i \in \mathcal{S}_k$. Then (9.5.16) gives

$$\bar{\lambda}_k = (A_k^T G^{-1} A_k)^{-1} [b^{(k)} + A_k^T G^{-1} g]. \tag{9.5.17}$$

When Lagrange multipliers in (9.5.10) are not all nonnegative, we should, by the active-set method, drop an index $i_k \in \bar{\mathcal{S}}_k$, that is, add the index i_k into $\mathcal{S}_k$. For convenience of sign, we write i_k as p. Then we have $\mathcal{S}_{k+1} = \mathcal{S}_k \cup \{p\}$.

Let

$$\bar{\lambda}_{k+1} = \begin{pmatrix} \bar{\lambda}_k \\ 0 \end{pmatrix} + \begin{pmatrix} \delta\lambda_k \\ \beta_k \end{pmatrix}. \tag{9.5.18}$$

It follows from (9.5.17) that

$$\begin{pmatrix} A_k^T G^{-1} A_k & A_k^T G^{-1} a_p \\ a_p^T G^{-1} A_k & a_p^T G^{-1} a_p \end{pmatrix} \begin{pmatrix} \delta\lambda_k \\ \beta_k \end{pmatrix} = \begin{pmatrix} 0 \\ b_p - a_p^T x_k \end{pmatrix}, \tag{9.5.19}$$

which gives

$$\bar{\lambda}_{k+1} = \begin{pmatrix} \bar{\lambda}_k \\ 0 \end{pmatrix} + \beta_k \begin{pmatrix} -(A_k^T G^{-1} A_k)^{-1} A_k^T G^{-1} a_p \\ 1 \end{pmatrix}. \tag{9.5.20}$$

So,

$$\begin{aligned} x_{k+1} &= x_k + G^{-1} A_{k+1} \left(\bar{\lambda}_{k+1} - \begin{bmatrix} \bar{\lambda}_k \\ 0 \end{bmatrix} \right) \\ &= x_k + \beta_k G^{-1} (I - A_k (A_k^T G^{-1} A_k)^{-1} A_k^T G^{-1}) a_p. \end{aligned} \tag{9.5.21}$$

Let

$$A_k^* = (A_k^T G^{-1} A_k)^{-1} A_k^T G^{-1}, \tag{9.5.22}$$

$$y_k = A_k^* a_p. \tag{9.5.23}$$

Since $\bar{\lambda}_{k+1}$ should satisfy $\bar{\lambda}_{k+1} \geq 0$, it follows from (9.5.20) and (9.5.23) that

$$0 \leq \beta_k \leq \min_{\substack{j \in \mathcal{S}_k \\ (y_k)_j > 0}} \frac{(\bar{\lambda}_k)_j}{(y_k)_j}. \tag{9.5.24}$$

If

$$G^{-1}(I - A_k A_k^*) a_p = 0 \tag{9.5.25}$$

and $y_k \leq 0$, then

$$(-y_k, 1)^T (A_{k+1}^T G^{-1} A_{k+1}) \begin{pmatrix} -y_k \\ 1 \end{pmatrix} = 0 \tag{9.5.26}$$

and

$$(-y_k, 1)^T(b^{(k+1)} + A_{k+1}^T G^{-1} g) = b_p - a_p^T x_k > 0, \tag{9.5.27}$$

which indicate that the dual problem (9.5.4)–(9.5.5) is unbounded. Further, we know by the duality theory that the original problem (9.5.1)–(9.5.3) has no feasible point.

Now, by use of the analysis above, we describe the dual method due to Goldfarb and Idnani [155] as follows (we consider the case in which $m_e = 0$, i.e., the problem with only inequality constraints).

Algorithm 9.5.1 *(Dual Method)*

Step 1. $x_1 = -G^{-1}g, f_1 = \frac{1}{2}g^T x_1, \mathcal{S}_1 = \Phi; k := 1, \bar{\lambda}_1 = \Phi, q = 0.$

Step 2. *Compute* $r_i = b_i - a_i^T x_k$, $i = 1, \cdots, m$.
If $r_i \leq 0$, *stop.*
Choose p *such that* $r_p = \max_{1 \leq i \leq m} r_i$;
$\bar{\lambda}_k := \begin{pmatrix} \bar{\lambda}_k \\ 0 \end{pmatrix}$.

Step 3. $d_k := \hat{G}_k a_p = G^{-1}(I - A_k A_k^*)a_p$; $y_k := A_k^* a_p$.
If $\{j | (y_k)_j > 0, j \in \mathcal{S}_k\}$ *is nonempty, set*

$$\alpha_k = \min_{\substack{(y_k)_j > 0 \\ j \in \mathcal{S}_k}} \frac{(\bar{\lambda}_k)_j}{(y_k)_j} = \frac{(\bar{\lambda}_k)_l}{(y_k)_l}; \tag{9.5.28}$$

else set $\alpha_k = \infty$.

Step 4. *If* $d_k \neq 0$, *go to Step 5;*
If $\alpha_k = \infty$, *stop (the original problem has no feasible point);*
$\mathcal{S}_k := \mathcal{S}_k \setminus \{l\}$; $q := q - 1$;
$\bar{\lambda}_k := \bar{\lambda}_k + \alpha_k \begin{pmatrix} -y_k \\ 1 \end{pmatrix}$;
Modify A_k^* *and* $\hat{G}_k$; *turn to Step 3.*

Step 5 $\hat{\alpha} := -(b_p - a_p^T x_k)/a_p^T d_k$;
$\alpha_k := \min\{\alpha_k, \hat{\alpha}\}$;
$x_{k+1} := x_k + \alpha_k d_k$;
$f_{k+1} := f_k + \alpha_k a_p^T d_k (\frac{1}{2}\alpha_k + (\bar{\lambda}_k)_{q+1})$;
$\bar{\lambda}_{k+1} := \bar{\lambda}_k + \alpha_k \begin{pmatrix} -y_k \\ 1 \end{pmatrix}$.

Step 6. *If $\alpha_k < \hat{\alpha}$, go to Step 7;*
$\mathcal{S}_{k+1} := \mathcal{S}_k \cup \{p\}$; $q := q + 1$;
Compute $\hat{G}_{k+1}$ and A^*_{k+1}, $k := k + 1$; turn to Step 2.

Step 7. $\mathcal{S}_k := \mathcal{S}_k \setminus \{l\}$; $q := q - 1$;
Remove the l-th component from $\bar{\lambda}_k$ and obtain a new $\bar{\lambda}_k$;
Compute $\hat{G}_k$ and A^*_k, turn to Step 3. □

Next, we give a simple example which employs Algorithm 9.5.1.

Example 9.5.2

$$\begin{aligned} \min \quad & \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2 + \frac{1}{2}x_3^2 - 3x_2 - x_3 && (9.5.29)\\ \text{s.t.} \quad & -x_1 - x_2 - x_3 \geq -1, && (9.5.30)\\ & x_3 - x_2 \geq -1. && (9.5.31) \end{aligned}$$

Solution. This example is a modification of the problem (9.3.16)–(9.3.18). The unique solution is still $(-1, \frac{3}{2}, \frac{1}{2})^T$. By use of Algorithm 9.5.1, we have

$$x_1 = -G^{-1}g = \begin{pmatrix} 0 \\ 3 \\ 1 \end{pmatrix},$$

$$r_1 = 3 > 0,\ r_2 = 1 > 0.$$

Then we have $p = 1$ from Step 2, and

$$d_1 = G^{-1}a_p = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}.$$

Since $\mathcal{S}_1$ is empty, $\alpha_1 = \infty$ in Step 3. In Step 5, we get

$$\hat{\alpha} = -r_1/a_p^T d_1 = 1,$$

and obtain

$$\alpha_1 = 1,\ x_2 = x_1 + \alpha_1 d_k = \begin{pmatrix} -1 \\ 2 \\ 0 \end{pmatrix},$$

$$\bar{\lambda}_2 = (1),\ \mathcal{S}_2 = \{1\}.$$

Thus, after one iteration, x_2 is the solution of the subproblem

$$\min \quad \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2 + \frac{1}{2}x_3^2 - 3x_2 - x_3 \tag{9.5.32}$$

$$\text{s.t.} \quad -x_1 - x_2 - x_3 = -1. \tag{9.5.33}$$

In the second iteration, we have

$$r_1 = 0, \; r_2 = 1.$$

Then $p = 2$ from Step 2, and

$$d_2 = G^{-1}(I - \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \frac{1}{3} (1\ 1\ 1)) \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}.$$

Since $y_2 = a_2^T a_1 = 0$, we have $\alpha_2 = \infty$ in Step 3. In Step 5, we have

$$\hat{\alpha} = -r_2 / a_2^T d_2 = \frac{1}{2}.$$

Then $\alpha_2 := \hat{\alpha} = \frac{1}{2}$,

$$x_3 = x_2 + \alpha_2 d_2 = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ -\frac{3}{2} \\ \frac{1}{2} \end{pmatrix},$$

and $\bar{\lambda}_3 = (1\ \frac{1}{2})^T$. Hence, x_3 is the solution of the original problem and $\bar{\lambda}_3$ is the corresponding Lagrange multiplier. □

In concrete computation, Goldfarb and Idnani suggested using the Cholesky factorization of G,

$$G = LL^T,$$

and then employing QR decomposition to $L^{-1}A_k$, that is,

$$L^{-1}A_k = Q_k \begin{bmatrix} R_k \\ 0 \end{bmatrix}.$$

This approach allows us to get better numerical stability than by using G^{-1} directly.

Instead, Powell [274] suggested using

$$A_k = Q_k \begin{bmatrix} R_k \\ 0 \end{bmatrix} = [Q_k^{(1)}\, Q_k^{(2)}] \begin{bmatrix} R_k \\ 0 \end{bmatrix}, \tag{9.5.34}$$

and then employing the inverse Cholesky factorization of $[Q_k^{(2)}]^T G Q_k^{(2)}$, i.e.,

$$U_k U_k^T = [Q_k^{(2)}]^T G Q_k^{(2)}, \tag{9.5.35}$$

where U_k is an upper triangular matrix. In the algorithm that Powell [274] presented, each iteration updates $Q_k^{(1)}, R_k$, and U_k.

9.6 Interior Ellipsoid Method

Karmarkar [184] introduced a new polynomial-time algorithm for solving linear programming problems that sparked enormous interest in the mathematical programming community. Karmarkar's algorithm generates a sequence of points in the interior of the feasible region while converging to the optimal solution. This algorithm is effective and competitive with the simplex method in terms of solution time for linear programming (LP).

Ye and Tse [364] present an extension of Karmarkar's LP algorithm for convex quadratic programming. We introduce this algorithm in brief. The interested readers can consult the original paper for details.

The original version of Karmarkar's algorithm solves a linear programming of the special form

$$\min \quad \hat{c}^T \hat{x} \tag{9.6.1}$$

$$\text{s.t.} \quad \hat{A}^T \hat{x} = 0, e^T \hat{x} = n+1, \hat{x} \geq 0, \tag{9.6.2}$$

where $\hat{c} \in R^{n+1}, \hat{x} \in R^{n+1}, \hat{A} \in R^{(n+1)\times(m+1)}, e = (1, \cdots, 1)^T \in R^{n+1}$. Now, we generalize the Karmarkar's algorithm to convex quadratic programming.

Consider convex quadratic programming problem

$$\min \quad g^T x + \frac{1}{2} x^T G x \triangleq q(x) \tag{9.6.3}$$

$$\text{s.t.} \quad A^T x = b, \tag{9.6.4}$$

$$x \geq 0, \tag{9.6.5}$$

where $A \in R^{n\times m}$. Let x_k be an interior point, i.e.,

$$A^T x_k = b, \tag{9.6.6}$$
$$x_k > 0. \tag{9.6.7}$$

Define

$$D_k = \mathrm{diag}(x_k) = \begin{bmatrix} (x_k)_1 & & 0 \\ & \ddots & \\ 0 & & (x_k)_n \end{bmatrix}. \tag{9.6.8}$$

Let the transformation $\hat{x} = T_k x \in R^{n+1}$ as follows:

$$\hat{x}_i = \frac{(n+1)(D_k^{-1}x)_i}{e^T D_k^{-1} x + 1},\ i = 1, \cdots, n; \tag{9.6.9}$$
$$\hat{x}_{n+1} = (n+1)/[e^T D_k^{-1} x + 1]. \tag{9.6.10}$$

Obviously, the inverse transformation $T_k^{-1} : R^{n+1} \to R^n$ is defined by

$$x = T_k^{-1}\hat{x} = \frac{D_k \hat{x}[n]}{\hat{x}_{n+1}}, \tag{9.6.11}$$

where $e = (1, \cdots, 1)^T \in R^{n+1}, \hat{x}[n] = (\hat{x}_1, \cdots, \hat{x}_n)^T$. Then, the problem (9.6.3)–(9.6.5) can be written as

$$\min_{\hat{x}\in R^{n+1}} \quad \hat{x}_{n+1} q(T_k^{-1}\hat{x}) \triangleq \hat{q}(\hat{x}) \tag{9.6.12}$$
$$\text{s.t.} \quad A^T D_k \hat{x}[n] - \hat{x}_{n+1} b = 0, \tag{9.6.13}$$
$$e^T \hat{x} = n + 1, \tag{9.6.14}$$
$$\hat{x}[n] \geq 0,\ \hat{x} > 0. \tag{9.6.15}$$

By substituting (9.6.11) into (9.6.12), we obtain an equivalent form of (9.6.12)–(9.6.15):

$$\min \quad \hat{g}_k^T \hat{x}[n] + \frac{1}{2}\hat{x}[n]^T \hat{G}_k \hat{x}[n]/\hat{x}_{n+1} \tag{9.6.16}$$
$$\text{s.t.} \quad \hat{A}_k^T \hat{x} = \hat{b}, \tag{9.6.17}$$
$$\hat{x}[n] \geq 0,\ \hat{x}_{n+1} > 0, \tag{9.6.18}$$

where

$$\hat{G}_k = D_k G D_k,\ \hat{g}_k = D_k g, \tag{9.6.19}$$

$$\hat{A}_k = \begin{bmatrix} D_k A & \\ & e \\ -b^T & \end{bmatrix}, \ \hat{b} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ n+1 \end{pmatrix}. \tag{9.6.20}$$

Using the interior ellipsoid method, we solve the following subproblem (9.6.22)–(9.6.24) over an interior ellipsoid centered at $\hat{x}_k$, instead of solving the subproblem (9.6.12)–(9.6.15). Note that, for a given iterate x_k, $\hat{x}_k = T_k x_k = e$ and $\hat{q}(\hat{x}_k) = \hat{q}(e) = q(x_k)$. So, the interior ellipsoid happens to be an interior sphere in the feasible area of problem (9.6.12)–(9.6.15). Thus, the condition (9.6.18) can be enhanced to

$$\|\hat{x} - e\|_2 \leq \beta < 1. \tag{9.6.21}$$

Obviously, (9.6.18) will hold provided (9.6.21) holds. Hence, we consider the subproblem

$$\min \quad \hat{g}_k^T \hat{x}[n] + \frac{1}{2}\hat{x}[n] + \frac{1}{2}\hat{x}[n]^T \hat{G}_k \hat{x}[n]/\hat{x}_{n+1} \tag{9.6.22}$$

$$\text{s.t.} \quad \hat{A}_k^T \hat{x} = \hat{b}, \tag{9.6.23}$$

$$\|\hat{x} - e\|_2 \leq \beta < 1, \tag{9.6.24}$$

where $\beta < 1$ is a constant independent of k.

By the Karush-Kuhn-Tucker Theorem, solving (9.6.22)–(9.6.24) is equivalent to solving the following system:

$$\hat{g}_k + \hat{x}_{n+1}^{-1} \hat{G}_k \hat{x}[n] = \hat{A}_k[n]\lambda + \mu(\hat{x}[n] - e[n]), \tag{9.6.25}$$

$$-\frac{1}{2}\frac{1}{\hat{x}_{n+1}^2}\hat{x}[n]^T \hat{G}_k \hat{x}[n] = (\hat{a}_{n+1}^{(k)})^T \lambda + \mu(\hat{x}_{n+1} - 1) = 0, \tag{9.6.26}$$

$$\hat{A}_k^T \hat{x} = \hat{b}, \tag{9.6.27}$$

$$\|\hat{x} - e\|_2 \leq \beta, \tag{9.6.28}$$

$$\mu[\|\hat{x} - e\|_2 - \beta] = 0, \ \mu \leq 0, \tag{9.6.29}$$

where (9.6.25) and (9.6.26) are the first n equations and the last equation of the stationary point condition in KKT conditions respectively. Here $\hat{A}_k[n]$ is the matrix of the first n rows of matrix $\hat{A}_k$, $\hat{a}_{n+1}^{(k)}$ is the $(n+1)$-th row of $\hat{A}_k$, $e[n] = (1, \cdots, 1)^T \in R^n$ and $\lambda \in R^{m+1}$. So, (9.6.25) and (9.6.27) can be written in the following form

$$P_k \begin{bmatrix} \hat{x}[n] \\ \hat{\lambda} \end{bmatrix} = \hat{x}_{n+1}\bar{b} + \tilde{b}, \tag{9.6.30}$$

where

$$P_k = \begin{bmatrix} \hat{G}_k + \hat{\mu} I & -\hat{A}[n] \\ \hat{A}[n]^T & 0 \end{bmatrix}, \tag{9.6.31}$$

$$\bar{b} = \begin{bmatrix} -\hat{g}_k \\ b \\ -1 \end{bmatrix}, \ \tilde{b} = \begin{bmatrix} \hat{\mu} e \\ 0 \\ n+1 \end{bmatrix}, \tag{9.6.32}$$

$$\hat{\lambda} = \hat{x}_{n+1} \lambda, \ \hat{\mu} = -\hat{x}_{n+1} \mu. \tag{9.6.33}$$

Then, for any given $\hat{\mu} \geq 0$, we can find $\hat{\lambda}$ and $\hat{x}[n]$ from (9.6.30). Then we obtain $\hat{x}_{n+1}$ from substituting $\hat{\lambda}$ and $\hat{x}[n]$ into (9.6.26). This indicates that for any given $\hat{\mu} \geq 0$, we can find $\hat{x}(\hat{\mu})$. Define the function

$$h(\hat{\mu}) = \|\hat{x}(\hat{\mu}) - e\|_2 - \beta. \tag{9.6.34}$$

If $h(0) \leq 0$, then $\hat{x}(0)$ is the solution of (9.6.12)–(9.6.15). In this case, $x = D_k \hat{x}(0)[n] / \hat{x}(0)_{n+1}$ is the solution of the original problem.

If $h(0) > 0$, since $\lim_{\hat{\mu} \to \infty} h(\hat{\mu}) = -\beta < 0$, we can find $\hat{\mu}_k$ by a bisectioning method such that $h(\hat{\mu}_k) = 0$, and further the solution $\hat{x}(\hat{\mu}_k)$ of problem (9.6.22)–(9.6.24). By back-substituting $\hat{x}(\hat{\mu}_k)$, we obtain the new iterate x_{k+1}, i.e.,

$$x_{k+1} = T_k^{-1} \hat{x}(\hat{\mu}_k) = \frac{D_k \hat{x}(\hat{\mu}_k)[n]}{\hat{x}(\hat{\mu}_k)_{n+1}}, \tag{9.6.35}$$

where $\hat{x}(\hat{\mu}_k)[n] = (\hat{x}(\hat{\mu}_k)_1, \cdots, \hat{x}(\hat{\mu}_k)_n)^T$.

The interior ellipsoid algorithm for solving convex quadratic programming problems is introduced as follows.

Algorithm 9.6.1 *(Interior Ellipsoid Method for Convex QP)*

Step 1. Given a strict interior point x_1 of (9.6.3)–(9.6.5); $k := 1$.

Step 2. Solve the subproblem (9.6.22)–(9.6.24) for $\hat{x}(\hat{\mu}_k)$; and compute x_{k+1} by (9.6.35).

Step 3. If x_{k+1} is a KKT point, stop;
$k := k + 1$, go to Step 2. □

The further details of interior ellipsoid methods for convex quadratic programming can be found in Ye and Tse (1989).

9.7 Primal-Dual Interior-Point Methods

The primal-dual interior-point method for linear programming can be applied to convex quadratic programming through a simple extension of the method. Since we have not discussed the topic *linear programming* in the book, we first outline this method for linear programming.

Consider the linear programming problem in standard form

$$\begin{array}{ll} \min_{x\in R^n} & c^T x \\ \text{s.t.} & Ax = b, \\ & x \geq 0, \end{array} \tag{9.7.1}$$

where c and x are vectors in R^n, b is a vector in R^m, and A is an $m \times n$ matrix. The dual problem for (9.7.1) is

$$\begin{array}{ll} \max_{\lambda\in R^m} & b^T \lambda \\ \text{s.t.} & A^T\lambda + s = c, \\ & s \geq 0, \end{array} \tag{9.7.2}$$

where λ is a vector in R^m and s is a vector in R^n. The primal-dual solution of (9.7.1) and (9.7.2) are characterized by the Karush-Kuhn-Tucker (KKT) conditions:

$$A^T\lambda + s = c, \tag{9.7.3}$$

$$Ax = b, \tag{9.7.4}$$

$$x_i s_i = 0, \quad i = 1, \cdots, n \tag{9.7.5}$$

$$(x, s) \geq 0, \tag{9.7.6}$$

where vectors λ and s are Lagrange multipliers for the constraints $Ax = b$ and $x \geq 0$, respectively.

Primal-dual interior-point methods find primal-dual solutions (x^*, λ^*, s^*) of KKT system by applying variants of Newton's method to the three equality conditions (9.7.3)–(9.7.5) of this system and modifying the search directions and steplength so that the inequalities $(x, s) \geq 0$ are satisfied strictly at every iteration.

To derive primal-dual interior-point methods, we restate the KKT conditions (9.7.3)–(9.7.6) in a slightly different form by means of a mapping

$F : R^{2n+m} \to R^{2n+m}$:

$$F(x, \lambda, s) = \begin{bmatrix} A^T\lambda + s - c \\ Ax - b \\ XSe \end{bmatrix} = 0, \tag{9.7.7}$$

$$(x, s) \geq 0, \tag{9.7.8}$$

where

$$X = \text{diag}(x_1, x_2, \cdots, x_n), \quad S = \text{diag}(s_1, s_2, \cdots, s_n),$$

and $e = (1, 1, \cdots, 1)^T$. Note that F is actually linear in its first two terms $Ax - b, A^T\lambda + s - c$, and only mildly nonlinear in the remaining term XSe.

Primal-dual interior-point methods generate iterates (x^k, λ^k, s^k) that satisfy the bound (9.7.8) strictly, that is, $x^k > 0$ and $s^k > 0$. This property is the origin of the term *interior-point*. By respecting these bounds, the methods avoid spurious solutions, which are points that satisfy $F(x, \lambda, s) = 0$ but not $(x, s) \geq 0$.

Newton's method forms a linear model of F around the current point and obtains the search direction $(\Delta x, \Delta\lambda, \Delta s)$ by solving the following system of linear equations:

$$J(x, \lambda, s) \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = -F(x, \lambda, s), \tag{9.7.9}$$

where J is the Jacobian of F. If the current point is strictly feasible, the Newton step equations become

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -XSe \end{bmatrix}. \tag{9.7.10}$$

Note that a full step along this direction usually is not permissible, since it would violate the bound $(x, s) \geq 0$. To avoid this difficulty, we perform a line search along the Newton direction so that the new iterate is

$$(x, \lambda, s) + \alpha(\Delta x, \Delta\lambda, \Delta s) \tag{9.7.11}$$

for some line search parameter $\alpha \in (0, 1]$. Unfortunately, we often can take only a small step along the direction ($\alpha \ll 1$) before violating the condition $(x, s) > 0$. Hence the pure Newton direction (9.7.10) often does not allow us to make much progress toward a solution.

In the following, we give the central path technique which modifies the basic Newton procedure.

The Central Path

The central path $\mathcal{C}$ is an arc of strictly feasible points that plays a vital role in primal-dual algorithms. It is parametrized by a scalar $\tau > 0$, and each point $(x_\tau, \lambda_\tau, s_\tau) \in \mathcal{C}$ solves the following system:

$$A^T\lambda + s = c, \tag{9.7.12}$$

$$Ax = b, \tag{9.7.13}$$

$$x_i s_i = \tau, \quad i = 1, 2, \cdots, n, \tag{9.7.14}$$

$$(x, s) > 0. \tag{9.7.15}$$

These conditions differ from KKT conditions only in the term τ on the right-hand side of (9.7.14). Instead of the complementarity condition (9.7.5), we require that the pairwise product $x_i s_i$ have the same value τ for all indices i. From (9.7.12)–(9.7.15), we can define the central path as

$$\mathcal{C} = \{(x_\tau, \lambda_\tau, s_\tau) \mid \tau > 0\}.$$

It can be shown that $(x_\tau, \lambda_\tau, s_\tau)$ is defined uniquely for each $\tau > 0$ if and only if the strictly feasible set $\mathcal{F}^o$ defined by

$$\mathcal{F}^o = \{(x, \lambda, s) \mid Ax = b, A^T\lambda + s = c, (x, s) > 0\}$$

is nonempty. Hence the entire path $\mathcal{C}$ is well defined.

Another way of defining $\mathcal{C}$ is to use the mapping F defined in (9.7.7) and write

$$F(x_\tau, \lambda_\tau, s_\tau) = \begin{bmatrix} 0 \\ 0 \\ \tau e \end{bmatrix}, \quad (x_\tau, s_\tau) > 0. \tag{9.7.16}$$

The equations (9.7.12)–(9.7.15) approximate (9.7.3)–(9.7.6) more and more closely as τ goes to zero. If $\mathcal{C}$ converges to anything as $\tau \downarrow 0$, it must converge to a primal-dual solution of the linear program. The central path thus guides us to a solution along a route that steers clear of spurious solutions by keeping all the pairwise products $x_i s_i$ strictly positive and decreasing them to zero at the same rate.

Primal-dual interior-point algorithms take Newton steps toward points on $\mathcal{C}$ for which $\tau > 0$, rather than pure Newton steps for F. Since these steps are

biased toward the interior of the nonnegative orthant defined by $(x, s) \geq 0$, it usually is possible to take longer steps along them than along the pure Newton steps for F before violating the positivity condition. To describe the biased search direction, we introduce a centering parameter $\sigma \in [0, 1]$ and a duality measure μ defined by

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i s_i = \frac{x^T s}{n}, \tag{9.7.17}$$

which measures the average value of the pairwise product $x_i s_i$. By writing $\tau = \sigma\mu$ and applying Newton's method to the system (9.7.16), we obtain

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -XSe + \sigma\mu e \end{bmatrix}. \tag{9.7.18}$$

The step $(\Delta x, \Delta\lambda, \Delta s)$ is a Newton step toward the point $(x_{\sigma\mu}, \lambda_{\sigma\mu}, s_{\sigma\mu}) \in \mathcal{C}$, at which the pairwise product $x_i s_i$ are all equal to $\sigma\mu$. In contrast, the step (9.7.10) aims directly for the point at which the KKT conditions (9.7.3)–(9.7.6) are satisfied.

If $\sigma = 1$, the equations (9.7.18) define a *centering direction*, a Newton step toward the point $(x_\mu, \lambda_\mu, s_\mu) \in \mathcal{C}$. If $\sigma = 0$, the (9.7.18) gives the standard Newton step.

In the following, we define a general framework of primal-dual interior-point algorithm.

Algorithm 9.7.1 *(A Primal-Dual Interior-Point Framework)*

Given $(x^0, \lambda^0, s^0) \in \mathcal{F}^o$.

For $k = 0, 1, 2, \cdots,$

solve

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta\lambda^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -X^k S^k e + \sigma_k \mu_k e \end{bmatrix},$$

where $\sigma_k \in [0, 1]$ *and* $\mu_k = (x^k)^T s^k / n$;

set

$$(x^{k+1}, \lambda^{k+1}, s^{k+1}) = (x^k, \lambda^k, s^k) + \alpha_k (\Delta x^k, \Delta\lambda^k, \Delta s^k),$$

choosing α_k *such that* $(x^{k+1}, s^{k+1}) > 0$.

end(For). □

For most problems, however, a strictly feasible starting point (x^0, λ^0, s^0) is difficult to find. Infeasible interior-point methods require only that the components of x^0 and s^0 be strictly positive. Therefore, we give a slight change to the equation (9.7.18). If we define the residuals for the two linear equations as

$$r_b = Ax - b, \quad r_c = A^T\lambda + s - c,$$

the modified step equation is

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe + \sigma\mu e \end{bmatrix}. \tag{9.7.19}$$

Primal-Dual Interior-Point Methods for Convex Quadratic Programming

Now we return to convex quadratic programming. Let us discuss convex quadratic programming with inequality constraints:

$$\min_{x\in R^n} \quad q(x) \stackrel{\text{def}}{=} \frac{1}{2}x^T Gx + x^T g \tag{9.7.20}$$

$$\text{s.t.} \qquad Ax \geq b, \tag{9.7.21}$$

where $g \in R^n, b \in R^m, A \in R^{m\times n}$ and $G \in R^{n\times n}$ is symmetric and positive semidefinite. The KKT conditions of (9.7.20)–(9.7.21) state as follows: If x^* is a solution of (9.7.20)–(9.7.21), there is a Lagrange multiplier vector λ^* such that the following conditions are satisfied for $(x, \lambda) = (x^*, \lambda^*)$:

$$Gx - A^T\lambda + g = 0, \tag{9.7.22}$$

$$Ax - b \geq 0, \tag{9.7.23}$$

$$(Ax - b)_i\lambda_i = 0, \quad i = 1, 2, \cdots, m, \tag{9.7.24}$$

$$\lambda \geq 0. \tag{9.7.25}$$

By introducing the slack vector $y = Ax - b$, we have

$$Gx - A^T\lambda + g = 0, \tag{9.7.26}$$

$$Ax - y - b = 0, \tag{9.7.27}$$

$$y_i\lambda_i = 0, \quad i = 1, 2, \cdots, m, \tag{9.7.28}$$

$$(y, \lambda) \geq 0. \tag{9.7.29}$$

As in the case of linear programming, here the KKT conditions are not only necessary but also sufficient, because problem (9.7.20)–(9.7.21) is convex programming. Hence we can solve it by finding solutions of system (9.7.26)–(9.7.29). As discussed above, we apply modifications of Newton's method to this system. We can define

$$F(x,y,\lambda)=\begin{bmatrix} Gx-A^T\lambda+g \\ Ax-y-b \\ Y\Lambda e \end{bmatrix}, \quad (y,\lambda)\geq 0, \tag{9.7.30}$$

where

$$Y=\mathrm{diag}(y_1,y_2,\cdots,y_m), \quad \Lambda=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_m), \quad e=(1,1,\cdots,1)^T.$$

Given a current iterate (x,y,λ) that satisfies $(y,\lambda)>0$, we can define a duality measure μ by

$$\mu=\frac{1}{m}\sum_{i=1}^{m}y_i\lambda_i=\frac{y^T\lambda}{m}. \tag{9.7.31}$$

The central path $\mathcal{C}$ is the set of points $(x_\tau,y_\tau,\lambda_\tau)(\tau>0)$ satisfying

$$F(x_\tau,y_\tau,\lambda_\tau)=\begin{bmatrix} 0 \\ 0 \\ \tau e \end{bmatrix}, \quad (y_\tau,\lambda_\tau)>0. \tag{9.7.32}$$

The generic step $(\Delta x,\Delta y,\Delta\lambda)$ is a Newton-type step toward the point $(x_{\sigma\mu},y_{\sigma\mu},\lambda_{\sigma\mu})\in\mathcal{C}$. As in (9.7.19), this step satisfies the following system:

$$\begin{bmatrix} G & -A^T & 0 \\ A & 0 & -I \\ 0 & Y & \Lambda \end{bmatrix}\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\lambda \end{bmatrix}=\begin{bmatrix} -r_g \\ -r_b \\ -\Lambda Se+\sigma\mu e \end{bmatrix}, \tag{9.7.33}$$

where

$$r_g=Gx-A^T\lambda+g, \quad r_b=Ax-y-b.$$

So, we obtain the next iterate

$$(x^+,y^+,\lambda^+)=(x,y,\lambda)+\alpha(\Delta x,\Delta y,\Delta\lambda), \tag{9.7.34}$$

where α is chosen so that $(y^+,\lambda^+)>0$.

For mode details of primal-dual interior-point methods for convex quadratic programming, please consult Wright [358].

Exercises

1. Let $H = \text{Diag}(h_{11}, h_{22}, ..., h_{nn})$ be a positive definite diagonal matrix. Find the minimizer of (9.1.1) subject to the condition $\|x\|_\infty \leq 1$.

2. Prove Theorem 9.1.1.

3. Prove that problem (9.2.8)–(9.2.10) is the dual of problem (9.1.1)–(9.1.3).

4. Solve the dual of the problem

$$\begin{array}{ll} \min & (x_1^2 + x_2^2)/2 + x_1 \\ \text{s.t.} & x_1 \geq 0. \end{array}$$

5. If $f(x)$ is a convex function and $c_i(x)(i = 1, \cdots, m)$ are concave functions, the problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & c_i(x) \geq 0 \end{array}$$

is called a convex programming problem. Generalize the dual theory in Section 9.2 from convex quadratic programming to general convex programming.

6. Find the smallest circle in the plane that contains the points $(1, -4)$, $(-2, -2)$, $(-4, 1)$ and $(4, 5)$. Formulate the problem as a convex programming problem, then solve the dual.

7. Assume that $B \in R^{n\times n}$ is positive definite, $A \in R^{m\times n}$, $g \in R^n$ and $b \in R^m$. Give the dual problem of the following QP:

$$\begin{array}{ll} \min & g^T x + \dfrac{1}{2} x^T B x \\ \text{s.t.} & Ax = b. \end{array}$$

8. Solve the equality constraint QP problem

$$\begin{array}{ll} \min & \begin{pmatrix} 1 \\ -1 \end{pmatrix}^T x + \dfrac{1}{2} x^T \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix} x \\ \text{s.t.} & x_1 + x_2 = 1. \end{array}$$

9. 1) Check that if set $V = \begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix}$ in (9.3.42), the choice (9.3.37)–(9.3.38) can be obtained.

2) Check that if set $V = Q_2$ in (9.3.42), the choice (9.3.40) can be obtained.

10. Show that if $A \in R^{n\times m}$ has full column-rank and Z^TGZ is positive definite, then KKT matrix (9.3.47) is nonsingular.

11. Show (9.3.53)–(9.3.55).

12. Show (9.3.58)–(9.3.60).

13. Assume $A \in R^{m\times n}$ has full row rank. Let $Z \in R^{n\times(n-m)}$ be any full column rank matrix such that $AZ = 0$. Prove that the matrix

$$\begin{bmatrix} B & A^T \\ A & 0 \end{bmatrix} \tag{9.7.35}$$

is nonsingular if and only if Z^TBZ is nonsingular.

14. Use the active set method to solve the problem

$$\begin{aligned} \min \quad & -1000x_1 - 1000x_2 + x_1^2 + x_2^2 \\ \text{s.t.} \quad & 3x_1 + x_2 \geq 3, \\ & x_1 + 4x_2 \geq 4, \\ & x_1 \geq 0, \\ & x_2 \geq 0. \end{aligned}$$

Illustrate the result by sketching the set of feasible solutions.

15. Program Algorithm 9.4.2 and use it to solve

$$\begin{aligned} \min \quad & x_1^2 + 2x_2^2 - 2x_1 - 6x_2 - 2x_1x_2 \\ \text{s.t.} \quad & \frac{1}{2}x_1 + \frac{1}{2}x_2 \leq 1, \\ & -x_1 + x_2 \leq 2, \\ x_1, x_2 \geq 0. & \end{aligned}$$

16. Try to give the primal-dual interior-point algorithm for convex quadratic programming (9.7.20)–(9.7.21).

Chapter 10

Penalty Function Methods

10.1 Penalty Function

The penalty function methods are an important class of methods for constrained optimization problem

$$\begin{aligned} \min_{x\in R^n} \quad & f(x) && (10.1.1)\\ \text{s.t.} \quad & c_i(x) = 0,\ i \in E \stackrel{Def}{=} \{1,\cdots,m_e\}, && (10.1.2)\\ & c_i(x) \geq 0,\ i \in I \stackrel{Def}{=} \{m_e+1,\cdots,m\}. && (10.1.3) \end{aligned}$$

In this class of methods we replace the original constrained problem by a sequence of unconstrained subproblems that minimizes the penalty functions. The penalty function is a function with penalty property

$$P(x) = \bar{P}(f(x), c(x)), \tag{10.1.4}$$

constructed from the objective function $f(x)$ and the constraints $c(x)$. The so-called "penalty" property requires $P(x) = f(x)$ for all feasible points $x \in X$ of (10.1.1)–(10.1.3), and $P(x)$ is much larger than $f(x)$ when the constraint violations are severe.

To describe the degree of constraint violation, we define the constraint violation function $c^{(-)}(x) = (c_1^{(-)}(x),\cdots,c_m^{(-)}(x))^T$ as follows:

$$\begin{aligned} c_i^{(-)}(x) &= c_i(x),\ i \in E, && (10.1.5)\\ c_i^{(-)}(x) &= \min\{c_i(x), 0\},\ i \in I. && (10.1.6) \end{aligned}$$

Define

$$C = \{c_i(x) \,|\, c_i(x) = 0, i \in E;\ c_i(x) \geq 0, i \in I\}. \tag{10.1.7}$$

Obviously, x is a feasible point if and only if $c(x) \in C$. Furthermore,

$$\text{if } c_i(x) \geq 0, \text{ i.e., } c_i(x) \in C, \text{ then } c_i^{(-)}(x) = 0;$$

$$\text{if } c_i(x) < 0, \text{ i.e., } c_i(x) \notin C, \text{ then } c_i^{(-)}(x) \neq 0.$$

This means, for each constraint, the constraint violation function is nonzero when the corresponding constraint is violated and zero when the corresponding constraint is feasible.

It is not difficult to see that for any $x \in R^n$, we have

$$\|c^{(-)}(x)\|_2 = \text{dist}\,(c(x), C), \tag{10.1.8}$$

where $\text{dist}(\cdot, \cdot)$ denotes the distance from a point to a set and is defined as

$$\text{dist}(x, Y) = \min\{\|x - y\|_2 \mid \forall y \in Y\}. \tag{10.1.9}$$

The penalty function consists of a sum of the original objective function and a penalty term, i.e.,

$$P(x) = f(x) + h(c^{(-)}(x)), \tag{10.1.10}$$

where the penalty term $h(c^{(-)}(x))$ is a function defined on R^m and satisfies

$$h(0) = 0, \lim_{\|c\| \to +\infty} h(c) = +\infty. \tag{10.1.11}$$

The earliest penalty function is the Courant penalty function, or called the quadratic penalty function, defined as

$$P(x) = f(x) + \sigma \|c^{(-)}(x)\|_2^2, \tag{10.1.12}$$

where $\sigma > 0$ is a positive constant, which is called the penalty parameter. We give an example to describe the penalty function.

$$\begin{aligned} \min \quad & x \\ \text{s.t.} \quad & x - 2 \geq 0. \end{aligned} \tag{10.1.13}$$

Then

$$h(c^{(-)}(x)) = \|c^{(-)}(x)\|_2^2 = [\min\{0, x-2\}]^2 = \begin{cases} 0 & \text{if } x \geq 2, \\ (x-2)^2 & \text{if } x < 2. \end{cases}$$

Note that the minimum of $f(x) + \sigma\|c^{(-)}(x)\|^2$ occurs at the point $2 - \frac{1}{\sigma}$, and approaches the minimum point $\bar{x} = 2$ of the original problem, as σ approaches ∞.

Obviously, (10.1.12) is a particular case of (10.1.10) in which $h(c) = \sigma\|c\|_2^2$. In fact, for any norm on R^m and any $\alpha > 0$, the function $h(c) = \sigma\|c\|^\alpha$ satisfies (10.1.11). So, a class of penalty functions can be defined as:

$$P(x) = f(x) + \sigma\|c^{(-)}(x)\|^\alpha, \tag{10.1.14}$$

where $\sigma > 0$ is a penalty parameter, $\alpha > 0$, and $\|\cdot\|$ is some norm on R^m. Typically, (10.1.12) is often written as

$$P(x) = f(x) + \frac{1}{2}\sigma\|c^{(-)}(x)\|^2 \tag{10.1.15}$$

$$= f(x) + \frac{1}{2}\sigma\sum_{i=1}^{m_e} c_i^2(x) + \frac{1}{2}\sigma\sum_{i=m_e+1}^{m}[c_i^{(-)}(x)]^2 \tag{10.1.16}$$

and is called the quadratic penalty function, where $\sigma > 0$ and $c_i^{(-)}(x) = \min\{0, c_i(x)\}$.

Besides (10.1.12), the common particular forms of (10.1.14) are

$$P_1(x) = f(x) + \sigma\|c^{(-)}(x)\|_1 \tag{10.1.17}$$

and

$$P_\infty(x) = f(x) + \sigma\|c^{(-)}(x)\|_\infty, \tag{10.1.18}$$

which are called L_1 penalty function and L_∞ penalty function respectively.

If the penalty function takes values approaching $+\infty$ as x approaches the boundary of the feasible region, it is called the interior point penalty function. The interior point penalty function is suitable only to inequality-constrained problems, i.e., $m_e = 0$. Typically, the two most important interior point penalty functions are the inverse barrier function

$$P(x) = f(x) + \frac{1}{\sigma}\sum_{i=1}^{m}\frac{1}{c_i(x)} \tag{10.1.19}$$

and the logarithmic barrier function

$$P(x) = f(x) - \frac{1}{\sigma}\sum_{i=1}^{m} \log c_i(x). \tag{10.1.20}$$

If given an initial point in the interior of the feasible region, the whole sequence generated by the interior point penalty function method is interior points. Since these functions set an infinitely high "barrier" on the boundary, they are also said to be barrier functions.

Let x^* be a KKT point of constrained optimization (10.1.1)–(10.1.3). Then it follows from (10.1.12) that $\nabla P(x^*) = \nabla f(x^*)$. In general, x^* is not a stationary point of the Courant penalty function, and the penalty function method attempts to create a local minimizer at x^* in the limit $\sigma_k \to \infty$. To overcome this shortcoming, we introduce parameters $\theta_i\,(i = 1, \cdots, m)$ with $\theta_i \geq 0\ (i = m_e + 1, \cdots, m)$ to change the origin of the penalty term. Write $\theta = (\theta_1, \cdots, \theta_m)^T$. Modifying (10.1.12) gives

$$\begin{aligned} P(x) &= f(x) + \sum_{i=1}^{m} \frac{\sigma_i}{2}\left(\left[(c(x) - \theta)_i^{(-)}\right]^2 - \theta_i^2\right) \\ &= f(x) + \sum_{i=1}^{m_e}\left[-\lambda_i c_i(x) + \frac{1}{2}\sigma_i (c_i(x))^2\right] \\ &+ \sum_{i=m_e+1}^{m} \begin{cases} -\lambda_i c_i(x) + \frac{1}{2}\sigma_i (c_i(x))^2, & \text{if } \ c_i(x) < \frac{\lambda_i}{\sigma_i}; \\ -\frac{1}{2}\lambda_i^2/\sigma_i, & \text{otherwise} \end{cases} \end{aligned} \tag{10.1.21}$$

where

$$\lambda_i = \sigma_i \theta_i,\ i = 1, \cdots, m. \tag{10.1.22}$$

Because the penalty function (10.1.21) can be obtained from Lagrange function (8.2.18) by adding a penalty term, (10.1.21) is referred to as an augmented Lagrangian function. Alternatively, (10.1.21) can be also obtained from the penalty function (10.1.12) by adding a multiplier term $-\lambda^T c$, (10.1.21) is also called a multiplier penalty function. Let x^* be a KKT point of the constrained optimization problem, and $\lambda_i^*\ (i = 1, \cdots, m)$ corresponding Lagrange multipliers. Then the augmented Lagrangian function with $\lambda_i^*\ (i = 1, \cdots, m)$ satisfies $\nabla P(x^*) = 0$. In addition, the Lagrange multiplier λ^* is not known in advance, so the augmented Lagrangian function method needs to update $\lambda_i\ (i = 1, \cdots, m)$ successfully.

For equality-constrained problem ($m_e = m$), we define

$$\lambda(x) = (A(x))^{+} g(x), \tag{10.1.23}$$

where $A(x) = (\nabla c_1(x), \cdots, \nabla c_m(x))$ is an $n \times m$ matrix, $g(x) = \nabla f(x)$, A^+ denotes the generalized inverse of A, and the multiplier $\lambda(x)$ is the minimum l_2 norm solution of the least-squares problem

$$\min_{\lambda \in R^m} \left\| \nabla f(x) - \sum_{i=1}^{m} \lambda_i \nabla c_i(x) \right\|_2^2. \tag{10.1.24}$$

By using (10.1.23), we can give Fletcher's smooth exact penalty function (or Fletcher's augmented Lagrangian function) for the case where only equality constraints are present in (10.1.1)–(10.1.3) as follows:

$$P(x) = f(x) - \lambda(x)^T c(x) + \frac{1}{2} \sum_{i=1}^{m} \sigma_i (c_i(x))^2, \tag{10.1.25}$$

where $\sigma_i > 0$ $(i = 1, \cdots, m)$ are penalty parameters.

Let x^* be the solution of the equality-constrained problem, $A(x^*)$ have full column rank,

$$\begin{aligned} \nabla_x P(x^*) &= g(x^*) - A(x^*)\lambda^* = 0, \\ \nabla^2_{xx} P(x^*) &= W^* + A(x^*)A(x^*)^+ W^* \\ &\quad + W^* A(x^*)A(x^*)^+ + A(x^*) D A(x^*)^T, \end{aligned} \tag{10.1.26}$$

where $D = \mathrm{diag}(\sigma_1, \cdots, \sigma_m)$ and

$$W^* = \nabla^2 f(x^*) - \sum_{i=1}^{m} \lambda_i^* \nabla^2 c_i(x^*) = \nabla^2_{xx} L(x^*, \lambda^*).$$

Lemma 10.1.1 *Let $H \in R^{n \times n}$ be symmetric and $A \in R^{n \times m}$. If*

$$d^T H d > 0 \tag{10.1.27}$$

for any nonzero vector d with $A^T d = 0$, then there exists $\sigma \geq 0$ such that

$$H + \sigma A A^T \tag{10.1.28}$$

is a positive definite matrix.

Proof. By assumption, there is $\delta > 0$, such that if $\|A^T d\|_2 \leq \delta$ and $\|d\|_2 = 1$, then (10.1.27) holds. Consider

$$\min_{\substack{\|A^T d\|_2 \geq \delta \\ \|d\|_2 = 1}} \frac{d^T H d}{\|A^T d\|_2^2}. \tag{10.1.29}$$

Since the set $\{d \,|\, \|d\|_2 = 1,\ \|A^T d\|_2 \geq \delta\}$ is a finite and closed set, then the minimum of (10.1.29) is achieved. Hence there is η such that for all $d \in \{d \,|\, \|d\|_2 = 1,\ \|A^T d\|_2 \geq \delta\}$ we have

$$\frac{d^T H d}{\|A^T d\|_2^2} > \eta. \tag{10.1.30}$$

Set $\sigma = \max\{-\eta, 0\}$. Therefore, for any d with $\|d\|_2 = 1$, we have

$$d^T (H + \sigma A A^T) d > 0. \tag{10.1.31}$$

We complete the proof. □

If the second-order sufficient condition

$$d^T W^* d > 0,\ \forall d \neq 0,\ (A^*)^T d = 0 \tag{10.1.32}$$

holds, then it follows from the above lemma that there exists $\bar{\sigma} \geq 0$ such that for all $\sigma_i \geq \bar{\sigma}$, the matrix $\nabla^2_{xx} P(x^*)$ defined by (10.1.26) is positive definite. Therefore the penalty function (10.1.25) is said to be exact if the solution x^* of the original problem is also the strict local minimizer of the penalty function $P(x^*)$.

If Lagrange multipliers λ_i of the augmented Lagrangian function are the Lagrange multipliers λ_i^* at the solution x^* of the problem, then under the second-order sufficient condition (10.1.32), x^* is also the local minimizer of the augmented Lagrangian function (10.1.21) when σ is large enough. Thus, the augmented Lagrangian function is also an exact penalty function.

For an L_1 penalty function, if

$$\sigma > \|\lambda^*\|_\infty, \tag{10.1.33}$$

then under the second-order sufficient condition (10.1.32), the solution x^* of the original problem is also a strict local minimizer of the L_1 penalty function. Thus, the L_1 penalty function is referred to as an L_1 exact penalty function. Similarly, an L_∞ penalty function is also exact.

Note that the KKT point of the constrained optimization problem is not, in general, the stationary point of the Courant penalty function. Thus, the Courant penalty function is not an exact penalty function.

In this chapter, we will describe the simple penalty function method, interior point penalty function method (i.e., barrier function method), multiplier penalty function method, smooth exact penalty function method and non-smooth exact penalty function method.

10.2 The Simple Penalty Function Method

The penalty function method is an approach to minimize a sequence of penalty functions and obtain the minimizer of the original constrained optimization problem.

Consider the simple penalty function

$$P_\sigma(x) = f(x) + \sigma \|c^{(-)}(x)\|^\alpha, \tag{10.2.1}$$

where $\sigma > 0$ is the penalty parameter, $\alpha > 0$ a positive constant, and $\|\cdot\|$ a given norm on R^m. Write $x(\sigma)$ as a solution of problem

$$\min_{x \in R^n} P_\sigma(x). \tag{10.2.2}$$

Next, we first give some lemmas.

Lemma 10.2.1 *Let $0 < \sigma_1 < \sigma_2$. Then*

$$\begin{aligned} P_{\sigma_1}(x(\sigma_1)) &\leq P_{\sigma_2}(x(\sigma_2)), & (10.2.3)\\ f(x(\sigma_1)) &\leq f(x(\sigma_2)), & (10.2.4)\\ \|c^{(-)}(x(\sigma_1))\| &\geq \|c^{(-)}(x(\sigma_2))\|. & (10.2.5) \end{aligned}$$

Proof. From the definition of $x(\sigma)$, we have

$$P_{\sigma_1}(x(\sigma_1)) \leq P_{\sigma_1}(x(\sigma_2)) \leq P_{\sigma_2}(x(\sigma_2)) \leq P_{\sigma_2}(x(\sigma_1)), \tag{10.2.6}$$

which shows (10.2.3). By use of (10.2.6) and (10.2.1), we have

$$\begin{aligned} 0 &\leq P_{\sigma_1}(x(\sigma_2)) - P_{\sigma_2}(x(\sigma_2)) - [P_{\sigma_1}(x(\sigma_1)) - P_{\sigma_2}(x(\sigma_1))]\\ &= (\sigma_1 - \sigma_2)[\|c^{(-)}(x(\sigma_2))\|^\alpha - \|c^{(-)}(x(\sigma_1))\|^\alpha], \end{aligned} \tag{10.2.7}$$

which means that (10.2.5) holds. Using (10.2.6) and (10.2.5) gives

$$\begin{aligned} f(x(\sigma_1)) &\leq f(x(\sigma_2)) + \sigma_1(\|c^{(-)}(x(\sigma_2))\|^\alpha - \|c^{(-)}(x(\sigma_1))\|^\alpha)\\ &\leq f(x(\sigma_2)). \end{aligned} \tag{10.2.8}$$

Hence (10.2.4) holds. We complete the proof. □

Lemma 10.2.2 *Let $\delta = \|c^{(-)}(x(\sigma))\|$. Then $x(\sigma)$ is also the solution of problem*

$$\begin{aligned} \min\nolimits_{x \in R^n} \quad & f(x) & (10.2.9)\\ \text{s.t.} \quad & \|c^{(-)}(x)\| \leq \delta. & (10.2.10) \end{aligned}$$

Proof. For any x satisfying (10.2.10), we have

$$\begin{aligned} 0 &\leq \sigma(\|c^{(-)}(x(\sigma))\|^\alpha - \|c^{(-)}(x)\|^\alpha) \\ &= P_\sigma(x(\sigma)) - f(x(\sigma)) - P_\sigma(x) + f(x) \\ &= [P_\sigma(x(\sigma)) - P_\sigma(x)] + f(x) - f(x(\sigma)) \\ &\leq f(x) - f(x(\sigma)). \end{aligned} \tag{10.2.11}$$

Hence, for any x satisfying (10.2.10), we have

$$f(x) \geq f(x(\sigma)), \tag{10.2.12}$$

which shows that $x(\sigma)$ is the solution of (10.2.9)–(10.2.10). □

By the definition of the constraint violation function $c^{(-)}(x)$, the original problem (10.1.1)–(10.1.3) can be written equivalently as

$$\min_{x \in R^n} \quad f(x), \tag{10.2.13}$$

$$\text{s.t.} \quad \|c^{(-)}(x)\| = 0. \tag{10.2.14}$$

Hence, if δ is sufficiently small, the problem (10.2.9)–(10.2.10) can be regarded as an approximation of (10.2.13)–(10.2.14), and so $x(\sigma)$ can be regarded as the approximate solution of the original problem. In fact, from Lemma 10.2.2, we know that when $c^{(-)}(x(\sigma)) = 0$, $x(\sigma)$ is just the solution of problem (10.1.1)–(10.1.3).

The basic idea of the penalty function method is that the penalty parameter σ is increased in each iteration until $\|c^{(-)}(x(\sigma))\|$ is smaller than a given tolerance. Below we give a penalty function method with the simple penalty function.

Algorithm 10.2.3 *(Simple Penalty Function Method)*

Step 1. Given $x_1 \in R^n, \sigma_1 > 0, \epsilon \geq 0, k := 1$.

Step 2. Find a solution $x(\sigma_k)$ of

$$\min_{x \in R^n} P_{\sigma_k}(x), \tag{10.2.15}$$

starting at x_k.

Step 3. If $\|c^{(-)}(x(\sigma_k))\| \leq \epsilon$, stop;
Set $x_{k+1} = x(\sigma_k), \sigma_{k+1} = 10\sigma_k$;
$k := k + 1$, turn to Step 2. □

Note that the parameter $\{\sigma_k\}$ can be chosen flexibly and adoptively. It means that you can choose $\sigma_{k+1} = 10\sigma_k$ or $\sigma_{k+1} = 2\sigma_k$, which depends on the difficulty of minimizing the penalty function at iteration k.

Now we discuss the convergence property of Algorithm 10.2.3.

Theorem 10.2.4 *Suppose that the tolerance ϵ in Algorithm 10.2.3 satisfies*

$$\epsilon > \min_{x \in R^n} \|c^{(-)}(x)\|, \tag{10.2.16}$$

then the algorithm must terminate finitely.

Proof. Suppose, by contradiction, that the theorem is not true. Then there must exist $\sigma_k \to +\infty$ and for all k,

$$\|c^{(-)}(x(\sigma_k))\| > \epsilon. \tag{10.2.17}$$

From (10.2.16), there exists $\hat{x} \in R^n$ such that

$$\|c^{(-)}(\hat{x})\| < \epsilon. \tag{10.2.18}$$

By use of the definition of $x(\sigma)$ and (10.2.4), we have

$$\begin{aligned} f(\hat{x}) + \sigma_k \|c^{(-)}(\hat{x})\|^\alpha &\geq f(x(\sigma_k)) + \sigma_k \|c^{(-)}(x(\sigma_k))\|^\alpha \\ &\geq f(x(\sigma_1)) + \sigma_k \|c^{(-)}(x(\sigma_k))\|^\alpha. \end{aligned} \tag{10.2.19}$$

By arranging (10.2.19) and taking the limit as $\sigma_k \to +\infty$, we obtain that

$$\begin{aligned} & \|c^{(-)}(\hat{x})\|^\alpha - \|c^{(-)}(x(\sigma_k))\|^\alpha \\ \geq \ & \frac{1}{\sigma_k}[f(x(\sigma_1)) - f(\hat{x})] \to 0, \end{aligned} \tag{10.2.20}$$

which contradicts (10.2.17)–(10.2.18). This completes the proof. □

Theorem 10.2.5 *If Algorithm 10.2.3 does not terminate finitely, then*

$$\min_{x \in R^n} \|c^{(-)}(x)\| \geq \epsilon \tag{10.2.21}$$

and

$$\lim_{k \to \infty} \|c^{(-)}(x(\sigma_k))\| = \min_{x \in R^n} \|c^{(-)}(x)\|, \tag{10.2.22}$$

and any accumulation point x^* of $\{x(\sigma_k)\}$ is the solution of problem

$$\min_{x \in R^n} \ f(x), \tag{10.2.23}$$

$$\text{s.t. } \|c^{(-)}(x)\| = \min_{y \in R^n} \|c^{(-)}(y)\|. \tag{10.2.24}$$

Proof. Suppose that the algorithm does not terminate finitely. It follows from Theorem 10.2.4 that (10.2.21) holds. Since $\sigma_k \to +\infty$ and by (10.2.20), we have that for given $\hat{x} \in R^n$,

$$\liminf_{k\to\infty} \left[\|c^{(-)}(\hat{x})\|^\alpha - \|c^{(-)}(x(\sigma_k))\|^\alpha\right] \geq 0 \tag{10.2.25}$$

which concludes (10.2.22).

Let x^* be any accumulation point of $\{x(\sigma_k)\}$. By (10.2.22), x^* must be feasible point of (10.3.25). If x^* is not the solution of (10.2.23)-(10.3.25), there exists $\bar{x}$ such that

$$f(\bar{x}) < f(x^*) \tag{10.2.26}$$

and

$$\|c^{(-)}(\bar{x})\| = \min_{y\in R^n} \|c^{(-)}(y)\|. \tag{10.2.27}$$

It follows from Lemma 10.2.1 that $f(x(\sigma_k))$ approach to $f(x^*)$. Then, by (10.2.26), we have that the inequality

$$f(\bar{x}) < f(x(\sigma_k)) \tag{10.2.28}$$

holds for k sufficiently large, which, together with (10.2.27), gives

$$\begin{aligned} f(\bar{x}) + \sigma_k\|c^{(-)}(\bar{x})\| &< f(x(\sigma_k)) + \sigma_k \min_y \|c^{(-)}(y)\| \\ &= f(x(\sigma_k)) + \sigma_k\|c^{(-)}(x(\sigma_k))\| \end{aligned} \tag{10.2.29}$$

for k sufficiently large. This is just

$$P_{\sigma_k}(\bar{x}) < P_{\sigma_k}(x(\sigma_k)). \tag{10.2.30}$$

This contradicts the definition of $x(\sigma_k)$. The contradiction proves our theorem. □

The above two theorems establish a direct consequence.

Corollary 10.2.6 *Let problem (10.1.1)-(10.1.3) have feasible points. Then Algorithm 10.2.3 either finitely terminates at the solution of (10.2.9)-(10.2.10), or any accumulation points of the generated sequence are the solution of the original problem.*

For the Courant penalty function, i.e., $\|\cdot\|_2$ and $\alpha = 2$ in (10.2.1), we have

$$\nabla f(x(\sigma_k)) + 2\sigma_k \sum_{i=1}^{m} c_i^{(-)}(x(\sigma_k))\nabla c_i^{(-)}(x(\sigma_k)) = 0. \tag{10.2.31}$$

Suppose that the infinite sequence $\{x_k\}$ from Algorithm 10.2.3 converges to x^*, we then have

$$\nabla f(x_{k+1}) = \sum_{i=1}^{m} \lambda_i^{(k+1)} \nabla c_i(x_{k+1}), \tag{10.2.32}$$

where

$$\lambda_i^{(k+1)} = -2\sigma_k c_i^{(-)}(x_{k+1}). \tag{10.2.33}$$

Hence the multiplier $\lambda^{(k+1)}$ given in (10.2.33) is an approximation to a Lagrangian multiplier. It is not difficult to see that if $x_k \to x^*, c^{(-)}(x^*) = 0$ and $\nabla c_i(x^*)\,(i \in E \cup I(x^*))$ are linearly independent, then $\lambda_k \to \lambda^*$. Since, in the general case, $\|\lambda^*\|_2 \neq 0$, it follows from (10.2.33) that

$$\frac{1}{\sigma_k} = O(\|c^{(-)}(x_{k+1})\|_2) = O(\|x_{k+1} - x^*\|_2). \tag{10.2.34}$$

On the other hand, by using (10.2.32), $c^{(-)}(x^*) = 0$ and $\|\lambda^{k+1} - \lambda^*\| = O(\|x_{k+1} - x^*\|)$, we can obtain

$$\begin{bmatrix} W^* & -A^* \\ -(A^*)^T & 0 \end{bmatrix} \begin{pmatrix} x_{k+1} - x^* \\ \hat{\lambda}^{(k+1)} - \hat{\lambda}^* \end{pmatrix} = \begin{pmatrix} 0 \\ \hat{c}(x_{k+1}) \end{pmatrix} + O(\|x_{k+1} - x^*\|), \tag{10.2.35}$$

where

$$W^* = \nabla^*_{xx} L(x^*, \lambda^*), \tag{10.2.36}$$

A^* is a matrix consisting of $\nabla c_i(x^*)$ $(i \in E$ or $\lambda_i^* > 0)$, $\hat{\lambda}^*$ is a vector consisting of those components of λ^* $(i \in E$ or $\lambda_i^* > 0)$, and the definitions of $\hat{\lambda}^{k+1}$ and $\hat{c}(x_{k+1})$ are similar to $\hat{\lambda^*}$. By (10.2.33), we have

$$\|\hat{c}(x_{k+1})\| = O\left(\frac{1}{\sigma_k}\right). \tag{10.2.37}$$

If the second sufficient conditions (8.3.35)-(8.3.36) are satisfied, then the matrix

$$\begin{bmatrix} W^* & -A^* \\ -(A^*)^T & 0 \end{bmatrix} \tag{10.2.38}$$

is nonsingular. By use of (10.2.35) and (10.2.37), we have

$$\|x_{k+1} - x^*\| = O\left(\frac{1}{\sigma_k}\right). \tag{10.2.39}$$

Then, the above equality and (10.2.34) implies that the rate of $\|x_{k+1}-x^*\| \to 0$ is the same as that of $\frac{1}{\sigma_k} \to 0$. This phenomenon can be illustrated by the following example.

Example 10.2.7 *Consider the problem*

$$\min_{(x_1,x_2)\in R^2} \quad x_1 + x_2, \tag{10.2.40}$$

$$\text{s.t.} \qquad x_2 - x_1^2 = 0. \tag{10.2.41}$$

Solution. For the Courant penalty function, we have

$$x(\sigma) = \begin{bmatrix} -\frac{1}{2} \\ \frac{1}{4} - \frac{1}{2\sigma} \end{bmatrix} = x^* - \begin{bmatrix} 0 \\ \frac{1}{2} \end{bmatrix} \frac{1}{\sigma}, \tag{10.2.42}$$

where $x^* = (-1/2, 1/4)$ is the unique solution of (10.2.40)-(10.2.41). Thus, the sequence $\{x_k\}$ generated by Algorithm 10.2.3 satisfies

$$x_{k+1} - x^* = \begin{bmatrix} 0 \\ -\frac{1}{2} \end{bmatrix} \frac{1}{\sigma_k}. \tag{10.2.43}$$

Therefore, we need to choose a very large penalty factor σ_k to solve constrained optimization problems. However, this leads to numerical difficulties of ill-conditioning. □

If L_1 or L_∞ penalty functions are employed, in general, Algorithm 10.2.3 terminates finitely at an exact solution of the original problem. In fact, let x^* be a solution of the original constrained problem, and λ^* a corresponding Lagrange multiplier, then x^* is a minimizer of the L_1 exact penalty function if $\sigma_1 > \|\lambda^*\|_\infty$. Unfortunately, it is a nonsmooth optimization problem to minimize a L_1 exact penalty function. The topic about minimization of nonsmooth exact penalty function will be discussed in detail in §10.6.

10.3 Interior Point Penalty Functions

Similar to the penalty functions discussed in the previous sections, the interior point penalty functions are also used to transform a constrained problem

into an unconstrained problem or into a sequence of unconstrained problems. These functions set a barrier against leaving the feasible region, i.e., these functions are characterized by the property of preserving strict constraint feasibility at all times (i.e., the generated sequence always lies in the interior of the feasible region), by using an interior point penalty term which is infinite on the constraint boundaries. Such methods based on an interior point penalty function are referred to as interior point penalty function methods. Note that the interior point penalty function is also said to be a barrier function, and the corresponding techniques are known as barrier function methods.

The interior point function methods are used to deal with the inequality-constrained optimization problem

$$\min_{x\in R^n} \quad f(x) \tag{10.3.1}$$

$$\text{s.t.} \quad c_i(x) \geq 0,\ i = 1, \cdots, m, \tag{10.3.2}$$

where

$$X = \{x \in R^n \mid c_i(x) \geq 0,\ i = 1, 2, \cdots, m\}$$

is a feasible region. The strictly feasible region is defined by

$$\text{int}(X) \triangleq \{x \in R^n \mid c_i(x) > 0 \text{ for all } i\}. \tag{10.3.3}$$

We assume that int$X$ is nonempty.

The interior point penalty function is of a general form

$$P_\sigma(x) = f(x) + \frac{1}{\sigma}\sum_{i=1}^{m} h(c_i(x)), \tag{10.3.4}$$

where $\sigma > 0$ is a barrier parameter, which controls the iteration. If $\{\sigma_k\} \to \infty$ is chosen, the barrier term becomes more and more negligible except close to the boundary. The $h(c_i)$ is a real function defined on $(0, +\infty)$ which satisfies that

$$\lim_{c_i \to 0_+} h(c_i) = +\infty, \tag{10.3.5}$$

which means the value $h(c_i)$ approaches ∞ as x approaches the boundary of int(X), and that

$$h(c_1) \geq h(c_2),\ \forall c_1 < c_2. \tag{10.3.6}$$

Some interior point penalty functions satisfy

$$h(c_i) > 0,\ \forall c_i > 0. \tag{10.3.7}$$

As we have seen, the inverse barrier function (10.1.19) and the logarithmic barrier function (10.3.16) are the two most important special cases of (10.3.4). For the inverse barrier function, (10.3.7) holds.

Let $x(\sigma)$ be the solution of the problem

$$\min_{x \in R^n} P_\sigma(x). \tag{10.3.8}$$

Assume that we solve (10.3.8) with a strict interior point as an initial point. Note that $P_\sigma(x)$ has value ∞ on its boundary, then $x(\sigma)$ must be an interior point.

Similar to Lemma 10.2.1 and Lemma 10.2.2, we can prove the following results.

Lemma 10.3.1 *Let $\sigma_2 > \sigma_1 > 0$, then*

$$f(x(\sigma_2)) \leq f(x(\sigma_1)), \tag{10.3.9}$$

$$\sum_{i=1}^{m} h(c_i(x(\sigma_2))) \geq \sum_{i=1}^{m} h(c_i(x(\sigma_1))). \tag{10.3.10}$$

Lemma 10.3.2 *Set $\delta = \sum_{i=1}^{m} h(c_i(x(\sigma)))$. Then $x(\sigma)$ is a solution of problem*

$$\min_{x \in R^n} \quad f(x) \tag{10.3.11}$$

$$\text{s.t.} \quad \sum_{i=1}^{m} h(c_i(x)) \leq \delta. \tag{10.3.12}$$

When δ is sufficiently large, the problem (10.3.11)-(10.3.12) can be regarded as an approximation to

$$\min_{x \in R^n} \quad f(x) \tag{10.3.13}$$

$$\text{s.t.} \quad \sum_{i=1}^{m} h(c_i(x)) < +\infty. \tag{10.3.14}$$

By the definition of $h(c_i)$, (10.3.14) is equivalent to

$$c_i(x) > 0, \ i = 1, \cdots, m. \tag{10.3.15}$$

The difference between (10.3.15) and (10.3.2) is whether the boundary of the feasible region is feasible points or not. If $\sigma > 0$ is very large and δ in Lemma

10.3.2 is very large, then $x(\sigma)$ is close to the boundary of the feasible region of (10.3.2). Hence, the feasible region of (10.3.12) is also close to that of the original problem. Therefore $x(\sigma)$ is close to the solution of the original problem.

If $\sigma > 0$ is very large but δ bounded, it follows from the definition of (10.3.4)that the interior point penalty function $P_\sigma(x)$ is very close to $f(x)$ near $x(\sigma)$. In this case, $x(\sigma)$ is regarded approximately as a local minimizer of $f(x)$. Based on these analyses, an algorithm can be written as follows. We assume that $h(\cdot)$ satisfies (10.3.7).

Algorithm 10.3.3 *(Algorithm based on interior point penalty function)*

Step 1. Given x_1 satisfying (10.3.15). Let $\sigma_1 > 0$, $\epsilon \geq 0$, $k := 1$.

Step 2. Starting with x_k solve the problem (10.3.8) for $x(\sigma_k)$. Set $x_{k+1} = x(\sigma_k)$.

Step 3. If

$$\frac{1}{\sigma_k} \sum_{i=1}^{m} h(c_i(x_{k+1})) \leq \epsilon, \tag{10.3.16}$$

stop; otherwise, set $\sigma_{k+1} = 10\sigma_k$, $k := k + 1$; go to Step 2. □

For Algorithm 10.3.3, we will establish the following convergence theorem.

Theorem 10.3.4 *Let $f(x)$ be bounded below on the feasible region X. Then Algorithm 10.3.3 will terminate finitely at $\epsilon > 0$, and when it does not terminate finitely, we have that*

$$\lim_{k\to\infty} \frac{1}{\sigma_k} h(c_i(x_{k+1})) = 0 \tag{10.3.17}$$

and

$$\lim_{k\to\infty} f(x_k) = \inf_{x\in int(X)} f(x) \tag{10.3.18}$$

hold, where int(X) is defined by (10.3.3). Furthermore, any accumulation point of $\{x_k\}$ is the solution of problem (10.3.1)-(10.3.2).

Proof. Obviously, we only need to prove that (10.3.17)-(10.3.18) hold when the algorithm does not terminate finitely.

First, for any $\eta > 0$, there exists $x_\eta \in \mathrm{int}(X)$ such that

$$f(x_\eta) < \inf_{x \in \mathrm{int}(X)} f(x) + \frac{\eta}{2}. \tag{10.3.19}$$

Since the algorithm does not terminate finitely, there is $\sigma_k \to +\infty$. Hence there exists $\bar{k}$ such that

$$\sigma_k > \frac{2}{\eta} \sum_{i=1}^{m} h(c_i(x_\eta)), \ \forall k \geq \bar{k}. \tag{10.3.20}$$

Then, by using the definition of x_{k+1}, and (10.3.19)-(10.3.20), we have

$$P_{\sigma_k}(x_{k+1}) = P_{\sigma_k}(x(\sigma_k)) \leq P_{\sigma_k}(x_\eta),$$

that is

$$\begin{aligned} \frac{1}{\sigma} \sum_{i=1}^{m} h(c_i(x_{k+1})) &\leq f(x_\eta) + \frac{1}{\sigma_k} \sum_{i=1}^{m} h(c_i(x_\eta)) - f(x_{k+1}) \\ &\leq \inf_{x \in \mathrm{int}(X)} f(x) + \frac{1}{2}\eta + \frac{1}{2}\eta - f(x_{k+1}) \\ &\leq \eta \end{aligned} \tag{10.3.21}$$

holds for all $k \geq \bar{k}$. Since $\eta > 0$ is arbitrary, it follows from (10.3.21) that (10.3.17) is true.

From the first row in (10.3.21), we also get

$$\begin{aligned} f(x_{k+1}) &\leq f(x_\eta) + \frac{1}{\sigma_k} \sum_{i=1}^{m} h(c_i(x_\eta)) \\ &\leq \inf_{x \in \mathrm{int}(X)} f(x) + \eta \end{aligned} \tag{10.3.22}$$

holds for all $k \geq \bar{k}$. Then (10.3.18) holds. □

Suppose that the sequence $\{x_k\}$ generated by Algorithm 10.3.3 converges to x^*. If x^* is a strict interior point, then it follows from

$$\nabla f(x_{k+1}) + \frac{1}{\sigma_k} \sum_{i=1}^{m} h'(c_i(x_{k+1})) \nabla c_i(x_{k+1}) = 0 \tag{10.3.23}$$

that

$$\|\nabla f(x_{k+1})\| = O\left(\frac{1}{\sigma_k}\right). \tag{10.3.24}$$

If the second-order sufficient condition (i.e., $\nabla^2 f(x^*)$ positive definite) is satisfied, then (10.3.24) is equivalent to

$$\|x_{k+1} - x^*\| = O\left(\frac{1}{\sigma_k}\right). \tag{10.3.25}$$

The above discussion also tells us that the rate of $\|x_{k+1} - x^*\| \to 0$ is, in general, no quicker than $1/\sigma_k$.

Now, let us consider $x_k \to x^*$, where x^* is a boundary point of the feasible region of (10.3.2), i.e., there exists i such that $c_i(x^*) = 0$. Let $\nabla c_i(x^*)(i \in I(x^*))$ be linearly independent. Let λ_i^* denote the Lagrange multiplier at x^*. From (10.3.23), we have

$$\lim_{k\to\infty} -h'(c_i(x_{k+1}))/\sigma_k = \lambda_i^*. \tag{10.3.26}$$

Write

$$\lambda^{(k+1)} = (-h'(c_1(x_{k+1})), \cdots, -h'(c_m(x_{k+1})))^T/\sigma_k.$$

Define A^* as a matrix consisting of $\nabla c_i(x^*)\,(i \in I(x^*))$, $\hat{\lambda}^*$ as a vector consisting of those components λ_i^* of λ^* $(i \in I(x^*))$, and note that the definitions of $\hat{\lambda}^{(k+1)}$ and $\hat{c}(x_{k+1})$ are similar to $\hat{\lambda}^*$. By (10.3.26), we have

$$|\lambda_i^{(k+1)}| = O\left(\frac{1}{\sigma_k}\right),\ \forall i \notin I(x^*). \tag{10.3.27}$$

Since the columns of A^* are linearly independent, the above equality gives

$$\|\hat{\lambda}^{(k+1)} - \hat{\lambda}^*\| = O\left(\|x_{k+1} - x^*\| + \frac{1}{\sigma_k}\right). \tag{10.3.28}$$

Note that by (10.3.23), we obtain

$$W^*(x_{k+1} - x^*) - A^*(\hat{\lambda}^{(k+1)} - \hat{\lambda}^*) = o(\|x_{k+1} - x^*\|) + O\left(\frac{1}{\sigma_k}\right). \tag{10.3.29}$$

Also,

$$-(A^*)^T(x_{k+1} - x^*) = -\hat{c}(x_{k+1}) + o(\|x_{k+1} - x^*\|). \tag{10.3.30}$$

Then the above two equalities give

$$\begin{aligned} &\begin{bmatrix} W^* & -A^* \\ -(A^*)^T & 0 \end{bmatrix} \begin{bmatrix} x_{k+1} - x^* \\ \hat{\lambda}^{(k+1)} - \hat{\lambda}^* \end{bmatrix} \\ = &\begin{bmatrix} 0 \\ -\hat{c}(x_{k+1}) \end{bmatrix} + o(\|x_{k+1} - x^*\|) + O\left(\frac{1}{\sigma_k}\right). \end{aligned} \tag{10.3.31}$$

Suppose that $\lambda_i^* > 0\,(i \in I(x^*))$. Then for an inverse barrier function and a logarithmic barrier function, by using (10.3.26), we obtain respectively

$$c_i(x_{k+1}) = O\left(\frac{1}{\sqrt{\sigma_k}}\right),\ i \in I(x^*) \tag{10.3.32}$$

and

$$c_i(x_{k+1}) = O\left(\frac{1}{\sigma_k}\right),\ i \in I(x^*). \tag{10.3.33}$$

Hence, provided that the second-order sufficient condition is satisfied, for the inverse barrier function and logarithmic barrier function, we have

$$\|x_{k+1} - x^*\| = O\left(\frac{1}{\sqrt{\sigma_k}}\right) \tag{10.3.34}$$

and

$$\|x_{k+1} - x^*\| = O\left(\frac{1}{\sigma_k}\right). \tag{10.3.35}$$

From (10.3.34)-(10.3.35), it is easy to see that the logarithmic barrier function converges more quickly than the inverse barrier function does.

Now we consider solving (10.3.8) inexactly by interior point function methods. Suppose that both $f(x)$ and $h(c_i(x))$ are convex functions of x, then $P_\sigma(x)$ is also convex with respect to x. Given starting point x_k, then for problem

$$\min\ P_{\sigma_k}(x), \tag{10.3.36}$$

the Newton step is

$$d_k = -[\nabla^2 P_{\sigma_k}(x_k)]^{-1} \nabla P_{\sigma_k}(x_k). \tag{10.3.37}$$

To avoid solving the subproblem exactly, the $x_k + d_k$ is regarded as an approximate solution of (10.3.36). For simplicity, we assume

$$h(c_i) = -\log c_i(x). \tag{10.3.38}$$

Set $x_{k+1} = x_k + d_k$. Then we have from (10.3.37) that

$$\nabla^2 P_{\sigma_k}(x_k)(x_{k+1} - x_k) = -\nabla P_{\sigma_k}(x_k). \tag{10.3.39}$$

Note that

$$\begin{aligned}
\nabla_x P_{\sigma_k}(x_k) &= \nabla f(x_k) - \sum_{i=1}^{m} \frac{1}{\sigma} \frac{1}{c_i(x_k)} \nabla c_i(x_k), \\
\nabla_x^2 P_{\sigma_k}(x_k) &= \nabla^2 f(x_k) - \sum_{i=1}^{m} \frac{1}{\sigma_k} \frac{1}{c_i(x_k)} \nabla^2 c_i(x_k) \\
&\quad + \sum_{i=1}^{m} \frac{1}{\sigma_k} \frac{1}{(c_i(x_k))^2} \nabla c - i(x_k) \nabla c_i(x_k)^T.
\end{aligned}$$

Write

$$\lambda_i^{(k)} = \frac{1}{\sigma_k c_i(x_k)}.$$

Then (10.3.39) can be written as

$$\begin{aligned}
&\left(\nabla^2 f(x_k) - \sum_{i=1}^{m} \lambda_i^{(k)} \nabla^2 c_i(x_k) + \sum_{i=1}^{m} \lambda_i^{(k)} \frac{1}{c_i(x_k)} \nabla c_i(x_k) \nabla c_i(x_k)^T \right) \\
&\cdot (x_{k+1} - x_k) = -\left[\nabla f(x_k) - \sum_{i=1}^{m} \lambda_i^{(k)} \nabla c_i(x_k) \right].
\end{aligned} \tag{10.3.40}$$

If the x_{k+1} defined above lies in the interior of the feasible region, it is regarded as next iterate. Otherwise, there is $\bar{\alpha} > 0$ such that the point $x_k + \bar{\alpha}_k d_k$ lies on the boundary of the feasible region. In such a case, we set

$$x_{k+1} = x_k + 0.9\bar{\alpha}_k d_k. \tag{10.3.41}$$

So, the x_{k+1} still is an interior point. Hence, an inexact interior point penalty function algorithm for subproblem (10.3.36) can be written as follows.

Algorithm 10.3.5 *(Inexact Log-Barrier Function Method)*

Step 1. Given x_1 satisfying (10.3.15), $\sigma_1 > 0, \epsilon \geq 0, k := 1$.

Step 2. Compute $\lambda_i^{(k)} = \frac{1}{\sigma_k c_i(x_k)}$, $i = 1, \cdots, m$;

$$d_k = -[\nabla^2 f(x_k) - \sum_{i=1}^{m} \lambda_i^{(k)} \nabla^2 c_i(x_k)$$

$$+\sum_{i=1}^{m}\lambda_i^{(k)}\frac{1}{c_i(x_k)}\nabla c_i(x_k)\nabla c_i(x_k)^T]^{-1}$$
$$\cdot(\nabla f(x_k)-\sum_{i=1}^{m}\lambda_i^{(k)}\nabla c_i(x_k)); \qquad (10.3.42)$$

If $d_k \neq 0$ go to Step 3;
If $\|\nabla f(x_k)\| \leq \epsilon$, stop;
Otherwise, $\sigma_k := 10\sigma_k$; go to step 2.

Step 3. *Set $\alpha_k = 1$;*
If $x_k + d_k$ is an interior point, go to Step 4;
Find $1 \geq \bar{\alpha}_k > 0$ such that $x_k + \bar{\alpha}_k d_k$ is on the boundary of the feasible region;
Set $\alpha_k := 0.9\bar{\alpha}_k$.

Step 4. *Set $x_{k+1} := x_k + \alpha_k d_k$;*
If

$$\frac{1}{\sigma_k}\sum_{i=1}^{m}\log\left(\frac{1}{c_i(x_k)}\right) \leq \epsilon, \qquad (10.3.43)$$

stop; $\sigma_{k+1} := 10\sigma_k$; $k := k+1$; go to Step 2. □

10.4 Augmented Lagrangian Method

In this section we discuss the augmented Lagrangian method (or the method of multiplier penalty function).

We know from §10.1 that this method is an extension of the quadratic penalty function method. It reduces the possibility of ill-conditioning of the subproblem by introducing Lagrange multiplier estimates. In fact, it is a combination of the Lagrangian function and the quadratic penalty function.

For the case where only equality constraints are presented ($m = m_e$), we rewrite the augmented Lagrangian function as

$$P(x,\lambda,\sigma) = f(x) - \sum_{i=1}^{m}\lambda_i c_i(x) + \frac{1}{2}\sum_{i=1}^{m}\sigma_i(c_i(x))^2. \qquad (10.4.1)$$

When we differentiate with respect to x, we obtain

$$\nabla_x P(x,\lambda,\sigma) = \nabla f(x) - \sum_{i=1}^{m}(\lambda_i - \sigma_i c_i(x))\nabla c_i(x), \qquad (10.4.2)$$

which suggests the formula

$$\lambda_i^{(k+1)} = \lambda_i^{(k)} - \sigma_i^{(k)} c_i(x_{k+1}). \tag{10.4.3}$$

Now we consider the general problem (10.1.1)–(10.1.3) by the augmented Lagrangian function. We rewrite the augmented Lagrangian function $P(x, \lambda, \sigma)$ (10.1.21) as follows:

$$\begin{aligned} P(x, \lambda, \sigma) &= f(x) + \sum_{i=1}^{m_e} \left[-\lambda_i c_i(x) + \frac{1}{2}\sigma_i c_i^2(x) \right] \\ &+ \sum_{i=m_e+1}^{m} \begin{cases} \left[-\lambda_i c_i(x) + \frac{1}{2}\sigma_i c_i^2(x) \right], & \text{if } c_i(x) < \frac{\lambda_i}{\sigma_i}; \\ -\frac{1}{2}\lambda_i^2/\sigma_i, & \text{otherwise} \end{cases} \end{aligned} \tag{10.4.4}$$

where λ_i $(i = 1, \cdots, m)$ are Lagrange multipliers, σ_i $(i = 1, \cdots, m)$ are penalty parameters.

Consider the k-th iteration, using $\lambda_i^{(k)}$ and $\sigma_i^{(k)}$ to denote corresponding components of λ and σ respectively at the k-th iteration. Let x_{k+1} be the solution of the subproblem

$$\min_{x \in R^n} P(x, \lambda^{(k)}, \sigma^{(k)}). \tag{10.4.5}$$

Then we have

$$\begin{aligned} \nabla f(x_{k+1}) &= \sum_{i=1}^{m_e} [\lambda_i^{(k)} - \sigma_i^{(k)} c_i(x_{k+1})] \nabla c_i(x_{k+1}) \\ &+ \sum_{i=m_e+1}^{m} \max\{\lambda_i^{(k)} - \sigma_i^{(k)} c_i(x_{k+1}), 0\} \nabla c_i(x_{k+1}). \end{aligned} \tag{10.4.6}$$

Hence we take

$$\lambda_i^{(k+1)} = \lambda_i^{(k)} - \sigma_i^{(k)} c_i(x_{k+1}),\ i = 1, \cdots, m_e; \tag{10.4.7}$$

$$\lambda_i^{(k+1)} = \max\{\lambda_i^{(k)} - \sigma_i^{(k)} c_i(x_{k+1}), 0\},\ i = m_e + 1, \cdots, m, \tag{10.4.8}$$

as next Lagrange multipliers. By (10.4.6)–(10.4.8), we have that

$$\nabla f(x_{k+1}) - \sum_{i=1}^{m} \lambda_i^{(k+1)} \nabla c_i(x_{k+1}) = 0, \tag{10.4.9}$$

which indicates that for any $k \geq 2$, the error of the KKT condition for (x_k, λ_k) is

$$\|\nabla_x L(x_k, \lambda^{(k)})\| + \|c^{(-)}(x_k)\| = \|c^{(-)}(x_k)\|, \tag{10.4.10}$$

where $L(x, \lambda)$ is the Lagrangian function defined in (8.2.18), $c^{(-)}(x)$ is a constraint violation function defined in (10.1.5)–(10.1.6). Therefore, for $k \geq 2$, provided that the inequality

$$|c_i^{(-)}(x_{k+1})| \leq \frac{1}{4}|c_i^{(-)}(x_k)| \tag{10.4.11}$$

is not satisfied, we enlarge the penalty parameters, i.e., set

$$\sigma_i^{(k+1)} = 10\sigma_i^{(k)}. \tag{10.4.12}$$

Below, we give an algorithm based on the augmented Lagrangian function.

Algorithm 10.4.1 *(Augmented Lagrangian Method)*

Step 1. Given starting point $x_1 \in R^n, \lambda^{(1)} \in R^m$ *with* $\lambda_i^{(1)} \geq 0$ $(i \in I)$; $\sigma_i^{(1)} > 0$ $(i = 1, \cdots, m)$; $\epsilon \geq 0$, $k := 1$.

Step 2. Find approximate minimizer x_{k+1} *to (10.4.5). If* $\|c^{(-)}(x_{k+1})\|_\infty \leq \epsilon$, *stop.*

Step 3. For $i = 1, \cdots, m$, *set*

$$\sigma_i^{(k+1)} = \begin{cases} \sigma_i^{(k)}, & \text{if (10.4.11) holds;} \\ \max[10\sigma_i^{(k)}, k^2], & \text{otherwise.} \end{cases} \tag{10.4.13}$$

Step 4. Update Lagrange multipliers using (10.4.7)–(10.4.8) to obtain $\lambda^{(k+1)}$, $k := k + 1$, *go to Step 2.* □

A practical implementation of the above algorithm is given in LANCELOT due to Conn, Gould, and Toint [68].

Now we establish the finite termination of Algorithm 10.4.1.

Theorem 10.4.2 *Let the feasible region X of problem (10.1.1)–(10.1.3) be nonempty. Then for some $\epsilon > 0$, Algorithm 10.4.1 is either finitely terminated, or the sequence $\{x_k\}$ produced by Algorithm 10.4.1 satisfies*

$$\liminf_{k \to \infty} f(x_k) = -\infty. \tag{10.4.14}$$

Proof. Suppose, on the contrary, that the theorem is not true, that is, for some $\epsilon > 0$, Algorithm 10.4.1 does not terminate finitely, and $\{f(x_k)\}$ is bounded below. Define the set J by

$$J = \left\{ i \mid \lim_{k\to\infty} |c_i^{(-)}(x_k)| = 0,\ 1 \le i \le m \right\}. \tag{10.4.15}$$

Since the algorithm does not terminate finitely, then the set

$$\hat{J} = \{1, 2, \cdots, m\}/J \tag{10.4.16}$$

is not empty. Thus, by the construction of the algorithm, for any $i \in \hat{J}$, we have

$$\lim_{k\to\infty} \sigma_i^{(k)} = +\infty. \tag{10.4.17}$$

Define

$$\mu_i^{(k)} = \lambda_i^{(k)} / \sqrt{\sigma_i^{(k)}}. \tag{10.4.18}$$

It is not difficult to prove that

$$\begin{aligned} \|\mu^{(k+1)}\|_2^2 &\le \sum_{i=1}^{m} [\lambda_i^{(k+1)}]^2 / \sigma_i^{(k)} \\ &\le \|\mu^{(k)}\|_2^2 + 2[P(x_{k+1}, \lambda^{(k)}, \sigma^{(k)}) - f(x_{k+1})] \\ &\quad -2[P(\bar{x}, \lambda^{(k)}, \sigma^{(k)}) - f(\bar{x})] \\ &\le \|\mu^{(k)}\|_2^2 + 2[f(\bar{x}) - f(x_{k+1})], \end{aligned} \tag{10.4.19}$$

where $\bar{x}$ is any feasible point of (10.1.1)–(10.1.3). Since $\{f(x_k)\}$ is bounded below, then (10.4.19) suggests that there exists $\delta > 0$ such that

$$\|\mu^{(k)}\|_2^2 \le \delta k \tag{10.4.20}$$

holds for all k. Set

$$\tilde{J} = \{ i \mid \lim_{k\to\infty} \sigma_i^{(k)} = +\infty \}. \tag{10.4.21}$$

Equation (10.4.17) indicates that $\hat{J} \subseteq \tilde{J}$. By the definition of x_{k+1}, we have

$$\begin{aligned} & f(\bar{x}) + \sum_{i>m_e} \frac{1}{2}\sigma_i^{(k)} \left[\left(c_i(\bar{x}) - \frac{\lambda_i^{(k)}}{\sigma_i^{(k)}} \right)_-^2 - \left(\frac{\lambda_i^{(k)}}{\sigma_i^{(k)}} \right)^2 \right] \\ \ge\ & f(x_{k+1}) + \sum_{i\le m_e} \frac{1}{2}\sigma_i^{(k)} \left[\left(c_i(x_{k+1}) - \frac{\lambda_i^{(k)}}{\sigma_i^{(k)}} \right)^2 - \left(\frac{\lambda_i^{(k)}}{\sigma_i^{(k)}} \right)^2 \right] \\ & + \sum_{i>m_e} \frac{1}{2}\sigma_i^{(k)} \left[\left(c_i(x_{k+1}) - \frac{\lambda_i^{(k)}}{\sigma_i^{(k)}} \right)_-^2 - \left(\frac{\lambda_i^{(k)}}{\sigma_i^{(k)}} \right)^2 \right], \end{aligned} \tag{10.4.22}$$

where $\bar{x}$ is any feasible point of (10.1.1)–(10.1.3), $(\alpha)_-$ denotes $\min\{0, \alpha\}$. By use of (10.4.20)–(10.4.22), we can deduce that

$$\begin{aligned} f(\bar{x}) \quad & -f(x_{k+1}) \geq O(k) \\ & + \sum_{\substack{i \leq me \\ i \in \tilde{J}}} \frac{1}{2}\sigma_i^{(k)} \left[\left(c_i(x_{k+1}) - \frac{\lambda_i^{(k)}}{\sigma_i^{(k)}} \right)^2 - \left(\frac{\lambda_i^{(k)}}{\sigma_i^{(k)}} \right)^2 \right] \\ & + \sum_{\substack{i > me \\ i \in \tilde{J}}} \frac{1}{2}\sigma_i^{(k)} \left[\left(c_i(x_{k+1}) - \frac{\lambda_i^{(k)}}{\sigma_i^{(k)}} \right)_-^2 - \left(\frac{\lambda_i^{(k)}}{\sigma_i^{(k)}} \right)^2 \right]. \quad (10.4.23) \end{aligned}$$

Since the algorithm does not terminate finitely, then for any k, there exists $\bar{k} > k$ such that for some $i \in \tilde{J}$, we have $\sigma_i^{(\bar{k}+1)} > \sigma_i^{(\bar{k})}$ and

$$|c_i(x_{\bar{k}+1})| > \epsilon \text{ if } i \leq m_e$$

or

$$|(c_i(x_{\bar{k}+1}))_-| > \epsilon \text{ if } i > m_e.$$

Then it follows from (10.4.23) that

$$\begin{aligned} f(\bar{x}) - f(x_{\bar{k}+1}) \quad & \geq \quad O(\bar{k}) + \frac{1}{2}\sigma_i^{(\bar{k})}\epsilon^2 + o(\sigma_i^{(\bar{k})}) \\ & \geq \quad O(\bar{k}) + \frac{1}{4}\bar{k}^2\epsilon \quad (10.4.24) \\ & \geq \quad \frac{1}{8}\bar{k}^2\epsilon. \quad (10.4.25) \end{aligned}$$

This contradicts the fact that the $\{f(x_k)\}$ is bounded below. The contradiction proves the theorem. □

Theorem 10.4.3 *Let the feasible region X of problem (10.1.1)–(10.1.3) be nonempty. Then for $\epsilon = 0$, any accumulation point x^* of the sequence $\{x_k\}$ generated by Algorithm 10.4.1 is feasible. Further, if $\{\lambda^{(k)}\}$ is bounded, then x^* is the solution of the original problem (10.1.1)–(10.1.3).*

Proof. By Algorithm 10.4.1 and Theorem 10.4.2, we have

$$\lim_{k \to \infty} \|c^-(x_k)\| = 0. \quad (10.4.26)$$

Hence, any accumulation point of $\{x_k\}$ is a feasible point of (10.1.1)–(10.1.3).

Suppose that $\{\lambda^{(k)}\}$ is bounded for all k. Then by (10.4.22) and (10.4.26), we can deduce that

$$\begin{aligned} f(\bar{x}) &\geq f(x_{k+1}) + \sum_{i\leq m_e} \frac{1}{2}\sigma_i^{(k)} c_i^2(x_{k+1}) \\ &\quad + \sum_{i>m_e} \frac{1}{2}\sigma_i^{(k)} \left[\left(c(x_{k+1}) - \frac{\lambda_i^{(k)}}{\sigma_i^{(k)}} \right)_-^2 - \left(-\frac{\lambda_i^{(k)}}{\sigma_i^{(k)}} \right)_-^2 \right] + o(1) \\ &\geq f(x_{k+1}) + o(1). \end{aligned} \tag{10.4.27}$$

Since $\bar{x} \in X$ is arbitrary, we have

$$\lim_{k\to\infty} f(x) = \inf_{x\in X} f(x). \tag{10.4.28}$$

Thus, any accumulation point x^* of $\{x_k\}$ is the solution of the original problem. □

Finally, we consider the convergence rate of Algorithm 10.4.1. For simplicity, we consider the case with equality constraints only.

Suppose that $x_k \to x^*$. By Theorem 10.4.3, x^* is the solution of (10.1.1)–(10.1.3). Let $A(x^*)$ have full column rank with $\text{rank}(A(x^*)) = m$. Let $A(x_k)$ have full column rank for all k too. Then the $\lambda^{(k+1)}$ generated by Algorithm 10.4.1 is equivalent to the $\lambda(x_{k+1})$ defined by the following expression

$$A(x_{k+1})\lambda(x_{k+1}) = g(x_{k+1}), \tag{10.4.29}$$

where $g(x) = \nabla f(x)$. Note that

$$\lambda(x) = [A(x)]^+ g(x). \tag{10.4.30}$$

It is not difficult to get

$$\nabla\lambda(x) = [A(x)]^+ W(x), \tag{10.4.31}$$

where

$$W(x) = \nabla^2 f(x) - \sum_{i=1}^{m} [\lambda(x)]_i \nabla^2 c_i(x). \tag{10.4.32}$$

By (10.4.7), we have

$$\lambda(x_{k+1}) + D_k c(x_{k+1}) = \lambda(x_k), \tag{10.4.33}$$

where

$$D_k = \begin{bmatrix} \sigma_1^{(k)} & & 0 \\ & \ddots & \\ 0 & & \sigma_m^{(k)} \end{bmatrix}. \tag{10.4.34}$$

Then, it suggests by differentiation that

$$[D_k A(x^*)^T + A(x^*)^+ W(x^*)](x_{k+1} - x^*) \approx A(x^*)^+ W(x^*)(x_k - x^*). \tag{10.4.35}$$

The above expression says that, unless $\sigma^{(k)} \to +\infty$, the sequence $\{x_k\}$ generated by Algorithm 10.4.1 is, in general, convergent linearly.

A shortcoming of the augmented Lagrangian function is that it is only once continuously differentiable. Hence it is possible that there will be some numerical difficulties in solving the subproblem (10.4.5).

10.5 Smooth Exact Penalty Functions

For the equality-constrained problem

$$\min_{x \in R^n} \quad f(x), \tag{10.5.1}$$

$$\text{s.t.} \quad c(x) = 0, \tag{10.5.2}$$

Fletcher [126] first presented a smooth exact penalty function

$$P(x, \sigma) = f(x) - \lambda(x)^T c(x) + \frac{1}{2} c(x)^T D c(x), \tag{10.5.3}$$

where $\lambda(x)$ is given by (10.1.23), $D = \text{diag}(\sigma_1, \cdots, \sigma_m)$. From the discussions in §10.1 and §10.4, we know that, if the second-order sufficient condition holds and σ_i are sufficiently large, then the local minimizer of (10.5.1)–(10.5.2) is a strict local minimizer of the penalty function (10.5.3). Conversely, if $\bar{x}$ is a minimizer of (10.5.3) and $c(\bar{x}) = 0$, then $\bar{x}$ is also the minimizer of problem (10.5.1)–(10.5.2).

If we set all σ_i equal in (10.5.3), then a simple form of Fletcher's smooth exact penalty function

$$P(x, \sigma) = f(x) - \lambda(x)^T c(x) + \frac{1}{2} \sigma \|c(x)\|_2^2 \tag{10.5.4}$$

is obtained. For this penalty function, let $x(\sigma)$ be a solution of the subproblem

$$\min_{x \in R^n} P(x, \sigma). \tag{10.5.5}$$

We have, similar to (10.2.5), that

$$\|c(x(\sigma_2))\|_2 \leq \|c(x(\sigma_1))\|_2, \ \forall \sigma_2 \geq \sigma_1 > 0. \tag{10.5.6}$$

Comparing with the simple penalty function, by use of (10.5.5) and (10.5.4), we need not require $\sigma \to +\infty$. Thus it is possible to attempt the solution of original problem (10.5.1)–(10.5.2) by solving (10.5.5) without needing $\sigma \to +\infty$. In addition, the exact penalty function (10.5.3) is smooth, and thus the convergence rate of methods to solve the unconstrained optimization problem (10.5.5) is rapid. A drawback of this approach is, however, that computing $\nabla_x P(x, \sigma)$ needs computation of $\nabla\lambda(x)$, and further $\nabla^2 f(x)$ and $\nabla^2 c_i(x), \ (i = 1, \cdots, m_e)$. It is expensive.

If, in (10.5.3), we replace D by

$$2\sigma A^+ (A^+)^T, \tag{10.5.7}$$

where $A = \nabla c(x)$, we obtain

$$P(x) = f(x) - \pi(x)^T c(x), \tag{10.5.8}$$

where

$$\pi(x) = A^+(g(x) - \sigma(A^+)^T c(x)). \tag{10.5.9}$$

It is not difficult to find that $\pi(x)$ is the Lagrange multiplier of the problem

$$\min_{d \in R^n} \quad \frac{1}{2}\sigma d^T d + g(x)^T d \tag{10.5.10}$$

$$\text{s.t} \quad A(x)^T d + c(x) = 0. \tag{10.5.11}$$

For general inequality constrained optimization problem (10.1.1)–(10.1.3), we can define $\pi(x)$ as the Lagrange multiplier of subproblem

$$\min_{d \in R^n} \quad g(x)^T d + \frac{1}{2}\sigma \|d\|_2^2, \tag{10.5.12}$$

$$\text{s.t.} \quad c_i(x) + d^T \nabla c_i(x) = 0, \ i \in E, \tag{10.5.13}$$

$$c_i(x) + d^T \nabla c_i(x) \geq 0, \ i \in I, \tag{10.5.14}$$

and then construct the penalty function

$$P(x) = f(x) - \pi(x)^T c(x). \tag{10.5.15}$$

The multiplier $\pi(x)$ can also be obtained by solve the dual problem of (10.5.12)-(10.5.14)

$$\min_{\substack{\pi_i \geq 0 \\ i \in I}} \frac{1}{2} \left\| g(x) - \sum_{i=1}^{m} \pi_i \nabla c_i(x) \right\|_2^2 + \sigma \pi^T c(x). \tag{10.5.16}$$

As an alternative to the subproblem (10.5.4), we may consider the smooth exact penalty function

$$\begin{aligned} P(x, \lambda) &= f(x) - c(x)^T \lambda + \frac{1}{2} \sigma \|c(x)\|_2^2 \\ &\quad + \frac{1}{2} \rho \|M(x)[g(x) - A(x)\lambda]\|_2^2 \end{aligned} \tag{10.5.17}$$

to deal with an equality constrained problem, where $M(x)$ can be $A(x)^T$, $A(x)^+$, or an identity matrix. Equation (10.5.17) may be extended to handle an inequality-constrained problem. We refer the reader to Di Pillo, Grippo and Lampariell [104] or Fletcher [132].

10.6 Nonsmooth Exact Penalty Functions

Let $h(c)$ be a convex function defined on R^m with $h(0) = 0$. If there exists a positive constant $\delta > 0$ such that

$$h(c) \geq \delta \|c\|_1 \tag{10.6.1}$$

holds for all $c \in R^m$, then $h(c)$ is called a strong distance function.

For any strong distance function $h(c)$, we say that the penalty function

$$P_{\sigma,h}(x) = f(x) + \sigma h(c^{(-)}(x)) \tag{10.6.2}$$

is a nonsmooth exact penalty function, where $\sigma > 0$ is a penalty parameter and $c^{(-)}(x)$ is a constraint violation function defined in (10.1.5)–(10.1.6).

For nonsmooth exact penalty function (10.6.2), we give the following theorem about necessity.

Theorem 10.6.1 *Let x^* be a local minimizer of constrained optimization problem (10.1.1)–(10.1.3) satisfying, together with the corresponding Lagrange multiplier vector λ^*, the second-order sufficient condition*

$$d^T \nabla_{xx}^2 L(x^*, \lambda^*) d > 0, \ \forall 0 \neq d \in LFD\ (x^*, X). \tag{10.6.3}$$

Then, if

$$\sigma\delta > \|\lambda^*\|_\infty, \tag{10.6.4}$$

the vector x^* is a strict local minimizer of penalty function $P_{\sigma,h}(x)$ defined in (10.6.2).

Proof. Let (10.6.4) hold. Suppose, to the contrary, that the theorem is not true. Then there exist x_k $(k = 1, 2, \cdots)$ such that $x_k \neq x^*, x_k \to x^*$ and

$$P_{\sigma,h}(x_k) \leq P(x^*), \ \forall k. \tag{10.6.5}$$

The above expression gives

$$f(x_k) + \sigma\delta\|c^{(-)}(x_k)\|_1 \leq f(x^*). \tag{10.6.6}$$

Without loss of generality, we assume that

$$(x_k - x^*)/\|x_k - x^*\| \to d. \tag{10.6.7}$$

By (10.6.6) and the definition of Lagrange multiplier, we obtain

$$\begin{aligned}
& (\sigma\delta - \|\lambda^*\|_\infty)\|c^{(-)}(x_k)\|_1 \\
= \ & (g(x^*) - A(x^*)\lambda^*)^T(x_k - x^*) + (\sigma\delta - \|\lambda^*\|_\infty)\|c^{(-)}(x_k)\|_1 \\
= \ & f(x_k) - \sum_{i=1}^{m} \lambda_i^* c_i(x_k) - f(x^*) - \frac{1}{2}(x_k - x^*)^T\nabla_{xx}^2 L(x^*, \lambda^*)(x_k - x^*) \\
& +(\sigma\delta - \|\lambda^*\|_\infty)\|c^{(-)}(x_k)\|_1 + o(\|x_k - x^*\|^2) \\
= \ & f(x_k) + \sigma\delta\|c^{(-)}(x_k)\|_1 - f(x^*) - \sum_{i=1}^{m}(\|\lambda^*\|_\infty|c_i^{(-)}(x_k)| + \lambda_i^* c_i(x_k)) \\
& -\frac{1}{2}d^T\nabla_{xx}^2 L(x^*, \lambda^*)d\|x_k - x^*\|_2^2 + o(\|x_k - x^*\|_2^2) \\
\leq \ & -\frac{1}{2}d^T\nabla_{xx}^2 L(x^*, \lambda^*)d\|x_k - x^*\|_2^2 + o(\|x_k - x^*\|_2^2).
\end{aligned} \tag{10.6.8}$$

By using (10.6.8) and (10.6.4), and taking the limit, we yield

$$\lim_{k\to\infty} \frac{\|c^{(-)}(x_k)\|_1}{\|x_k - x^*\|} = 0, \tag{10.6.9}$$

which indicates

$$d \in \mathrm{LFD}(x^*, X). \tag{10.6.10}$$

From the second-order sufficient condition we have

$$d^T \nabla^2_{xx} L(x^*, \lambda^*) d > 0 \tag{10.6.11}$$

which shows that the last row in inequality (10.6.8) is negative when k is sufficiently large. Then it produces a contradiction. The contradiction proves the theorem. □

The common nonsmooth exact penalty functions are the L_1 exact penalty function

$$P_1(x) = f(x) + \sigma \|c^{(-)}(x)\|_1 \tag{10.6.12}$$

and the L_∞ exact penalty function

$$P_\infty(x) = f(x) + \sigma \|c^{(-)}(x)\|_\infty. \tag{10.6.13}$$

For nonsmooth exact penalty function (10.6.2), let $x(\sigma)$ be a minimizer of the problem

$$\min_{x \in R^n} P_{\sigma,h}(x). \tag{10.6.14}$$

Completely similar to Lemma 10.2.1 and Lemma 10.2.2, we have the following lemmas. The proofs are omitted.

Lemma 10.6.2 *Let $\sigma_2 > \sigma_1 > 0$. Then we have*

$$f(x(\sigma_2)) \geq f(x(\sigma_1)), \tag{10.6.15}$$

$$h(c^{(-)}(x(\sigma_2))) \leq h(c^{(-)}(x(\sigma_1))). \tag{10.6.16}$$

Lemma 10.6.3 *Let $\eta = h(c^{(-)}(x(\sigma)))$, then $x(\sigma)$ is also the solution of cconstrained problem*

$$\min_{x \in R^n} \quad f(x) \tag{10.6.17}$$

$$\text{s.t.} \quad h(c^{(-)}(x)) \leq \eta. \tag{10.6.18}$$

It is advantageous for an exact penalty function that it is possible to attempt exactly the solution of a constrained optimization problem by solving only a single or finitely many unconstrained problems.

The nonsmooth exact penalty methods can be written in the following form:

Algorithm 10.6.4 *(Nonsmooth Exact Penalty Method)*

Step 1. *Given* $x_1 \in R^n, \sigma_1 > 0, k := 1$.

Step 2. *Solve*

$$\min_{x \in R^n} P_{\sigma,h}(x) \tag{10.6.19}$$

at x_k *to obtain* $x(\sigma)$.

Step 3. *If* $c^{(-)}(x(\sigma_k)) = 0$, *stop;*
$x_{k+1} := x(\sigma_k),\ \sigma_{k+1} := 10\sigma_k;$
$k := k + 1$; *go to Step 2.* □

Note that since $P_{\sigma,h}(x)$ is an exact penalty function, we can obtain the exact solution of the original problem provided that σ is sufficiently large. The following is the convergence result of Algorithm 10.6.4.

Theorem 10.6.5 *Let* $f(x)$, $c_i(x)\,(i = 1, \cdots, m)$ *be twice continuously differentiable. Let the feasible region of constrained optimization problem (10.1.1)–(10.1.3) be nonempty. If second order sufficient condition (10.6.3) holds, then either Algorithm 10.6.4 terminates at a strict local minimizer of problem (10.1.1)–(10.1.3) in finitely many iterations, or the generated sequence satisfies* $\|x_k\| \to \infty$.

Proof. (1) If the algorithm terminates finitely at $x(\sigma)$, then $x(\sigma_k)$ must be a local minimizer of $P_{\sigma_k,h}(x)$. By Lemma 10.6.6, which will be presented below, $x(\sigma_k)$ is also a local minimizer of the original problem (10.1.1)–(10.1.3). Since the second-order sufficient condition is satisfied at $x(\sigma_k)$, then $x(\sigma_k)$ is a strict local minimizer.

(2) Now we prove the second conclusion by contradiction. Suppose that the theorem is not true. Then, for any k, we have $c^{(-)}(x_k) \neq 0$, $\{\|x_k\|\}$ has a bounded subsequence and $\sigma_k \to \infty$. Let $\bar{x}$ be any local minimizer of problem (10.1.1)–(10.1.3). By the definition of $x(\sigma_k)$, we have

$$f(x_{k+1}) + \sigma_k h(c^{(-)}(x_{k+1})) \leq f(\bar{x}) + \sigma_k h(c^{(-)}(\bar{x})) = f(\bar{x}), \tag{10.6.20}$$

which means

$$\sigma_k h(c^{(-)}(x_{k+1})) \leq f(\bar{x}) - f(x_{k+1}).$$

Then we have

$$\lim_{k \to \infty} h(c^{(-)}(x_{k+1})) = 0. \tag{10.6.21}$$

The above expression and (10.6.1) suggest that

$$\lim_{k\to\infty} \|c^{(-)}(x_k)\| = 0. \tag{10.6.22}$$

Since $\{\|x_k\|\}$ has a bounded subsequence, we may let $\hat{x}$ be an accumulation point of $\{x_k\}$ and therefore using (10.6.22) we have

$$c^{(-)}(\hat{x}) = 0. \tag{10.6.23}$$

Let $x_{k_j} \to \hat{x}$. If $\hat{x}$ is not a local minimizer of (10.1.1)–(10.1.3), then there exists $\tilde{x}$ sufficiently approaching $\hat{x}$ and we have

$$f(\tilde{x}) < f(\hat{x}), \tag{10.6.24}$$
$$c^{(-)}(\tilde{x}) = 0. \tag{10.6.25}$$

From (10.6.24) and that $x_{k_j} \to \hat{x}$, we obtain

$$f(x_{k_j}) > f(\tilde{x}) \tag{10.6.26}$$

for j sufficiently large. Hence, we deduce

$$P_{\sigma_{k_j-1},h}(x_{k_j}) > P_{\sigma_{k_j-1},h}(\tilde{x}), \tag{10.6.27}$$

which contradicts the definition of $\{x_{k_j}\}$. Therefore, $\hat{x}$ is a local minimizer of the original problem (10.1.1)–(10.1.3).

Then it follows from Theorem 10.6.1 that there exist $\bar{\delta}$ and $\bar{\sigma}$ such that

$$P_{\bar{\sigma},h}(x) > P_{\bar{\sigma},h}(\hat{x}),\ \forall \|x - \hat{x}\| \le \delta,\ x \ne \hat{x}. \tag{10.6.28}$$

The above expression suggests that

$$P_{\sigma,h}(x) > P_{\sigma,h}(\hat{x}),\ \forall x \ne \hat{x},\ \|x - \hat{x}\| \le \delta,\ \sigma > \bar{\sigma}. \tag{10.6.29}$$

Since $\sigma_{k_j} \to \infty, x_{k_j} \to \hat{x}$ and $x_{k_j} \ne \hat{x}$, then there exists j such that $\|x_{k_j} - \hat{x}\| < \delta$ and $\sigma_{k_j-1} > \bar{\sigma}$. Hence

$$P_{\sigma_{k_j-1},h}(x_{k_j}) > P_{\sigma_{k_j-1},h}(\hat{x}), \tag{10.6.30}$$

which contradicts the definition of x_{k_j}. The contradiction shows the theorem. □

If Algorithm 10.6.4 terminates finitely, it is sure that it terminates at a local minimizer of the original problem. This is based on the following lemma.

Lemma 10.6.6 *For any $\sigma > 0$ and $\bar{x} \in R^n$, if $h(c^{(-)}(\bar{x})) = 0$ and $\bar{x}$ is a local minimizer of the nonsmooth exact penalty function $P_{\sigma,h}(x)$, then $\bar{x}$ is also a local minimizer of the constrained optimization problem (10.1.1)–(10.1.3).*

Proof. Let $\bar{x}$ satisfy $h(c^{(-)}(\bar{x})) = 0$ and be a local minimizer of $P_{\sigma,h}(x)$. Suppose, to the contrary, that the lemma is not true. Then there exist x_k, $(k = 1, 2, \cdots)$, such that $x_k \to \bar{x}, x_k \neq \bar{x}$ and

$$f(x_k) < f(\bar{x}), \tag{10.6.31}$$

$$c^{(-)}(x_k) = 0. \tag{10.6.32}$$

Then, we have

$$P_{\sigma,h}(x_k) < P_{\sigma}(\bar{x}), \tag{10.6.33}$$

which contradicts the fact that $\bar{x}$ is a local minimizer of $P_{\sigma,h}(x)$. Then we complete the proof. □

We would like to mention that, in a rare case, it is possible that there is $\|x_k\| \to \infty$ for Algorithm 10.6.4. For example, consider

$$\min_{x \in R^1} \quad 100e^{-x} - \frac{1}{x^2+1} \tag{10.6.34}$$

$$\text{s.t.} \qquad xe^{-x} = 0. \tag{10.6.35}$$

Taking $h(c) = |c|$ yields that the penalty function is

$$P_{\sigma}(x) = 100e^{-x} - \frac{1}{x^2+1} + \sigma \left| xe^{-x} \right|. \tag{10.6.36}$$

For a sufficiently large $\sigma > 0$, the minimizer $x(\sigma)$ of $P_{\sigma}(x)$ satisfies the equation

$$-100 + \frac{2xe^x}{(x^2+1)^2} = \sigma(x-1), \tag{10.6.37}$$

and $x(\sigma) > 1$. Then

$$\lim_{\sigma \to \infty} \frac{2e^{x(\sigma)}}{(x(\sigma)^2+1)^2 \sigma} = 1. \tag{10.6.38}$$

Therefore

$$\lim_{\sigma \to \infty} x(\sigma) = +\infty, \tag{10.6.39}$$

which says that the sequence generated by Algorithm 10.6.4 satisfies $x_k \to +\infty$.

We note that when the gradients of the constraint function are linearly dependent, it is possible that the minimizer of the original problem (10.1.1)–(10.1.3) is not a stationary point of the exact penalty function (10.6.2). For example,

$$\min_{x\in R^1} \quad x \tag{10.6.40}$$

$$\text{s.t.} \quad c(x) = x^2 = 0. \tag{10.6.41}$$

Taking $h(c) = c$ yields that, for any given $\sigma > 0$, the solution $x^* = 0$ of problem (10.6.40)–(10.6.41) is not the stationary point of the exact penalty function

$$P_{\sigma,h}(x) = x + \sigma x^2. \tag{10.6.42}$$

However, if the gradients of the constraint function are linearly independent, the minimizer of original problem (10.1.1)–(10.1.3) is also the minimizer of the exact penalty function.

Theorem 10.6.7 *Let x^* be a local minimizer of constrained optimization problem (10.1.1)–(10.1.3) and λ^* be a corresponding Lagrange multiplier. If*

$$\nabla c_i(x^*),\ i \in E \cup I(x^*) \tag{10.6.43}$$

are linearly independent, then when (10.6.4) holds, the x^* is also a local minimizer of the penalty function (10.6.2).

Proof. If the theorem is not true, then there exist x_k $(k = 1, 2, \cdots)$ such that $x_k \neq x^*, x_k \to x^*$ and

$$P_{\sigma,h}(x_k) < P(x^*),\ \forall k. \tag{10.6.44}$$

Then by (10.6.1) we have

$$f(x_k) + \sigma\delta\|c^{(-)}(x_k)\|_1 < f(x^*). \tag{10.6.45}$$

Similar to the proof of Theorem 10.6.1, we may assume that (10.6.7) holds and use (10.6.8) to get

$$d \in \text{LFD}\,(x^*, X). \tag{10.6.46}$$

The second-order necessary condition gives

$$d^T \nabla^2_{xx} L(x^*, \lambda^*) d \geq 0, \tag{10.6.47}$$

from which together with (10.6.8), we can deduce that

$$d^T \nabla^2_{xx} L(x^*, \lambda^*) d = 0. \tag{10.6.48}$$

Then

$$\|c^{(-)}(x_k)\|_1 = o(\|x_k - x^*\|^2). \tag{10.6.49}$$

Since x^* is a local minimizer of the original problem (10.1.1)–(10.1.3), it follows from (10.6.45) that

$$\|c^{(-)}(x_k)\|_1 > 0 \tag{10.6.50}$$

for all sufficiently large k. Since the gradients of all active constraints are linearly independent, then there is some y_k such that

$$c^{(-)}(y_k) = 0 \tag{10.6.51}$$

and

$$\|y_k - x_k\| = O(\|c^{(-)}(x_k)\|). \tag{10.6.52}$$

By use of the optimality of x^* and (10.6.51), we have

$$f(y_k) \geq f(x^*). \tag{10.6.53}$$

On the other hand, by the KKT condition, (10.6.4) and (10.6.45), we can obtain that

$$\begin{aligned} f(y_k) &= f(x_k) + \nabla f(x^*)^T (y_k - x_k) + o(\|y_k - x_k\|) \\ &= f(x_k) + \sum_{i=1}^{m} \lambda_i^* (y_k - x_k)^T \nabla c_i(x^*) + o(\|y_k - x_k\|) \\ &\leq f(x_k) + \lambda^*\|_\infty \|c^{(-)}(x_k)\|_1 + o(\|c^{(-)}(x_k)\|_1) \\ &< f(x_k) + \sigma\delta\|c^{(-)}(x_k)\|_1 \\ &< f(x^*), \end{aligned} \tag{10.6.54}$$

which contradicts (10.6.53). The contradiction proves the theorem. □

It is not difficult to see that the equivalence between the nonsmooth exact penalty function (10.6.2) and the constrained optimization problem is based on (10.6.4). In fact, if the inequality (10.6.4) is not satisfied, then the local minimizer of (10.1.1)–(10.1.3) is not necessarily a stationary point of the penalty function (10.6.2).

Theorem 10.6.8 *Let x^* be a local minimizer of constrained optimization (10.1.1)–(10.1.3) and $\nabla f(x^*) \neq 0$. Write*

$$T = \max_{v \in \partial h(0)} \|v\|. \tag{10.6.55}$$

Then, when

$$\sigma \|\nabla c^{(-)}(x^*)\| < \|\nabla f(x^*)\|/T, \tag{10.6.56}$$

x^* is not the stationary point of the penalty function (10.6.2).

Proof. Since the subgradient of the penalty function (10.6.2) at x^* is

$$\partial P_{\sigma,h}(x^*) = \nabla f(x^*) + \sigma \nabla c^{(-)}(x^*)^T \partial h(0), \tag{10.6.57}$$

then, by (10.6.56), we have

$$0 \notin \partial P_{\sigma,h}(x^*). \tag{10.6.58}$$

Therefore, x^* is not a stationary point of $P_{\sigma,h}(x)$. □

Exercises

1. Use the Courant penalty function method to solve the problem

$$\begin{aligned} \min \quad & -2x_1 + x_2 \\ \text{s.t.} \quad & x_2 - x_1^2 = 0. \end{aligned}$$

2. Apply the inverse penalty function method to solve the problem

$$\begin{aligned} \min \quad & -x_1^2 - x_2^2 \\ \text{s.t.} \quad & x_1 \leq 8, \\ & x_2 \leq 8, \\ & x_1 + x_2 \geq 1 \end{aligned}$$

with the initial point $(2 \quad 2)^T$.

3. Apply the logarithmic barrier function method to solve the problem

$$\begin{aligned} \min \quad & x_1 - x_2 + x_2^2 \\ \text{s.t.} \quad & x_1 \geq 0, \\ & x_2 \geq 0 \end{aligned}$$

with the initial point $(1,1)^T$.

4. Apply the Augmented Lagrangian function method to solve the problem in the previous exercise, using initial multipliers $\lambda_1 = 1$ and $\lambda_2 = 1$. Compare the performances of the two methods (Logarithmic barrier function method and the Augmented Lagrangian function method).

5. Let $x(\sigma)$ be the solution of

$$\min P(x,\sigma) = f(x) + \frac{1}{\sigma}\sum_{i=1}^{m}\frac{1}{c_i(x)} \tag{10.6.59}$$

in the interior region $\{x|c_i(x) > 0, i = 1, ..., m\}$, where $\sigma > 0$ is a parameter. Prove that, as σ increases,

(1) $P(x(\sigma),\sigma)$ is non-increasing;

(2) $\sum_{i=1}^{m}\frac{1}{c_i(x(\sigma))}$ is non-decreasing;

(3) $f(x(\sigma)$ is non-increasing.

6. Discuss the penalty function (10.1.10) when $h(c) = e^c$.

7. Using the approximation

$$\max\{c_1, ..., c_m\} \approx \log\left(\sum_{i=1}^{m} e^{c_i}\right),$$

we can replace the L_∞ penalty function by

$$P_e(x) = f(x) + \sigma\log\left(\sum_{i=1}^{m} e^{|c_i^-(x)|}\right).$$

Study the properties of the above penalty function $P_e(x)$.

8. Introducing the slack variables for the inequality constraints, we can reformulate (8.1.1)–(8.1.3) as

$$\begin{aligned}
\min \quad & f(x) \\
\text{s.t.} \quad & c_i(x) = 0, \quad i = 1, \cdots, m_e, \\
& c_i(x) - y_i = 0, \quad i = m_e + 1, ..., m, \\
& y_i \geq 0, \quad i = m_e + 1, ..., m.
\end{aligned}$$

Compare the augmented Lagrange function for the reformulated problem and (10.4.4).

9. Prove (10.5.6).

10. Prove Lemma 10.6.2.

Chapter 11

Feasible Direction Methods

11.1 Feasible Point Methods

A feasible point method requires that all iterate points x_k generated are feasible points of the constraints. For general constrained optimization problems (8.1.1)-(8.1.3), given a current iterate $x_k \in X$, if we can find a descent direction d which is also a feasible direction at x_k, namely

$$d^T \nabla f(x_k) < 0, \tag{11.1.1}$$

$$d \in \text{FD } (x_k, X), \tag{11.1.2}$$

there must exist new feasible points in the form of $x_k + \alpha d$ with the property that $f(x_k + \alpha d) < f(x_k)$. Here FD (x_k, X) is defined by Definition 8.2.1. A direction d satisfying (11.1.1)-(11.1.2) is called a feasible descent direction at x_k.

Let $c_1 \in (0, 1)$ be a given positive constant, x_k be any point in the feasible set X, and d be a vector that satisfies (11.1.1)-(11.1.2). We call α a feasible point Armijo step along direction d at point x_k if $\alpha > 0$ satisfies

$$f(x_k + \alpha d) \leq f(x_k) + \alpha c_1 d^T \nabla f(x_k), \tag{11.1.3}$$

and

$$f(x_k + 2\alpha d) > f(x_k) + 2\alpha c_1 d^T \nabla f(x_k) \tag{11.1.4}$$

holds when $x_k + 2\alpha d \in X$.

Lemma 11.1.1 *Assume that $x_k \in X$ and d satisfies (11.1.1)-(11.1.2). Let α be a feasible point Armijo step along direction d at point x_k, then*

$$f(x_k + \alpha d) \leq f(x_k) - \frac{c_1(1-c_1)}{M}\left[\frac{d^T \nabla f(x_k)}{\|d\|_2}\right]^2 \tag{11.1.5}$$

if $x_k + 2\alpha d \in X$, and

$$f(x_k + \alpha d) \leq f(x_k) + c_1 \frac{\Gamma(x_k)}{2\|d\|_2} d^T \nabla f(x_k), \tag{11.1.6}$$

if $x_k + 2\alpha d \notin X$, where $M = \max_{0\leq t\leq 2} \|\nabla^2 f(x_k + td)\|_2$ and $\Gamma(\bar{x})$ is the distance from $\bar{x}$ to the set of all infeasible points, namely

$$\Gamma(\bar{x}) = \inf_{y \notin X} \|\bar{x} - y\|. \tag{11.1.7}$$

Proof. First assume that $x_k + 2\alpha d \in X$. It follows from (11.1.4) and Taylor expansion that

$$\begin{aligned} 2\alpha c_1 d^T \nabla f(x_k) &< 2\alpha d^T \nabla f(x_k) + \frac{1}{2}(2\alpha d)^T \nabla^2 f(x_k + \eta_k 2\alpha d)(2\alpha d) \\ &\leq 2\alpha d^T \nabla f(x_k) + 2\alpha^2 M \|d\|_2^2, \end{aligned} \tag{11.1.8}$$

where $\eta_k \in (0,1)$. From the above inequality we can obtain that

$$\alpha > -\frac{(1-c_1)}{M\|d\|_2^2} d^T \nabla f(x_k). \tag{11.1.9}$$

Inequality (11.1.15) follows from the above relation and (11.1.3).

Now we consider the case when $x_k + 2\alpha d \notin X$. It follows from the definition of $\Gamma(x)$ that

$$2\alpha\|d\|_2 > \Gamma(x_k). \tag{11.1.10}$$

Thus, $\alpha > \frac{\Gamma(x_k)}{2\|d\|_2}$. This inequality and condition (11.1.3) imply (11.1.6). □

The algorithm given below is a simple algorithm for calculating a feasible point Armijo step. It tries to find an acceptable step by repeatedly doubling or halving the step.

Algorithm 11.1.2

Step 1. *Given $x \in X$, $d \in DF(x, X)$ and $d^T \nabla f(x) < 0$;*
given $c_1 \in (0, 1)$;
let $\alpha_{\max} = +\infty$, $\alpha = 1$.

Step 2. *if*
$$f(x + \alpha d) > f(x) + c_1 \alpha d^T \nabla f(x),$$
or $x + \alpha d \notin X$ go to Step 3;
if $\alpha_{\max} < +\infty$ then stop;
$\alpha := 2\alpha$; Go to Step 2.

Step 3. *$\alpha_{\max} := \alpha$; $\alpha := \alpha/2$; Go to Step 2.* □

It is easy to see that Algorithm 11.1.2 terminates after finitely many iterations with a feasible point Armijo step unless $x + 2^k d \in X$ for all k and $f(x + 2^k d) \to -\infty$. Instead of simply doubling or halving the trial step, we can also use quadratic or cubic interpolations in the above algorithm to accelerate the convergence speed.

For any $x \in X$ and $d \in \text{FD}\ (x, X)$, we call the step $\alpha^* > 0$ that satisfies

$$\alpha^* : \min_{\substack{\alpha > 0 \\ x + \alpha d \in X}} f(x + \alpha d) \tag{11.1.11}$$

a feasible point exact line search step.

Lemma 11.1.3 *Assume that $x \in X$, $d \in FD\,(x, X)$, and α^* satisfies (11.1.11). It follows that*

$$f(x) - f(x + \alpha^* d) \geq \frac{1}{2M} \left[\frac{d^T \nabla f(x)}{\|d\|_2} \right]^2, \tag{11.1.12}$$

or

$$f(x) - f(x + \alpha^* d) \geq -\frac{\Gamma(x)}{2\|d\|_2} d^T \nabla f(x), \tag{11.1.13}$$

where $M = \max_{t \geq 0} \|\nabla^2 f(x + td)\|_2$ and where $\Gamma(x)$ is defined by (11.1.7).

Proof. From Taylor expansion, it follows that

$$f(x + \alpha d) \leq f(x) + \alpha d^T \nabla f(x) + \frac{M}{2} \|d\|_2^2 \alpha^2 \triangleq \phi(\alpha). \tag{11.1.14}$$

Let $\alpha_0 = -d^T \nabla f(x)/(M\|d\|_2^2)$. If $x + \alpha_0 d \in X$, we have that

$$\begin{aligned} f(x+\alpha^* d) &\leq f(x+\alpha_0 d) \leq \phi(\alpha_0) \\ &= f(x) - \frac{1}{2M}\left[\frac{d^T \nabla f(x)}{\|d\|_2}\right]^2. \end{aligned} \tag{11.1.15}$$

If $x + \alpha_0 d \notin X$, it follows that $\alpha_0 \geq \Gamma(x)/\|d\|$. From the convexity of $\phi(\alpha)$, we can show that

$$\begin{aligned} f(x) - f(x+\alpha^* d) &\geq \sup_{0<\alpha<\Gamma(x)/\|d\|_2} [f(x) - f(x+\alpha d)] \\ &\geq \sup_{0<\alpha<\Gamma(x)/\|d\|_2} [f(x) - \phi(\alpha)] \\ &= f(x) - \phi[\Gamma(x)/\|d\|_2] \\ &\geq \frac{\Gamma(x)}{\|d\|_2 \alpha_0}[f(x) - \phi(\alpha_0)] \\ &= \frac{-\Gamma(x)}{2\|d\|_2} d^T \nabla f(x). \end{aligned} \tag{11.1.16}$$

The above two inequalities indicate that the lemma is true. □

Having the technique of searching along a feasible direction in the feasible region, we can solve a constrained optimization problem iteratively as long as we can find a feasible descent direction in every iteration. However, it is not always possible to find a feasible descent direction. For example, for the constraint

$$c(x,y) = y - x^2 = 0, \quad \begin{pmatrix} x \\ y \end{pmatrix} \in \Re^2, \tag{11.1.17}$$

$\mathrm{FD}((x,y), X) = \Phi$ at every feasible point. Therefore no feasible descent direction exists at any feasible point. Fortunately, when the feasible set X is convex, at any point $x \in X$ there exists a feasible descent direction provided that x is not a KKT point. We write this result in the form of a lemma as follows.

Lemma 11.1.4 *Assume that $x \in X$, X is a convex set and $f(x)$ is a convex function. Then there exists a feasible descent direction at x if and only if x is not a minimizer of problem (8.1.1)-(8.1.3).*

Proof. It is obvious that there exists no feasible descent direction at x if x is a minimizer.

Now assume that x is not a minimizer, then there exists an $\hat{x} \in X$ such that

$$f(\hat{x}) < f(x). \tag{11.1.18}$$

Because $f(x)$ is a convex function, it follows from (11.1.18) that

$$d^T \nabla f(x) < 0,$$

where $d = \hat{x} - x$. Because of the convexity of X, $d \in \text{FD}\ (x, X)$. Therefore d is a feasible descent direction. □

A general algorithm that uses feasible descent directions is given as follows.

Algorithm 11.1.5

Step 1. Given initial point $x_1 \in X$, $k := 1$;

Step 2. If no vector d satisfies (11.1.1)-(11.1.2) then stop; find d_k that satisfies (11.1.1)-(11.1.2);

Step 3. Carry out a certain feasible point search, obtaining $\alpha_k > 0$.

Step 4. $x_{k+1} = x_k + \alpha_k d_k$; $k := k + 1$; Go to Step 2.

We can use a feasible point exact line search or a feasible point Armijo search to obtain α_k in Step 3 of the above algorithm.

From example (11.1.17), even if Algorithm 11.1.5 terminates, it may not stop at a stationary point. However, when the objective function $f(x)$ is convex and when the feasible set is convex, x_k must be the optimal solution if Algorithm 11.1.5 terminates at iteration k.

An important issue is the choice of d_k that satisfies (11.1.1)-(11.1.2). Consider the very special case when $X = \Re^n$. Let $f(x)$ be a uniformly convex function defined on $\Re^n$. Assume that d_k is the search direction at the k-th iteration satisfying

$$d_k^T \nabla f(x_k) < 0. \tag{11.1.19}$$

Let θ_k be the angle between d_k and the steepest descent direction $-\nabla f(x_k)$, namely

$$\cos\theta_k = -\frac{d_k^T \nabla f(x_k)}{\|d_k\|_2 \|\nabla f(x_k)\|}. \tag{11.1.20}$$

Lemma 11.1.6 *For an unconstrained optimization problem, assume that the objective function is twice continuously differentiable and uniformly convex and that a line search algorithm with* $x_{k+1} = x_k + \alpha_k d_k$ *and* $\|\nabla f(x_k)\| \neq 0$ *for all* k*, satisfies*

$$\sum_{k=1}^{\infty} \cos^2 \theta_k < +\infty, \tag{11.1.21}$$

where $\cos\theta_k$ *is defined by (11.1.20). Then*

$$\lim_{k\to\infty} \inf \|\nabla f(x_k)\| > 0. \tag{11.1.22}$$

Proof. Because $f(x)$ is uniformly convex, there exists x^* such that

$$f(x^*) = \min_{x\in\Re^n} f(x). \tag{11.1.23}$$

It is obvious that (11.1.22) is equivalent to

$$\lim_{k\to\infty} f(x_k) > f(x^*). \tag{11.1.24}$$

Define $X_1 = \{x | f(x) \leq f(x_1)\}$ and

$$m_1 = \min_{x\in X_1} \min_{\|d\|_2=1} d^T \nabla^2 f(x) d, \tag{11.1.25}$$

$$M_1 = \max_{x\in X_1} \max_{\|d\|_2=1} d^T \nabla^2 f(x) d. \tag{11.1.26}$$

It can be shown that $0 < m_1 \leq M_1 < +\infty$ because $f(x)$ is uniformly convex. Therefore,

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\leq f(x_k) - \min_{t>0} f(x_k + t d_k) \\ &\leq \frac{1}{2m_1} \|\nabla f(x_k)\|_2^2 \cos^2\theta_k \\ &\leq \frac{\cos^2\theta_k}{2m_1} (M_1 \|x_k - x^*\|_2)^2 \\ &\leq \frac{\cos^2\theta_k}{2} \left(\frac{M_1}{m_1}\right)^2 [f(x_k) - f(x^*)]. \end{aligned} \tag{11.1.27}$$

Consequently, it follows that

$$f(x_{k+1}) - f(x^*) \geq \left(1 - \frac{M_1^2}{2m_1^2} \cos^2\theta_k\right) [f(x_k) - f(x^*)], \tag{11.1.28}$$

for all k. Assumption (11.1.21) implies the existence of k_0 such that

$$\frac{M_1^2}{2m_1^2}\cos^2\theta_k < 1, \quad \forall k \geq k_0. \tag{11.1.29}$$

Because $\|\nabla f(x_{k_0})\| \neq 0$, we have that $f(x_{k_0}) - f(x^*) = \delta > 0$. From (11.1.21) there exists $\eta > 0$ such that

$$\prod_{j=k_0}^{\infty}\left(1 - \frac{M_1^2}{2m_1^2}\cos^2\theta_k\right) \geq \eta > 0. \tag{11.1.30}$$

Thus, it follows from (11.1.28) and (11.1.30) that

$$f(x_k) - f(x^*) \geq \eta\delta > 0$$

for all $k \geq k_0$. This implies that (11.1.24). □

The above lemma tells us that we require

$$\sum_{k=1}^{\infty}\cos^2\theta_k = +\infty, \tag{11.1.31}$$

to ensure the algorithm converging to a stationary point.

Similar to the steepest descent direction, we can define the feasible steepest descent direction as follows.

Definition 11.1.7 *Let $x \in X$; if a vector d in the closure of $FD(x, X)$ solves*

$$\min_{\substack{d\in \mathrm{FD}\ (x,X)\\ d\neq 0}} \frac{d^T\nabla f(x)}{\|d\|_2}, \tag{11.1.32}$$

it is called a feasible steepest descent direction.

Because $\mathrm{FD}(x, X)$ may not be a closed set, the minimum of (11.1.32) can not be reached by any $d \in \mathrm{FD}\ (x, X)$. Thus, a feasible steepest descent direction may not belong to $\mathrm{FD}(x, X)$. Therefore, it is not easy to generate the steepest direction directly to constrained optimization by simply making the steepest descent direction "feasible".

Consider the inequality constrained optimization problem

$$\min f(x), \tag{11.1.33}$$

$$\text{s.t. } c_i(x) \geq 0 \quad i = 1, \cdots, m. \tag{11.1.34}$$

Let $x_k \in X$. It is obvious that $I(x_k) = \{i | c_i(x_k) = 0\}$. In order to find a feasible descent direction at the k-th iteration, we consider the approximate subproblem

$$\min \ \alpha d^T \nabla f(x_k), \tag{11.1.35}$$

$$\text{s.t. } x_k + \alpha d \in X. \tag{11.1.36}$$

As the aim for constructing this subproblem is to find a search direction, we can assume that $\|\alpha d\|$ is very small. When $\|\alpha d\|$ is sufficiently small, (11.1.36) is equivalent to

$$c_j(x_k + \alpha d) \geq 0, \quad j \in I(x_k). \tag{11.1.37}$$

The above inequalities hold if we require that

$$\alpha d^T \nabla c_j(x_k) - \frac{1}{2} M \alpha^2 \|d\|_2^2 \geq 0, \quad j \in I(x_k), \tag{11.1.38}$$

where $M > 0$ is an upper bound for

$$\max_{x \in X} \max_{j \in I(x_k)} \|\nabla^2 c_j(x)\|_2. \tag{11.1.39}$$

Replacing αd by d, we can obtain the following subproblem

$$\min d^T \nabla f(x_k), \tag{11.1.40}$$

$$\text{s.t. } d^T \nabla c_i(x_k) - \frac{M}{2} \|d\|_2^2 \geq 0, \quad i \in I(x_k). \tag{11.1.41}$$

By further replacing d by Md, the above problem can be rewritten as

$$\min d^T \nabla f(x_k), \tag{11.1.42}$$

$$\text{s.t. } d^T \nabla c_i(x_k) - \frac{1}{2} \|d\|_2^2 \geq 0, \quad i \in I(x_k). \tag{11.1.43}$$

The dual problem for the above problem is

$$\max_{\lambda} \min_{d \in \Re^n} \left[d^T \nabla f(x_k) - \sum_{i \in I(x_k)} \lambda_i \left(d^T \nabla c_i(x_k) - \frac{1}{2} \|d\|_2^2 \right) \right] \tag{11.1.44}$$

$$\text{s.t. } \lambda_i \geq 0, \quad i \in I(x_k). \tag{11.1.45}$$

The above problem can be written in the equivalent form

$$\min_\lambda \quad \frac{\left\|\nabla f(x_k) - \sum_{i\in I(x_k)} \lambda_i \nabla c_i(x)\right\|_2^2}{\sum_{i\in I(x_k)} \lambda_i}, \tag{11.1.46}$$

$$\text{s.t.} \quad \lambda_i \geq 0, \quad i \in I(x_k), \qquad \sum_{i\in I(x_k)} \lambda_i > 0. \tag{11.1.47}$$

Define

$$d(\lambda) = -\frac{1}{\sum_{i\in I(x_k)} \lambda_i} \left(\nabla f(x_k) - \sum_{i\in I(x_k)} \lambda_i \nabla c_i(x_k) \right). \tag{11.1.48}$$

The objective function in (11.1.46) can be written as

$$\phi(\lambda) = \sum_{i\in I(x_k)} \lambda_i \|d(\lambda)\|_2^2. \tag{11.1.49}$$

Direct calculations show that

$$\nabla\phi(\lambda) = \begin{bmatrix} 2d(\lambda)^T \nabla c_{k_1}(x_k) - \|d(\lambda)\|_2^2 \\ \vdots \\ 2d(\lambda)^T \nabla c_{k_I}(x_k) - \|d(\lambda)\|_2^2 \end{bmatrix}, \tag{11.1.50}$$

where $\{k_1, k_2, \cdots, k_I\}$ are the elements of $I(x_k)$. Furthermore, we have that

$$\nabla^2\phi(\lambda) = \frac{2}{\sum_{i\in I(x_k)} \lambda_i} T(\lambda)^T T(\lambda), \tag{11.1.51}$$

where

$$T(\lambda) = (d(\lambda) - \nabla c_{k_1}(x_k), d(\lambda) - \nabla c_{k_2}(x_k), \cdots, d(\lambda) - \nabla c_{k_I}(x_k)). \tag{11.1.52}$$

Thus, $\phi(\lambda)$ is a convex function. Let $\lambda^{(k)}$ be a solution of (11.1.46)-(11.1.47), then $d(\lambda^{(k)})$ is a solution of problem (11.1.40)-(11.1.41). In that case, x_k is a KKT point of (11.1.33)-(11.1.34) if $d(\lambda^{(k)}) = 0$, and $d(\lambda^{(k)})$ is a feasible descent direction at x_k satisfying

$$d(\lambda^{(k)})^T \nabla f(x_k) = \frac{\|d(\lambda^{(k)})\|_2^2}{\sum_{i\in I(x_k)} \lambda_i^{(k)}} < 0, \tag{11.1.53}$$

if $d(\lambda^{(k)}) \neq 0$.

It is not difficult to show that subproblem (11.1.40)-(11.1.41) has nonzero minimum if and only if

$$\begin{aligned} d^T \nabla f(x_k) &< 0, && (11.1.54)\\ d^T \nabla c_i(x_k) &> 0, \quad i \in I(x_k) && (11.1.55) \end{aligned}$$

has a solution. That is to say, when (11.1.54)-(11.1.55) has a solution we can obtain a feasible descent direction by solving (11.1.40)-(11.1.41). On the other hand, if (11.1.54)-(11.1.55) has no solution, similar to Lemma 8.2.5 it can be shown that there exist $\lambda_i^* (i \in I(x_k)) \geq 0$ and $\lambda_0^* \geq 0$ such that

$$\lambda_0^* \nabla f(x_k) - \sum_{i \in I(x_k)} \lambda_i^* \nabla c_i(x^*) = 0, \tag{11.1.56}$$

and that $\sum_{i \in I(x_k)} \lambda_i^{*2} + \lambda_0^* \neq 0$. Therefore we know that x_k is a Fritz John point of the original optimization problem (11.1.33)-(11.1.34).

Another subproblem for finding a feasible descent direction is directly based on (11.1.54)-(11.1.55), having the form:

$$\begin{aligned} \min\ & \sigma && (11.1.57)\\ \text{s.t. } d^T \nabla f(x_k) &\leq +\sigma, && (11.1.58)\\ d^T \nabla c_i(x_k) &\geq -\sigma, \quad i \in I(x_k), && (11.1.59)\\ \|d\| &\leq 1. && (11.1.60) \end{aligned}$$

It is easy to see that the minimum of the above subproblem $\sigma^* = 0$ if and only if (11.1.54)-(11.1.55) has no solutions.

11.2 Generalized Elimination

Consider the equality constrained problem

$$\begin{aligned} \min\ & f(x) && (11.2.1)\\ \text{s.t.}\ & c(x) = 0, && (11.2.2) \end{aligned}$$

where $c(x) = (c_1(x), \cdots, c_m(x))^T$. Assume that we have a certain partition on the variable x:

$$x = \begin{bmatrix} x_B \\ x_N \end{bmatrix}, \tag{11.2.3}$$

where $x_B \in \Re^m$, $x_N \in \Re^{n-m}$. Therefore (11.2.2) can be written as

$$c(x_B, x_N) = 0. \tag{11.2.4}$$

Suppose that we can solve x_B from (11.2.4), namely

$$x_B = \phi(x_N), \tag{11.2.5}$$

then (11.2.1)-(11.2.2) is equivalent to

$$\min_{x_N \in \Re^{n-m}} f(x_B, x_N) = f(\phi(x_N), x_N) = \tilde{f}(x_N). \tag{11.2.6}$$

The vector

$$\tilde{g}(x_N) = \nabla x_N \tilde{f}(x_N) \tag{11.2.7}$$

is called the reduced gradient. It is easy to verify that

$$\tilde{g}(x_N) = \frac{\partial}{\partial x_N} f(x_B, x_N) + \frac{\partial x_B^T}{\partial x_N} \frac{\partial}{\partial x_B} f(x_B, x_N). \tag{11.2.8}$$

From (11.2.4) we can see that $\frac{\partial x_B^T}{\partial x_N}$ satisfies that

$$\frac{\partial x_B^T}{\partial x_N} \frac{\partial}{\partial x_B} c(x_B, x_N)^T + \frac{\partial}{\partial x_N} c(x_B, x_N)^T = 0. \tag{11.2.9}$$

If $\frac{\partial c^T}{\partial x_B}$ is nonsingular, the above two equations imply that

$$\begin{aligned} \tilde{g}(x_N) &= \frac{\partial f(x_B, x_N)}{\partial x_N} \\ &\quad - \frac{\partial c(x_B, x_N)^T}{\partial x_N} \left[\frac{\partial c(x_B, x_N)^T}{\partial x_B} \right]^{-1} \frac{\partial f(x_B, x_N)}{\partial x_B}. \end{aligned} \tag{11.2.10}$$

Therefore, the reduced gradient can be expressed as the gradient of the Lagrangian function at the reduced space:

$$\tilde{g}(x_N) = \frac{\partial}{\partial x_N} [f(x) - \lambda^T c(x)], \tag{11.2.11}$$

where λ is a multiplier satisfying

$$\frac{\partial f(x)}{\partial x_B} = \frac{\partial c^T(x)}{\partial x_B} \lambda. \tag{11.2.12}$$

In other words, when the Lagrange multiplier λ is chosen as

$$\left[\frac{\partial c^T(x_B)}{\partial x_B}\right]^{-1} \frac{\partial f(x)}{\partial x_B}, \tag{11.2.13}$$

we have that

$$\nabla_x L(x,\lambda) = \begin{bmatrix} 0 \\ \tilde{g}(x_N) \end{bmatrix}. \tag{11.2.14}$$

Therefore, the reduced gradient can be viewed as the nonzero part of the gradient of the Lagrangian function.

Using the reduced gradients, we can construct line search directions for the unconstrained problem (11.2.6). For example, we can use the steepest descent direction

$$\bar{d}_k = -\tilde{g}((x_N)_k) \tag{11.2.15}$$

or the quasi-Newton direction

$$\bar{d}_k = -B_k^{-1}\tilde{g}((x_N)_k). \tag{11.2.16}$$

Here the subscript k indicates the iterate number, B_k is an approximate Hessian matrix which can be updated from iteration to iteration(for example, by BFGS formula). It is worth pointing out that carrying out a line search

$$\min_{\alpha\geq 0} f(\phi((x_N)_k + \alpha\bar{d}_k), \quad (x_N)_k + \alpha\bar{d}_k) \tag{11.2.17}$$

on the unconstrained problem (11.2.6) is equivalent to carrying out a curve search on the original objective function $f(x)$ along the following curve:

$$c(x_B, (x_N)_k + \alpha\bar{d}_k) = 0. \tag{11.2.18}$$

Because the function $\phi(x)$ is not known explicitly, for every trial α we need to solve (11.2.18) to obtain

$$x_B = \phi((x_N)_k + \alpha\bar{d}_k) \tag{11.2.19}$$

when carrying out line searches (11.2.17). This can be done by an approximate Newton's method, namely

$$x_B^{(0)} = (x_B)_k, \tag{11.2.20}$$

$$x_B^{(i+1)} = x_B^{(i)} - \left[\frac{\partial c(x_k)^T}{\partial x_B}\right]^{-1} c(x_B^{(i)}, (x_N)_k + \alpha\bar{d}_k). \tag{11.2.21}$$

Because Newton's method converges quadratically, usually an acceptable x_B will be obtained after applying (11.2.21) for a few iterations. If $x_B^{(i)}$ does not converge after some iterations, α should be reduced to continue the line search procedure.

The following is a general framework of the variable elimination method.

Algorithm 11.2.1 *(Variable Elimination Method)*

Step 1. Given a feasible point $x_1 \in X$, $\epsilon \geq 0$, $k = 1$;

Step 2. Compute

$$\frac{\partial c(x_k)^T}{\partial x} = \begin{bmatrix} A_B \\ A_N \end{bmatrix}, \tag{11.2.22}$$

where the partition satisfies that A_B is nonsingular. Compute λ by (11.2.12), and $\tilde{g}_k$ by (11.2.11).

Step 3. If $\|\tilde{g}_k\| \leq \epsilon$ then Stop;
Generate a feasible descent direction $\bar{d}_k$ satisfying

$$\bar{d}_k^T \tilde{g}_k < 0. \tag{11.2.23}$$

Step 4. Carry out line search (11.2.17) obtaining $\alpha_k > 0$,
Let $x_{k+1} = (\phi((x_N)_k + \alpha_k \bar{d}_k), (x_N)_k + \alpha_k \bar{d}_k)$,
$k := k + 1$; go to Step 2.

It is very easy to see that the above algorithm is in fact a descent method for the unconstrained optimization problem (11.2.6). The only thing that we should keep in mind is that the partition (x_B, x_N) may differ from iteration to iteration. Using the convergence results of descent methods for unconstrained optimization, we can easily establish the following result.

Theorem 11.2.2 *Assume that $f(x)$ and $c(x)$ are twice continuously differentiable. If $[(\nabla c(x)^T)^T \nabla c(x)^T]^{-1}$ is bounded above uniformly on the feasible set X, Algorithm 11.2.1 with exact line searches and the assumption*

$$\sum \cos^2 \langle \bar{d}_k, \tilde{g}_k \rangle = \infty \tag{11.2.24}$$

ensures that

$$\liminf_{k \to \infty} \|(\nabla f(x_k) - \nabla c(x_k)^T \lambda_k)\| = 0, \tag{11.2.25}$$

or

$$\lim_{k\to\infty} f(x_k) = -\infty, \tag{11.2.26}$$

where $\lambda_k = [\nabla c(x_k)^T]^+ \nabla f(x_k)$.

Let $\bar{d}_k = -\tilde{g}_k$, then (11.2.23) and (11.2.24) hold. In that case, Algorithm 11.2.1 is exactly the steepest descent method in the lower dimensional space using the variable partition.

Consider any nonsingular matrix $S \in \Re^{n\times n}$ and variable transformation:

$$x = Sw. \tag{11.2.27}$$

We partition the variable w:

$$w = \begin{bmatrix} w_B \\ w_N \end{bmatrix}, \tag{11.2.28}$$

where $w_B \in \Re^m$, $w_N \in \Re^{n-m}$. Using the constrained condition

$$c((S)_B w_B + (S)_N w_N) = 0 \tag{11.2.29}$$

to eliminate variable w_B, namely

$$w_B = \bar{\phi}(w_N). \tag{11.2.30}$$

In this way, the optimization problem (11.2.1)-(11.2.2) is equivalent to

$$\min_{w_N \in \Re^{n-m}} f(S_B w_B + S_N w_N) = \bar{f}(w_N). \tag{11.2.31}$$

Provided that $S_B^T \nabla C(x)^T$ is nonsingular, direct calculations give that

$$\nabla_{w_N} \bar{f}(w_N) = \bar{g}(w_N) = S_N^T[\nabla f(x) - \nabla c(x)^T \lambda], \tag{11.2.32}$$

where λ satisfies

$$S_B^T[\nabla f(x) - \nabla c(x)^T \lambda] = 0. \tag{11.2.33}$$

Thus, we have obtained an elimination method based on the variable transformations at every iterations. This method is called the generalized elimination method.

Algorithm 11.2.3 *(General Elimination Method)*

Step 1. *Given a feasible point $x_1 \in X$, $\epsilon \geq 0$, $k = 1$;*

Step 2. *Construct a nonsingular matrix S_k, and a partition $S_k = [(S_k)_B \ , \quad (S_k)_N]$ such that $(S_k)_B^T \nabla c(x_k)^T$ nonsingular; Compute λ by (11.2.33) and $\bar{g}_k$ by (11.2.32).*

Step 3. *If $\|\bar{g}_k\| \leq \epsilon$ then stop;*
Generate a descent direction $\bar{d}_k$ satisfying

$$\bar{d}_k^T \bar{g}_k < 0; \tag{11.2.34}$$

Step 4. *Carry out line search:*

$$\min_{\alpha>0} f((S_k)_B \bar{\phi}((w_k)_N + \alpha \bar{d}_k) + (S_k)_N[(w_k)_N + \alpha \bar{d}_k]) \tag{11.2.35}$$

obtaining $\alpha_k > 0$; let

$$x_{k+1} = (S_k)_B \bar{\phi}((w_k)_N + \alpha_k \bar{d}_k) + (S_k)_N[(w_k)_N + \alpha_k \bar{d}_k]; \tag{11.2.36}$$

$k := k + 1$; go to Step 2. $\square$

In the algorithm, w_k is a vector satisfying $x_k = S_k w_k$. Similar to the elimination method, for each trial step $\alpha > 0$, we need to compute

$$w_B = \bar{\phi}((w_k)_N + \alpha \bar{d}_k), \tag{11.2.37}$$

which can be done by applying the approximate Newton's method to the nonlinear system

$$c((S_k)_B w_B + (S_k)_N[(w_k)_N + \alpha \bar{d}_k]) = 0. \tag{11.2.38}$$

That is,

$$\begin{aligned} w_B^{(i+1)} &= w_B^{(i)} - [(\nabla c(x_k)^T)^T (S_k)_B]^{-1} c((S_k)_B w_B^{(i)} \\ &+ (S_k)_N[(w_k)_N + \alpha \bar{d}_k]), \quad i = 1, 2, \cdots. \end{aligned} \tag{11.2.39}$$

It is not difficult to see that if S_k is the unit matrix in every iteration, the generalized elimination method is exactly the original elimination method.

The variable increment $x_{k+1} - x_k$ of the generalized elimination method in every iteration is actually the sum of two parts:

$$x_{k+1} = x_k + d_k^{(1)} + d_k^{(2)}, \tag{11.2.40}$$

where

$$d_k^{(1)} = \alpha_k (S_k)_N \bar{d}_k, \tag{11.2.41}$$

$$d_k^{(2)} = (S_k)_B [\bar{\phi}((w_k)_N + \alpha_k \bar{d}_k) - (w_k)_B]. \tag{11.2.42}$$

The iteration process first obtains the step $d_k^{(1)}$, then uses the approximate Newton method along the direction $d_k^{(2)}$ to find a point in the feasible set, as shown in Figure 11.2.1.

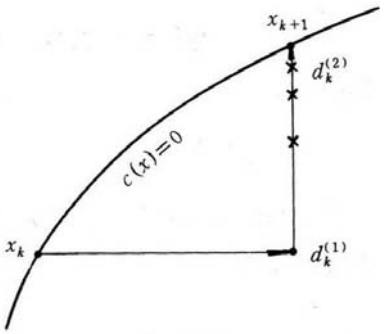


Figure 11.2.1 Iterative procedure of generalized elimination method

Looking at Figure 11.2.1, we see an undesirable property of such a process. The iteration first moves away from the feasible set, and then it comes back, though the essential idea for feasible point methods is to force all iterate points inside the feasible region. Except for very special constraints, it is unavoidable to use the technique of moving away and coming back if we require that all iterate points are feasible. But, how to make the "moving away" as small as possible? An intuitive answer is to choose $d_k^{(1)}$ to be a linearized feasible direction at x_k. It is reasonable to believe that such a $d_k^{(1)}$ would make $x_k + d_k^{(1)}$ closer to the feasible region, consequently the approximate Newton's method will bring $x_k + d_k^{(1)}$ back to the feasible region more quickly. Figure 11.2.2 illustrates the above discussions.

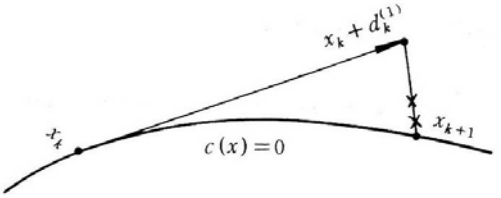


Figure 11.2.2

When $d_k^{(1)}$ is a linearized feasible direction, the method is called a feasible direction method. It is obvious that if

$$(S_k)_N^T \nabla c(x_k)^T = 0 \tag{11.2.43}$$

holds, $d_k^{(1)}$ is a linearized feasible direction. Because of this, feasible direction methods can also be viewed as special generalized elimination methods.

11.3 Generalized Reduced Gradient Method

The generalized reduced gradient method (GRG method) is in fact Algorithm 11.2.1 with $\bar{d}_k = -\tilde{g}_k$. It is the steepest descent method in the reduced space.

In each iteration, the line search can be the Armijo rule, namely reducing the trial step repeatedly until an acceptable one is obtained. The condition for accepting the new point can be the simple reduction

$$f(x_{k+1}) < f(x_k). \tag{11.3.1}$$

In each iteration, we apply (11.2.21) for at most N times to compute x_B, where N is a given positive number. If the approximate Newton's method has not converged after N iterations, we reduce the trial step α and repeat the iteration. Because the quadratic convergence of Newton's method, normally one or two iterations of (11.2.21) will return a sufficiently accurate feasible point x_{k+1}. Therefore in practice we can choose N between 3 to 6.

The algorithm is the generalized reduced gradient method with Armijo line searches requiring simple reductions.

Algorithm 11.3.1 *(Generalized Reduced Gradient Method)*

Step 1. *Given a feasible point $x_1 \in X$, $\epsilon \geq 0$, $\bar{\epsilon} > 0$; positive integer M; $k := 1$.*

Step 2. *Compute*

$$\nabla c(x_k)^T = \begin{bmatrix} A_B \\ A_N \end{bmatrix}, \tag{11.3.2}$$

where the partition satisfies that $A_B \in \Re^{m\times m}$ is nonsingular; Compute λ from (11.2.12) and $\tilde{g}_k$ from (11.2.11).

Step 3. *If $\|\tilde{g}_k\| \leq \epsilon$ then stop;*
let $\bar{d}_k = -\tilde{g}_k$; and $\alpha = \alpha_k^{(0)} > 0$.

Step 4. *$x_N = (x_k)_N + \alpha\bar{d}_k$;*
$x_B = (x_k)_B$; $j := 0$.

Step 5. *$x_B = x_B - A_B^{-T}c(x_B, x_N)$;*
compute $c(x_B, x_N)$;
if $\|c(x_B, x_N)\| \leq \bar{\epsilon}$ then go to Step 7;
$j := j + 1$; if $j < M$ go to Step 5.

Step 6. *$\alpha := \alpha/2$, go to Step 4.*

Step 7. *If $f(x_B, x_N) \geq f(x_k)$ then go to Step 6.*
$x_{k+1} = (x_B, x_N)$, $k := k + 1$; go to Step 2.

This algorithm is in fact a gradient method. Thus the simple reduction (11.3.1) on the objective function can not guarantee convergence. In other words, we can not show that the iterates generated by Algorithm 11.3.1 converges to a KKT point of the original optimization problem (11.2.1)-(11.2.2). There are two ways to overcome this. The first one is to use a better line search condition. For example, we can replace the simple reduction condition (11.3.1) by the Wolfe line search condition

$$\tilde{f}((x_k)_N + \alpha_k\bar{d}_k) \leq \tilde{f}((x_k)_N) + \beta\alpha_k\bar{d}_k^T\tilde{g}_k, \tag{11.3.3}$$

where α is the step length, $\beta \in (0, 1)$ is a positive constant, and $\tilde{f}(x_N)$ is defined by (11.2.6). Condition (11.3.3) can be written as

$$f(x_{k+1}) \leq f(x_k) - \alpha_k\beta\|\tilde{g}_k\|_2^2. \tag{11.3.4}$$

Thus if we replace the condition $f(x_B, x_N) \geq f(x_k)$ for rejecting a new point in Step 7 of Algorithm 11.3.1 by

$$f(x_B, x_N) > f(x_k) - \alpha\beta\|\tilde{g}_k\|_2^2, \tag{11.3.5}$$

the Wolfe line search condition (11.3.3) is satisfied. Another way is to require that the initial trial step length $\alpha_k^{(0)}$ in Step 3 of the algorithm satisfies that

$$\frac{\alpha_k^{(0)}}{\|\tilde{g}_k\|} \to 0, \tag{11.3.6}$$

$$\sum_{k=1}^{\infty} \frac{\alpha_k^{(0)}}{\|\tilde{g}_k\|} = +\infty. \tag{11.3.7}$$

Similar to convergence analyses for unconstrained optimization methods, we can prove the following convergence results.

Theorem 11.3.2 *Assume that $f(x)$, $c(x)$ are twice continuously differentiable, that the matrices A_B^{-1} in Step 2 of Algorithm 11.3.1 are bounded above uniformly, and that $\alpha_k^{(0)}$ in Step 3 satisfies that $(\alpha_k^{(0)})^{-1}$ is uniformly bounded. If the condition $f(x_B, x_N) \geq f(x_k)$ in Step 7 is replaced by (11.3.5), if $\epsilon = 0$ and if Algorithm 11.3.1 does not terminate, it follows that either*

$$\lim_{k\to\infty} \|\tilde{g}_k\| = 0 \tag{11.3.8}$$

or

$$\lim_{k\to\infty} f(x_k) = -\infty. \tag{11.3.9}$$

Theorem 11.3.3 *Assume that $f(x)$, $c(x)$ are twice continuously differentiable, that the matrices A_B^{-1} in Step 2 of Algorithm 11.3.1 are bounded above uniformly, and that $\alpha_k^{(0)}$ in Step 3 satisfies (11.3.6) and (11.3.7). Then if $\epsilon = 0$ and if Algorithm 11.3.1 does not terminate, either (11.3.8) or (11.3.9) holds.*

One advantage of the generalized reduced gradient method is the dimension of the problem is reduced due to variable elimination. The method can also make good use of the special structure of the problem such as sparsity and constant coefficients so that λ and $\tilde{g}$ can be computed quickly. Similarly, if A_B is sparse, sparse linear system solvers can be used when using

the approximate Newton's method to obtain x_B. Therefore, for large-scale nonlinear programming problems there are many linear constraints or with sparse structures, the generalized reduced gradient method is one of the most efficient methods.

Because one or more systems of nonlinear equations have to be solved in the generalized reduced gradient method, its computation cost is very high if the matrix A_B is not sparse and without special structures.

11.4 Projected Gradient Method

From the discussions at the end of Section 11.2, in order to choose $d_k^{(1)}$ as a linearized feasible direction in the generalized elimination method, S_k should satisfy

$$(S_k)_N^T \nabla c(x_k)^T = 0. \tag{11.4.1}$$

Consider the case that the steepest descent direction is used in the generalized elimination method, namely

$$\bar{d}_k = -\bar{g}_k. \tag{11.4.2}$$

From (11.2.41) it follows that

$$d_k^{(1)} = -\alpha_k (S_k)_N (S_k)_N^T \nabla f(x_k). \tag{11.4.3}$$

Obviously, $(S_k)_N (S_k)_N^T$ is a linear projection from $\Re^n$ to the subspace spanned by the columns of $(S_k)_N$. Suppose $A_k = \nabla c(x_k)^T$ is full column rank, the subspace spanned by the columns of $(S_k)_N$ is the null space of A_k^T. Therefore the direction defined by (11.4.3) is actually the projection of the negative gradient of the objective function to the null space of the Jacobi matrix. If S_k satisfies

$$(S_k)_N^T (S_k)_N = I, \tag{11.4.4}$$

$(S_k)_N (S_k)_N^T$ is an orthogonal projector, and

$$\begin{aligned} P_k &= (S_k)_N (S_k)_N^T \\ &= I - A_k (A_k^T A_k)^{-1} A_k^T, \end{aligned} \tag{11.4.5}$$

when A_k has full column rank. In this case, $P_k \nabla f(x_k)$ is an orthogonal projection of $\nabla f(x_k)$ to the null space of A_k^T. Therefore, the generalized elimination method is a projected gradient method. In a practical implementation

of the projected gradient method, we can use the QR factorization of A_k:

$$A_k = [Y_k \ Z_k] \begin{bmatrix} R_k \\ 0 \end{bmatrix}. \tag{11.4.6}$$

It is easy to see that we can let $(S_k)_N = Z_k$. Hence,

$$\bar{g}_k = Z_k^T g_k \tag{11.4.7}$$

is the reduced gradient and

$$d_k = -Z_k \bar{g}_k = -Z_k Z_k^T g_k \tag{11.4.8}$$

is a projection of the negative gradient to the null space of A_k^T, which is a descent direction of $f(x)$. Thus, we can choose α_k such that

$$f(x_k + \alpha_k d_k) < f(x_k). \tag{11.4.9}$$

The point $x_k + \alpha_k d_k$ may be infeasible. A feasible point can be obtained by the approximate Newton's method

$$x_k^{(1)} = x_k + \alpha_k d_k, \tag{11.4.10}$$

$$x_k^{(i+1)} = x_k^{(i)} - Y_k R_k^{-1} c(x_k^{(i)}), \quad i = 1, 2, \ldots. \tag{11.4.11}$$

When $c(x_k^{i+1})$ is sufficiently small, we terminate (11.4.11) and set $x_{k+1} = x_k^{(i+1)}$. The above iteration process is essentially (11.2.39) with $(S_k)_B = Y_k$. If α_k is sufficiently small, we have that

$$\|x_{k+1} - (x_k + \alpha_k d_k)\| = O(\alpha_k^2), \tag{11.4.12}$$

therefore there exists $\alpha_k > 0$ such that

$$f(x_{k+1}) < f(x_k). \tag{11.4.13}$$

The algorithm given below is the projected gradient method with Armijo line searches requiring simple reductions.

Algorithm 11.4.1 *(Projected Gradient Method)*

Step 1. Given a feasible point $x_1 \in X$, $\epsilon \geq 0$, $\bar{\epsilon} > 0$, a positive integer N, $k := 1$.

Step 2. Compute the QR Factorization

$$\nabla C(x_k)^T = [Y_k \ Z_k] \begin{bmatrix} R_k \\ 0 \end{bmatrix};$$

$\bar{g}_k = Z_k^T \nabla f(x_k)$;
if $\|\bar{g}_k\| \leq \epsilon$ *then stop;*
$d_k = -Z_k \bar{g}_k$; *set* $\alpha = \alpha_k^{(0)} > 0$.

Step 3. $y := x_k + \alpha d_k$; $i := 0$.

Step 4. $y := y - Y_k R_k^{-1} c(y)$;
if $\|C(y)\| \leq \bar{\epsilon}$ *and* $f(y) < f(x_k)$ *then go to Step 5;*
$i := i + 1$; *if* $i < N$ *then go to Step 4;*
$\alpha = \alpha/2$; *go to Step 3.*

Step 5. $x_{k+1} := y$, $k := k + 1$; *go to Step 2.*

Similar to the generalized reduced gradient method, Algorithm 11.4.1 needs to modify its line search conditions or to impose certain conditions on the initial steplength in order to guarantee convergence.

For inequality constraints, active set technique can be used to obtain feasible directions. However, one difficulty of the active set technique is the zigzagging phenomenon, which was pointed out by Wolfe [350]. There are many ways to overcome zigzagging in feasible direction methods. The main idea for avoiding zigzagging is not to delete constraints from the active set unless it is absolutely needed.

If the search direction $d_k = -Z_k \bar{g}_k$ in the last line of Step 2 in Algorithm 11.4.1 is replaced by

$$d_k = -Z_k z_k, \tag{11.4.14}$$

where $z_k \in \Re^{n-m}$ is any vector that satisfies

$$z_k^T \bar{g}_k < 0, \tag{11.4.15}$$

then the algorithm is a general form of the linearized feasible direction method, often called feasible direction method.

Based on our definitions, the search directions in a feasible direction method is only a linearized feasible direction instead of a feasible direction. An exception is the case when all the constraints are linear functions. In this

case, the linearized feasible directions are also feasible directions. Because feasible direction methods were first used for linearly constrained problems, when they are generalized to nonlinear constraints they are still called feasible direction methods. To be precise, this method, when applied to nonlinearly constrained problems, should be called linearized feasible direction method instead of feasible direction method.

For nonlinear constraints, normally linearized feasible directions are not feasible directions. Therefore searching along a linearized feasible direction may return an infeasible point. That is why Newton's method or approximate Newton's method should be applied to bring the iterate back to the feasible region before continuing the next search along a search direction and another moving back to the feasible region. This procedure leads to the sawtooth phenomenon, as indicated by Figure 11.4.1

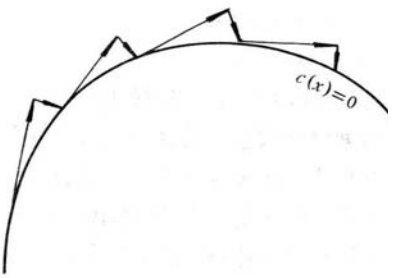


Figure 11.4.1

11.5 Linearly Constrained Problems

Feasible direction methods are very efficient for linearly constrained problems, for example, for equality constrained problem

$$\min_{x\in\Re^n} \quad f(x), \tag{11.5.1}$$

$$\text{s.t.} \quad A^T x = b, \tag{11.5.2}$$

where $b \in \Re^m$, $A \in \Re^{n\times m}$, $rank(A) = m$, and $f(x)$ is a nonlinear function. The search direction of a feasible direction method can be expressed by

$$d_k = Z\bar{d}_k, \tag{11.5.3}$$

where $\bar{d}_k \in \Re^{n-m}$, and $Z \in \Re^{n\times(n-m)}$ satisfying

$$A^T Z = 0, \tag{11.5.4}$$
$$\bar{d}_k^T Z^T \nabla f(x_k) < 0. \tag{11.5.5}$$

Specifically, we can let $\bar{d}_k = -Z^T \nabla f(x_k)$, which leads to the following feasible direction method based on steepest descent direction.

Algorithm 11.5.1

Step 1. *Given a feasible point x_1;*
Compute Z such that $A^T Z = 0$ and $Rank(Z) = n - m$;
$k = 1$, $\epsilon \geq 0$.

Step 2. *$d_k = -ZZ^T \nabla f(x_k)$; if $\|d_k\| \leq \epsilon$ then stop;*
Carry out line search along d_k obtaining $\alpha_k > 0$;
$x_{k+1} = x_k + \alpha_k d_k$; $k := k + 1$; go to Step 2.

The algorithm is actually a steepest descent method in the feasible region. Thus, its convergence can be established under certain line search conditions. When Z satisfies $Z^T Z = I$, Algorithm 11.5.1 is a projected gradient method.

Now we discuss a projected gradient method for general linearly constrained optimization problems, which was proposed by Calamai and Moré [50].

For a general linearly constrained optimization problem

$$\min_{x\in\Re^n} \quad f(x), \tag{11.5.6}$$
$$\text{s.t.} \quad a_i^T x = b_i, \quad i \in E, \tag{11.5.7}$$
$$a_i^T x \geq b_i, \quad i \in I, \tag{11.5.8}$$

the feasible set X is

$$X = \{x | a_i^T x = b_i,\ i \in E; a_i^T x \geq b_i,\ i \in I\}. \tag{11.5.9}$$

Define the mapping P,

$$P(x) = \arg\min\{\|z - x\|,\ z \in X\}, \tag{11.5.10}$$

where arg min indicates any $z \in X$ that minimizes $\|z - x\|$, $\|\cdot\|$ is a norm. For simplicity, we assume that $\|\cdot\|$ is the Euclidean norm $\|\cdot\|_2$.

Consider the steepest descent method. The iterate x_{k+1} should be a point on the straight line

$$\bar{x}_k(\alpha) = x_k - \alpha \nabla f(x_k). \tag{11.5.11}$$

But we need the iterate points on the feasible region, we use the projection P to project the line (11.5.11) to X, obtaining the piecewise line

$$x_k(\alpha) = P[x_k - \alpha \nabla f(x_k)]. \tag{11.5.12}$$

We search along the piecewise line, namely find $\alpha_k > 0$ such that

$$f(x_k(\alpha_k)) \leq f(x_k) + \mu_1 (x_k(\alpha_k) - x_k)^T \nabla f(x_k), \tag{11.5.13}$$

$$\alpha_k \geq \gamma_1 \quad \text{or} \quad \alpha_k \geq \gamma_2 \ \bar{\alpha}_k > 0, \tag{11.5.14}$$

where $\bar{\alpha}_k$ satisfies

$$f(x_k(\bar{\alpha}_k)) > f(x_k) + \mu_2 (x_k(\bar{\alpha}_k) - x_k)^T \nabla f(x_k). \tag{11.5.15}$$

Here, $\gamma_1, \gamma_2, \mu_1, \mu_2$ are positive constants and $\mu_1, \mu_2 \in (0, 1)$.

The method of Calamai and Moré can be stated as follows.

Algorithm 11.5.2

Step 1. *Given a feasible point* x_1, $\mu \in (0,1)$, $\gamma > 0$, $\alpha_0 = 1$, $k := 1$;

Step 2. $\alpha_k := \max\{2\alpha_{k-1}, \gamma\}$.

Step 3. *if (11.5.13) holds go to Step 4;*
$\alpha_k = \alpha_k/4$; *go to Step 3;*

Step 4. $x_{k+1} := x_k(\alpha_k)$; $k := k + 1$; *go to Step 2.*

It is easy to see that α_k computed by Algorithm 11.5.2 satisfies (11.5.13)-(11.5.15) for $\mu_2 = \mu_1 = \mu$, $\gamma_1 = \gamma$, $\gamma_2 = 1/4$.

From the definition of $P(x)$, for any $x \in \Re^n$ it follows that

$$(x - P(x))^T (z - P(x)) \leq 0, \quad \forall z \in X. \tag{11.5.16}$$

Let $x = x_k - \alpha_k \nabla f(x_k)$ and $z = x_k$ in the above relation, then we obtain that

$$(x_k - \alpha_k \nabla f(x_k) - x_{k+1})^T (x_k - x_{k+1}) \leq 0. \tag{11.5.17}$$

Thus, it follows from (11.5.13) and (11.5.17) that

$$f(x_k) - f(x_{k+1}) \geq \mu_1 \frac{\|x_{k+1} - x_k\|_2^2}{\alpha_k}. \tag{11.5.18}$$

First we have the following lemmas.

Lemma 11.5.3 *Assume that $f(x)$ is continuously differentiable and bounded below on the feasible set X. If $\nabla f(x)$ is uniformly continuous on X, the iterates generated by Algorithm 11.5.2 satisfy*

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x_k\|}{\alpha_k} = 0. \tag{11.5.19}$$

Proof. If the lemma is not true, there is an infinite subsequence K_0 such that

$$\frac{\|x_{k+1} - x_k\|}{\alpha_k} \geq \delta, \quad \forall k \in K_0 \tag{11.5.20}$$

where $\delta > 0$ is a positive constant independent of k. It follows from the above relation and (11.5.18) that for all $k \in K_0$ we have that

$$f(x_k) - f(x_{k+1}) \geq \delta\mu_1\|x_{k+1} - x_k\| \geq \delta^2\mu_1\alpha_k. \tag{11.5.21}$$

Because $f(x)$ is bounded below on the feasible set and all x_k are feasible, it follows that

$$\sum_{k=1}^{\infty}[f(x_k) - f(x_{k+1})] < +\infty. \tag{11.5.22}$$

Inequalities (11.5.21) and (11.5.22) imply that

$$\lim_{\substack{k\to\infty \\ k\in K_0}} \|x_{k+1} - x_k\| = \lim_{\substack{k\to\infty \\ k\in K_0}} \alpha_k = 0. \tag{11.5.23}$$

Therefore the first condition of (11.5.14) does not hold for sufficiently large $k \in K_0$, which shows that

$$\alpha_k \geq \gamma_2\bar{\alpha}_k, \tag{11.5.24}$$

and that (11.5.15) holds. Using the monotonically non-increasing property of

$$\Psi(\alpha) = \frac{\|P(x + \alpha d) - x\|}{\alpha}, \quad \alpha > 0 \tag{11.5.25}$$

and relation (11.5.24), we can prove that

$$\frac{\|x_k - x_k(\bar{\alpha}_k)\|}{\bar{\alpha}_k} \geq \min\left\{1, \frac{1}{\gamma_2}\right\} \frac{\|x_k - x_k(\alpha_k)\|}{\alpha_k}. \tag{11.5.26}$$

Thus, letting $x = x_k - \bar{\alpha}_k \nabla f(x_k)$ and $z = x_k$ in (11.5.16) gives that

$$\begin{aligned} -(x_k(\bar{\alpha}_k) - x_k)^T \nabla f(x_k) &\geq \frac{\|x_k - x_k(\bar{\alpha}_k)\|^2}{\bar{\alpha}_k} \\ &\geq \min\left\{1, \frac{1}{\gamma_2}\right\} \delta \|x_k - x_k(\bar{\alpha}_k)\| \end{aligned} \tag{11.5.27}$$

for all sufficiently large $k \in K_0$. The uniform continuity of $\nabla f(x)$ on X implies that

$$f(x_k(\bar{\alpha}_k)) - f(x_k) = (x_k(\bar{\alpha}_k) - x_k)^T \nabla f(x_k) + o(\|x_k(\bar{\alpha}_k) - x_k\|). \tag{11.5.28}$$

It follows from (11.5.15) and (11.5.28) that

$$-(x_k(\bar{\alpha}_k) - x_k)^T \nabla f(x_k) \leq o(\|x_k - x_k(\bar{\alpha}_k)\|). \tag{11.5.29}$$

The above inequality contradicts (11.5.27), which shows that the lemma is true. □

Lemma 11.5.4 *A point $x^* \in X$ is a KKT point of problem (11.5.6)-(11.5.8) if and only if there exists $\bar{\delta} > 0$ such that*

$$P(x^* - \alpha \nabla f(x^*)) = x^* \tag{11.5.30}$$

holds for all $\alpha \in [0, \bar{\delta}]$.

Proof. Equation (11.5.30) is equivalent to

$$\|x^* - \bar{\delta} \nabla f(x^*) - x^*\|_2^2 \leq \|x^* - \bar{\delta} \nabla f(x^*) - x\|_2^2 \tag{11.5.31}$$

holds for all $x \in X$. Because X is a convex set, (11.5.31) is equivalent to

$$(x - x^*) \nabla f(x^*) \geq 0 \tag{11.5.32}$$

holds for all feasible points sufficiently close to x^*. This means that x^* is the minimizer of function $x^T \nabla\ f(x^*)$ on X, which is equivalent to that x^* is a KKT point of problem (11.5.6)-(11.5.8). □

From the above two lemmas, we can easily establish the convergence result of Algorithm 11.5.2.

Theorem 11.5.5 *Assume that $f(x)$ is continuously differentiable on the feasible set X. Then, any accumulation point x^* of $\{x_k\}$ generated by Algorithm 11.5.2 is a KKT point of problem (11.5.6)-(11.5.8).*

Proof. If the theorem is not true, there exist a subsequence of $\{x_k\}$ satisfying

$$\lim_{\substack{k\in K_0\\ k\to\infty}} x_k = x^*, \tag{11.5.33}$$

and

$$P(x^* - \bar{\delta}\nabla f(x^*)) \neq x^*, \tag{11.5.34}$$

where $\bar{\delta} > 0$, K_0 is a subset of $\{1, 2, \cdots\}$. Because of (11.5.33), we can assume that $x_k \in S$ $(k \in K_0)$, and S is a bounded closed set. Because $\nabla f(x)$ is continuous on S, it is also uniformly continuous on S. It follows from Lemma 11.5.3 that

$$\lim_{\substack{k\in K_0\\ k\to\infty}} \frac{\|x_{k+1} - x_k\|}{\alpha_k} = 0. \tag{11.5.35}$$

From the continuity of $\nabla f(x)$ and (11.5.33)-(11.5.34), we can show that

$$\lim_{\substack{k\in k_0\\ k\to\infty}} \frac{\|x_k(\bar{\delta}) - x_k\|}{\bar{\delta}} = \frac{\|P(x^* - \bar{\delta}\nabla f(x^*)) - x^*\|}{\bar{\delta}} > 0. \tag{11.5.36}$$

Because the function $\Psi(\alpha)$ defined by (11.5.25) is monotonically non-increasing, it follows from (11.5.35) and (11.5.36) that $\alpha_k \geq \bar{\delta}$ holds for all sufficiently large $k \in K_0$. Therefore,

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq -\mu_1(\nabla f(x_k))^T(x_k(\alpha_k) - x_k) \\ &\geq -\mu_1(\nabla f(x_k))^T(x_k(\bar{\delta}) - x_k) \\ &\geq \mu_1 \frac{\|x_k(\bar{\delta}) - x_k\|^2}{\bar{\delta}}. \end{aligned} \tag{11.5.37}$$

Now it follows from (11.5.37) and (11.5.36) that

$$\lim_{\substack{k\in K_0\\ k\to\infty}} \inf[f(x_k) - f(x_{k+1})] > 0. \tag{11.5.38}$$

This contradicts the fact that $\lim_{k\to\infty} f(x_k) = f(x^*)$. Therefore the theorem is true. □

Exercises

1. Assume that X is a convex polyhedral defined by $X = \{x \mid \quad Ax \geq b\}$. Show that finding a direction d satisfying (11.1.1)–(11.1.2) is a convex programming problem and give its dual.

2. By direct elimination, find the point on the ellipse defined by the intersection of the surface $x + y = 1$ and $x^2 + 2y^2 + z^2 = 1$ which is nearest to the origin.

3. Apply Newton's method with the generalized elimination to solve the problem

$$\begin{array}{ll} \min & 8x_1^4 - x_2^4 \\ \text{s.t.} & x_1 + x_2 = 1. \end{array}$$

4. For the above problem, at the point $(3, -2)^T$, please give the projected gradient and the projected Hessian. What are the projected gradient and the projected Hessian at the solution?

5. Give the projected gradient algorithm for the box constrained problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & l \leq x \leq u. \end{array}$$

6. Prove Theorem 11.3.2.

7. Assume the symmetric matrix $B \in \Re^{n\times n}$ is invertible and $b \in \Re^n$. Prove that the matrix

$$\hat{B} = \begin{bmatrix} B & b \\ b^T & \beta \end{bmatrix} \tag{11.5.39}$$

is invertible if and only if $\beta - b^T B^{-1} b \neq 0$. And prove that, when $\hat{B}$ is invertible, there exist μ and u such that

$$\hat{B}^{-1} = \begin{bmatrix} B^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \mu u u^T. \tag{11.5.40}$$

Chapter 12

Sequential Quadratic Programming

12.1 Lagrange-Newton Method

Consider the equality constrained optimization problem

$$\min_{x\in\Re^n} \quad f(x) \tag{12.1.1}$$

$$\text{s.t.} \quad c(x) = 0, \tag{12.1.2}$$

where $c(x) = (c_1(x), \cdots, c_m(x))^T \in \Re^n$. The Lagrangian function is

$$L(x, \lambda) = f(x) - \lambda^T c(x). \tag{12.1.3}$$

A point x is a KKT point of (12.1.1)-(12.1.2) if and only if there exists $\lambda \in \Re^m$ such that

$$\nabla_x L(x, \lambda) = \nabla f(x) - \nabla c(x)^T \lambda = 0, \tag{12.1.4}$$

$$\nabla_\lambda L(x, \lambda) = -c(x) = 0. \tag{12.1.5}$$

The nonlinear system (12.1.4)-(12.1.5) requires x to be a stationary point of the Lagrangian function. Therefore any method based on solving (12.1.4)-(12.1.5) can be called a Lagrange method. For a given iterate point $x_k \in \Re^n$ and an approximate Lagrange multiplier $\lambda_k \in \Re^m$, the Newton-Raphson step for solving (12.1.4)-(12.1.5) is $((\delta x)_k, (\delta\lambda)_k)$, which satisfies

$$\begin{pmatrix} W(x_k, \lambda_k) & -A(x_k) \\ -A(x_k)^T & 0 \end{pmatrix} \begin{pmatrix} (\delta x)_k \\ (\delta\lambda)_k \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) - A(x_k)\lambda_k \\ -c(x_k) \end{pmatrix}, \tag{12.1.6}$$

where

$$A(x_k) = \nabla c(x_k)^T, \tag{12.1.7}$$

$$W(x_k, \lambda_k) = \nabla^2 f(x_k) - \sum_{i=1}^{m} (\lambda_k)_i \nabla^2 c_i(x_k). \tag{12.1.8}$$

Consider the penalty function

$$P(x, \lambda) = \|\nabla f(x) - A(x)\lambda\|_2^2 + \|c(x)\|_2^2; \tag{12.1.9}$$

it is easy to show that $(\delta x)_k$ and $(\delta\lambda)_k$ defined by (12.1.6) satisfy that

$$((\delta x)_k^T, (\delta\lambda)_k^T)\nabla P(x_k, \lambda_k) = -2P(x_k, \lambda_k) \leq 0. \tag{12.1.10}$$

Here ∇P is the gradient of P in the space (x, λ). The method given below is based on (12.1.6), hence it is called the Lagrange-Newton method.

Algorithm 12.1.1 *(Lagrange-Newton Method)*

Step 1. *Given* $x_1 \in \Re^n$, $\lambda_1 \in \Re^m$, $\beta \in (0, 1)$, $\epsilon \geq 0$, $k := 1$;

Step 2. *Compute* $P(x_k, \lambda_k)$; *if* $P(x_k, \lambda_k) \leq \epsilon$ *then stop;*
solving (12.1.6) obtaining $(\delta x)_k$ *and* $(\delta\lambda)_k$;
$\alpha = 1$;

Step 3. *if*

$$P(x_k + \alpha(\delta x)_k, \lambda_k + \alpha(\delta\lambda)_k) \leq (1 - \beta\alpha)P(x_k, \lambda_k), \tag{12.1.11}$$

then go to Step 4;
$\alpha = \alpha/4$, *go to Step 3;*

Step 4. $x_{k+1} = x_k + \alpha(\delta x)_k$; $\lambda_{k+1} = \lambda_k + \alpha(\delta\lambda)_k$;
$k := k + 1$; *go to Step 2.* □

For the above algorithm, we have the following convergence result.

Theorem 12.1.2 *Assume that* $f(x)$ *and* $c(x)$ *are twice continuously differentiable. If the matrix*

$$\begin{bmatrix} W(x_k, \lambda_k) & -A(x_k) \\ -A(x_k)^T & 0 \end{bmatrix}^{-1} \tag{12.1.12}$$

is uniformly bounded, then any accumulation point of $\{(x_k, \lambda_k)\}$ *generated by Algorithm 12.1.1 is a root of* $P(x, \lambda) = 0$.

Proof. Suppose that the theorem is not true, i.e., suppose $(\bar{x}, \bar{\lambda})$ is an accumulation point of $\{(x_k, \lambda_k)\}$ and

$$P(\bar{x}, \bar{\lambda}) > 0. \tag{12.1.13}$$

Then there exists a subset $K_0 \subseteq \{1, 2, \cdots\}$ which has infinitely many elements and satisfies that

$$\lim_{\substack{k \in K_0 \\ k \to \infty}} x_k = \bar{x}, \quad \lim_{\substack{k \in K_0 \\ k \to \infty}} \lambda_k = \bar{\lambda}. \tag{12.1.14}$$

From the line search condition (12.1.11), we can see that

$$P(x_{k+1}, \lambda_{k+1}) \leq (1 - \beta\alpha_k) P(x_k, \lambda_k). \tag{12.1.15}$$

It follows from (12.1.13)-(12.1.15) that

$$\lim_{\substack{k \in K_0 \\ k \to \infty}} \alpha_k = 0. \tag{12.1.16}$$

Therefore we have that

$$P(x_k + \hat{\alpha}_k(\delta x)_k, \lambda_k + \hat{\alpha}_k(\delta\lambda)_k) > (1 - \beta\hat{\alpha}_k) P(x_k, \lambda_k) \tag{12.1.17}$$

for all sufficiently large $k \in K_0$, where $\hat{\alpha}_k = 4\alpha_k \in (0, 1)$. Let $(\overline{\delta x}, \overline{\delta\lambda})$ be the solution of

$$\begin{pmatrix} W(\bar{x}, \bar{\lambda}) & -A(\bar{x}) \\ -A(\bar{x})^T & 0 \end{pmatrix} \begin{pmatrix} \delta x \\ \delta\lambda \end{pmatrix} = - \begin{pmatrix} \nabla f(\bar{x}) - \nabla c(\bar{x})^T \bar{\lambda} \\ c(\bar{x}) \end{pmatrix}. \tag{12.1.18}$$

Because $\hat{\alpha}_k \to 0$, we can show that

$$\lim_{\substack{k \in K_0 \\ k \to \infty}} \frac{P(\bar{x} + \hat{\alpha}_k \overline{\delta x}, \bar{\lambda} + \hat{\alpha}_k \overline{\delta\lambda}) - P(\bar{x}, \bar{\lambda})}{\hat{\alpha}_k} = -2P(\bar{x}, \bar{\lambda}) < -P(\bar{x}, \bar{\lambda}). \tag{12.1.19}$$

From the uniform boundedness of (12.1.12) and the fact that $(x_k, \lambda_k) \to (\bar{x}, \bar{\lambda})(k \in K_0)$, it follows that $((\delta x)_k, (\delta\lambda)_k) \to (\overline{\delta x}, \overline{\delta\lambda})$. Therefore, for sufficiently large $k \in K_0$ we have that

$$\frac{P(x_k + \hat{\alpha}_k(\delta x)_k, \lambda_k + \hat{\alpha}_k(\delta\lambda)_k) - P(x_k, \lambda_k)}{\hat{\alpha}_k} \leq -P(x_k, \lambda_k). \tag{12.1.20}$$

Because $\beta < 1$, (12.1.20) contradicts (12.1.17). This implies that the theorem is true. □

Theorem 12.1.3 *Assume that $f(x)$ and $c(x)$ are twice continuously differentiable. If the matrix (12.1.12) is uniformly bounded, then any accumulation point of $\{x_k\}$ generated by Algorithm 12.1.1 is a KKT point of (12.1.1)-(12.1.2).*

Proof. If the theorem is not true, it follows from the monotonicity of $P(x_k, \lambda_k)$ that

$$\lim_{k\to\infty} P(x_k, \lambda_k) > 0. \tag{12.1.21}$$

This limit and condition (12.1.11) imply that

$$\prod_{k=1}^{\infty}(1 - \beta\alpha_k) > 0. \tag{12.1.22}$$

The above relation indicates that

$$\sum_{k=1}^{\infty} \alpha_k < +\infty. \tag{12.1.23}$$

Because

$$\begin{bmatrix} W(x_k, \lambda_k) & -A(x_k) \\ -A(x_k)^T & 0 \end{bmatrix} \begin{bmatrix} (\delta x)_k \\ \lambda_k + (\delta\lambda)_k \end{bmatrix} = \begin{bmatrix} -\nabla f(x_k) \\ c(x_k) \end{bmatrix}, \tag{12.1.24}$$

there exists a positive constant $\gamma > 0$ such that

$$\|(\delta x)_k\| + \|\lambda_k + (\delta\lambda)_k\| \le \gamma(\|\nabla f(x_k)\| + \|c(x_k)\|). \tag{12.1.25}$$

Let $\bar{x}$ be any accumulation point of $\{x_k\}$. Define the set

$$S_\delta = \{x | \|x - \bar{x}\| \le \delta\}, \tag{12.1.26}$$

where $\delta > 0$ is any given positive constant. From (12.1.25) we know that there exists a constant $\eta > 0$ such that

$$\|(\delta x)_k\| \le \eta \tag{12.1.27}$$

for all $x_k \in S_\delta$. It follows from (12.1.23) that there exists $\bar{k}$ such that

$$\sum_{k=\bar{k}}^{\infty} \alpha_k < \frac{\delta}{2\eta}. \tag{12.1.28}$$

Because $\bar{x}$ is an accumulation point of $\{x_k\}$, there exists $\hat{k} > \bar{k}$ such that

$$\|x_{\hat{k}} - \bar{x}\| < \frac{\delta}{2}. \tag{12.1.29}$$

From (12.1.27)-(12.1.29) and the fact that $\|x_{k+1} - x_k\| = \alpha_k \|(\delta x)_k\|$ we have that

$$x_k \in S_\delta, \quad \forall k \geq \hat{k}. \tag{12.1.30}$$

Therefore (12.1.27) holds for all $k \geq \hat{k}$. Thus, it follows from (12.1.23) that

$$\lim_{k \to \infty} x_k = \bar{x}. \tag{12.1.31}$$

This relation and the last theorem imply that there are no accumulation points of $\{(x_k, \lambda_k)\}$, which shows that

$$\lim_{k \to \infty} \|\lambda_k\| = \infty. \tag{12.1.32}$$

Hence, it follows from (12.1.32) and (12.1.25) that

$$\begin{aligned} \|\lambda_{k+1}\| &= \|\lambda_k + \alpha_k (\delta\lambda)_k\| \\ &= \|(1 - \alpha_k)\lambda_k + \alpha_k(\lambda_k + (\delta\lambda)_k)\| \\ &= (1 - \alpha_k)\|\lambda_k\| + O(\alpha_k) < \|\lambda_k\| \end{aligned} \tag{12.1.33}$$

holds for all sufficiently large k, which contradicts (12.1.32). This completes our proof. □

About the convergence rate of Algorithm 12.1.1, we have the following result.

Theorem 12.1.4 *Assume that the sequence $\{x_k\}$ generated by Algorithm 12.1.1 converges to x^*, if $f(x)$ and $c(x)$ are three times continuously differentiable near x^*, $A(x^*)$ is full column rank, and the second-order sufficient condition is satisfied at x^*, then $\lambda_k \to \lambda^*$, and*

$$\left\| \begin{pmatrix} x_{k+1} - x^* \\ \lambda_{k+1} - \lambda^* \end{pmatrix} \right\| = O\left(\left\| \begin{pmatrix} x_k - x^* \\ \lambda_k - \lambda^* \end{pmatrix} \right\|^2 \right). \tag{12.1.34}$$

Proof. Because Algorithm 12.1.1 is the Newton-Raphson method for (12.1.4)-(12.1.5), and because the second-order sufficient condition implies that the matrix

$$\begin{bmatrix} W(x^*, \lambda^*) & -A(x^*) \\ -A(x^*)^T & 0 \end{bmatrix} \tag{12.1.35}$$

is nonsingular, we have that

$$\left\| \begin{pmatrix} x_k + (\delta x)_k - x^* \\ \lambda_k + (\delta\lambda)_k - \lambda^* \end{pmatrix} \right\| = O\left(\left\| \begin{pmatrix} x_k - x^* \\ \lambda_k - \lambda^* \end{pmatrix} \right\|^2 \right) \tag{12.1.36}$$

for all sufficiently large k. The above relation, and the fact that $f(x)$ and $c(x)$ are three times continuously differentiable imply that (12.1.11) holds for $\alpha = 1$. Therefore (12.1.34) holds. □

It should be pointed out that (12.1.34) is not equivalent to the usual quadratic convergence, which is

$$\|x_{k+1} - x^*\| = O(\|x_k - x^*\|^2). \tag{12.1.37}$$

For the analysis of the convergence rate of the iterates $\{x_k\}$, we need the following result.

Lemma 12.1.5 *Under the assumptions of Theorem 12.1.4, we have that*

$$\epsilon_{k+1} = O(\|x_k - x^*\|\epsilon_k), \tag{12.1.38}$$

where

$$\epsilon_k = \|x_k - x^*\| + \|\lambda_k - \lambda^*\|. \tag{12.1.39}$$

Proof. From the proof of Theorem 12.1.4 we see that $\alpha_k = 1$ for all sufficiently large k. Therefore it follows from the definitions of $(\delta x)_k$ and $(\delta\lambda)_k$ that

$$\begin{aligned}
&\begin{bmatrix} W(x_k, \lambda_k) & -A(x_k) \\ -A(x_k)^T & 0 \end{bmatrix} \begin{bmatrix} x_{k+1} - x^* \\ \lambda_{k+1} - \lambda^* \end{bmatrix} = \begin{bmatrix} -\nabla f(x_k) + A(x_k)\lambda_k \\ c(x_k) \end{bmatrix} \\
&\quad + \begin{bmatrix} W(x_k, \lambda_k)(x_k - x^*) - A(x_k)(\lambda_k - \lambda^*) \\ -A(x_k)^T(x_k - x^*) \end{bmatrix} \\
&\quad = \begin{bmatrix} (A(x^*) - A(x_k))(\lambda_k - \lambda^*) + O(\|x_k - x^*\|^2) \\ O(\|x_k - x^*\|^2) \end{bmatrix} \\
&\quad = \begin{bmatrix} O(\|x_k - x^*\|[\|x_k - x^*\| + \|\lambda_k - \lambda^*\|]) \\ O(\|x_k - x^*\|^2) \end{bmatrix} \\
&\quad = O(\|x_k - x^*\|\epsilon_k).
\end{aligned} \tag{12.1.40}$$

The above relation and the nonsingularity of matrix (12.1.35) show that the lemma holds. □

Theorem 12.1.6 *Under the assumptions of Theorem 12.1.4, the sequence* $\{x_k\}$ *converges to* x^* *superlinearly and*

$$\|x_{k+1} - x^*\| = o\left(\|x_k - x^*\| \prod_{j=1}^{p} \|x_{k-j} - x^*\|\right) \tag{12.1.41}$$

holds for any given positive integer p.

Proof. It follows from (12.1.38) that $\{x_k\}$ converges to x^* superlinearly. For any given positive integer p, applying (12.1.38) recursively we obtain that

$$\begin{aligned} \|x_{k+1} - x^*\| &= O(\epsilon_{k+1}) = O(\|x_k - x^*\|\epsilon_k) \\ &= O(\|x_k - x^*\| \ \|x_{k-1} - x^*\|\epsilon_{k-1}) \\ &= O\left(\|x_k - x^*\| \prod_{j=1}^{p} \|x_{k-j} - x^*\|\epsilon_{k-p}\right) \\ &= o\left(\|x_k - x^*\| \prod_{j=1}^{p} \|x_{k-j} - x^*\|\right). \end{aligned} \tag{12.1.42}$$

Therefore the theorem is true. □

One of the most important contributions of the Lagrange-Newton method is the development of the sequential quadratic programming method based on it. Sequential quadratic programming algorithms are the most important algorithms for solving medium and small scale nonlinear constrained optimization problems.

Setting $\bar{\lambda}_k = \lambda_k + (\delta\lambda)_k$, we can write (12.1.6) in the following equivalent form

$$W(x_k, \lambda_k)(\delta x)_k + \nabla f(x_k) = A(x_k)[\lambda_k + (\delta\lambda)_k], \tag{12.1.43}$$

$$c(x_k) + A(x_k)^T(\delta x)_k = 0, \tag{12.1.44}$$

which is just, in matrix form,

$$\begin{bmatrix} W(x_k, \lambda_k) & -A(x_k) \\ -A(x_k)^T & 0 \end{bmatrix} \begin{bmatrix} (\delta x) \\ \bar{\lambda} \end{bmatrix} = \begin{bmatrix} -g(x_k) \\ c(x_k) \end{bmatrix} \tag{12.1.45}$$

with solution $(\delta x)_k$ and $\bar{\lambda}_k$. Then x_{k+1} is given by

$$x_{k+1} = x_k + (\delta x)_k. \tag{12.1.46}$$

It is easy to show that the above system (12.1.45) is a KKT condition of the quadratic programming subproblem

$$\min_{d\in\Re^n} \quad d^T\nabla f(x_k) + \frac{1}{2}d^T W(x_k,\lambda_k)d, \tag{12.1.47}$$

$$\text{s.t.} \quad c(x_k) + A(x_k)^T d = 0 \tag{12.1.48}$$

with $((\delta x)_k, \bar{\lambda}_k)$ being the corresponding KKT pair thereof. Therefore, the Lagrange-Newton method can be viewed as a method that solves the quadratic programming subproblem (12.1.47)-(12.1.48) successively.

12.2 Wilson-Han-Powell Method

In this section we present a sequential quadratic programming method, which was proposed by Han [169]. The method is based on the Lagrange-Newton method discussed in the previous section. In each iteration the matrix $W(x_k,\lambda_k)$ is replaced by a matrix B_k. Because the Lagrange-Newton method was first considered by Wilson [349], and because Han's method was modified and analyzed by Powell [268], the method presented in this section is often called the Wilson-Han-Powell method.

Consider nonlinearly constrained optimization problem (8.1.1)-(8.1.3), Similar to (12.1.47)-(12.1.48), we construct the following subproblem

$$\min_{d\in\Re^n} \quad g_k^T d + \frac{1}{2}d^T B_k d, \tag{12.2.1}$$

$$\text{s.t.} \quad a_i(x_k)^T d + c_i(x_k) = 0, \ i\in E, \tag{12.2.2}$$

$$a_i(x_k)^T d + c_i(x_k) \geq 0, \quad i\in I, \tag{12.2.3}$$

where

$$A(x_k) = [a_1(x_k),\cdots,a_m(x_k)] = \nabla c(x_k)^T, \tag{12.2.4}$$

$g_k = g(x_k) = \nabla f(x_k)$, $E = \{1,2,\cdots m_e\}$, $I = \{m_e{+}1,\cdots,m\}$, and $B_k \in \Re^{n\times n}$ is an approximation to the Hessian matrix of the Lagrangian function. Let d_k be a solution of (12.2.1)-(12.2.3). The vector d_k is the search direction in the k-th iteration by the Wilson-Han-Powell method. Let λ_k be the corresponding Lagrange multiplier of (12.2.1)-(12.2.3) (just like $\bar{\lambda}_k$ in the previous section), then it follows that

$$g_k + B_k d_k = A(x_k)\lambda_k, \tag{12.2.5}$$

$$(\lambda_k)_i \geq 0, \quad i\in I, \tag{12.2.6}$$

$$(\lambda_k)_i[c_i(x_k) + a_i(x_k)^T d_k] = 0, \quad i\in I. \tag{12.2.7}$$

A very good property of d_k is that it is a descent direction of many penalty functions. For example, considering the L_1 exact penalty function, we have the following result.

Lemma 12.2.1 *Let d_k be a KKT point of (12.2.1)-(12.2.3) and λ_k be the corresponding Lagrange multiplier. Consider the L_1 penalty function*

$$P(x, \sigma) = f(x) + \sigma \|c^{(-)}(x)\|_1, \tag{12.2.8}$$

where $c^{(-)}(x)$ is defined by (10.1.2)-(10.1.3). Then we have that

$$P'_\alpha(x_k + \alpha d_k, \sigma)|_{\alpha=0} \leq -d_k^T B_k d_k - \sigma \|c^{(-)}(x_k)\|_1 + \lambda_k^T c(x_k). \tag{12.2.9}$$

If $d_k^T B_k d_k > 0$ and $\sigma \geq \|\lambda_k\|_\infty$, then d_k is a descent direction of the penalty function (12.2.8) at x_k.

Proof. By Taylor expression and using convexity of $\|(c + Ad)^{(-)}\|_1$, we have that

$$\begin{aligned} P'_\alpha(x_k + \alpha d_k, \sigma)|_{\alpha=0} &= \lim_{\alpha \to 0_+} \frac{P(x_k + \alpha d_k) - P(x_k)}{\alpha} \\ &= g_k^T d_k + \lim_{\alpha \to 0_+} \sigma \frac{\|[c(x_k) + \alpha A(x_k)^T d_k]^{(-)}\|_1 - \|c^{(-)}(x_k)\|_1}{\alpha} \\ &\leq g_k^T d_k + \sigma[\|(c(x_k) + A(x_k)^T d_k)^{(-)}\|_1 - \|c^{(-)}(x_k)\|_1] \\ &= g_k^T d_k - \sigma \|c^{(-)}(x_k)\|_1. \end{aligned} \tag{12.2.10}$$

It follows from (12.2.5) and (12.2.7) that

$$g_k^T d_k = -d_k^T B_k d_k + \lambda_k^T c(x_k). \tag{12.2.11}$$

Therefore (12.2.9) follows from (12.2.10) and (12.2.11).

Because λ_k satisfies (12.2.6), it follows from the definition of $c^{(-)}(x)$ that

$$\lambda_k^T c(x_k) \leq \sum_{i=1}^m |(\lambda_k)_i| \ |c_i^{(-)}(x_k)|. \tag{12.2.12}$$

Substituting the above inequality into (12.2.9), and using the assumptions that $d_k^T B_k d_k > 0$ and $\sigma \geq \|\lambda_k\|_\infty$, we have that

$$P'_\alpha(x_k + \alpha d_k, \sigma)|_{\alpha=0} \leq -d_k^T B_k d_k - \sum_{i=1}^m (\sigma - |(\lambda_k)_i|)|c_i^{(-)}(x_k)| < 0. \tag{12.2.13}$$

This shows that the lemma is true. □

The following algorithm is the sequential quadratic programming algorithm proposed by Han [169].

Algorithm 12.2.2

Step 1. *Given* $x_1 \in \Re^n$, $\sigma > 0$, $\delta > 0$, $B_1 \in \Re^{n\times n}$, $\epsilon \geq 0$, $k := 1$;

Step 2. *Solve (12.2.1)-(12.2.3) giving* d_k;
if $\|d_k\| \leq \epsilon$ *then stop;*
find $\alpha_k \in [0, \delta]$ *such that*

$$P(x_k + \alpha_k d_k, \sigma) \leq \min_{0\leq\alpha\leq\delta} P(x_k + \alpha d_k, \sigma) + \epsilon_k. \quad (12.2.14)$$

Step 3. $x_{k+1} = x_k + \alpha_k d_k$;
Evaluate $f(x_{k+1}), g_{k+1}, c(x_{k+1}), A_{k+1}$;

Step 4. *Compute* $\lambda_{k+1} = -(A_{k+1}^T A_{k+1})^{-1} A_{k+1}^T g_{k+1}$;
Set $s_k = \alpha d_k$, $y_k = \nabla_x L(x_{k+1}, \lambda_{k+1}) - \nabla_x L(x_k, \lambda_{k+1})$;
generate B_{k+1} *by updating* B_k *using a quasi-Newton formula;*
$k := k + 1$; *go to Step 2.* □

In (12.2.14), the penalty function $P(x, \sigma)$ is the L_1 exact penalty function, ϵ_k is a sequence of nonnegative numbers satisfying

$$\sum_{k=1}^{\infty} \epsilon_k < +\infty. \quad (12.2.15)$$

The global convergence result of the above algorithm is as follows.

Theorem 12.2.3 *Assume that* $f(x)$ *and* $c_i(x)$ *are continuously differentiable, and that there exist constants* $m, M > 0$ *such that*

$$m\|d\|^2 \leq d^T B_k d \leq M\|d\|^2 \quad (12.2.16)$$

holds for all k *and* $d \in \Re^n$, *if* $\|\lambda_k\|_\infty \leq \sigma$ *for all* k, *then any accumulation point of* $\{x_k\}$ *generated by Algorithm 12.2.2 is a KKT point of (8.1.1)-(8.1.3).*

Proof. If the theorem is not true, there exists a subsequence of $\{x_k\}$ converging to $\bar{x}$ which is not a KKT point. Therefore there exists a subset K_0 having infinitely many elements such that

$$\lim_{\substack{k\in K_0\\ k\to\infty}} x_k = \bar{x}. \tag{12.2.17}$$

Without loss of generality, we can assume that

$$\lim_{\substack{k\in K_0\\ k\to\infty}} \lambda_k = \bar{\lambda}, \quad \lim_{\substack{k\in K_0\\ k\to\infty}} B_k = \bar{B}. \tag{12.2.18}$$

If

$$\lim_{\substack{k\in K_0\\ k\to\infty}} \|d_k\| = 0, \tag{12.2.19}$$

from the relation

$$g_k + B_k d_k = A(x_k)\lambda_k, \tag{12.2.20}$$

it follows that

$$g(\bar{x}) = A(\bar{x})\bar{\lambda}. \tag{12.2.21}$$

This contradicts the fact that $\bar{x}$ is not a KKT point. Therefore we can assume that

$$\|d_k\| \geq \eta > 0, \quad \forall k \in K_0, \tag{12.2.22}$$

where η is a constant. The above relation and (12.2.13) imply that

$$P'_\alpha(x_k + \alpha d_k, \sigma)|_{\alpha=0} \leq -m\eta\|d_k\|, \tag{12.2.23}$$

holds for all $k \in K_0$. It follows from (12.2.23) and the continuity assumptions on the functions that there exists a positive constant $\bar{\eta}$ such that

$$\min_{0\leq\alpha\leq\delta} P(x_k + \alpha d_k, \sigma) \leq P(x_k, \sigma) - \bar{\eta} \tag{12.2.24}$$

hold for all $k \in K_0$. Thus,

$$P(x_{k+1}, \sigma) \leq P(x_k, \sigma) - \bar{\eta} + \epsilon_k, \quad \forall k \in K_0. \tag{12.2.25}$$

Consequently we can derive the inequality

$$\begin{aligned} \sum_{k\in K_0} \bar{\eta} &\leq \sum_{k\in K_0} [P(x_k, \sigma) - P(x_{k+1}, \sigma)] + \sum_{k\in K_0} \epsilon_k \\ &\leq \sum_{k=1}^{\infty} [P(x_k, \sigma) - P(x_{k+1}, \sigma)] + \sum_{k=1}^{\infty} \epsilon_k. \end{aligned} \tag{12.2.26}$$

Because $\lim_{k\to\infty} P(x_k, \sigma) = P(\bar{x}, \sigma)$, it follows that

$$\sum_{k\in K_0} \bar{\eta} \leq P(x_1, \sigma) - P(\bar{x}, \sigma) + \sum_{k=1}^{\infty} \epsilon_k < +\infty. \tag{12.2.27}$$

From the above inequality and $\bar{\eta} > 0$ we see that K_0 can have only finitely many elements, which contradicts our assumption in the beginning of the proof. This indicates that the theorem is true. □

The global convergence requires that

$$\sigma > \|\lambda_k\|_\infty \tag{12.2.28}$$

for all k. However, in practice it is very difficult to choose such a penalty parameter. If σ is too small, condition (12.2.28) may be violated. If σ is too large, the step-length α_k may tend to be too small to prevent the fast convergence of the algorithm. Powell [268] suggests using exact penalty function

$$P(x, \sigma_k) = f(x) + \sum_{i=1}^{m} (\sigma_k)_i |c_i^{(-)}(x)| \tag{12.2.29}$$

in the k-th iteration, where $(\sigma_k)_i > 0$ and these parameters are updated in the following way.

$$(\sigma_1)_i = (\lambda_1)_i, \tag{12.2.30}$$

$$(\sigma_k)_i = \max\left\{ |[\lambda_k]_i|, \frac{1}{2}[(\sigma_{k-1})_i + |(\lambda_k)_i|] \right\}, \quad k > 1, \tag{12.2.31}$$

for all $i = 1, \cdots, m$. The parameters σ_k defined above satisfy

$$(\sigma_k)_i \geq |(\lambda_k)_i|, \quad i = 1, 2, \cdots, m. \tag{12.2.32}$$

This very clever update technique allows the penalty parameters to change from iteration to iteration, and, intuitively, the inequality (12.2.32) offers a similar property to (12.2.28). But, because $(\sigma_k)_i$ are not constants, the conditions of Theorem 12.2.3 do not hold. And Chamberlain [53] gives an example to show that cycles may happen due to this update technique.

Now we discuss the update of B_{k+1}, which is usually generated by a certain quasi-Newton formula. From our analyses in Section 12.1, we hope that B_{k+1} is an approximation to the Hessian matrix of the Lagrangian function.

Similar to unconstrained optimization, we can apply the standard quasi-Newton updates using

$$\begin{aligned} s_k &= x_{k+1} - x_k, && (12.2.33)\\ y_k &= \nabla f(x_{k+1}) - \nabla f(x_k) \\ &\quad - \sum_{i=1}^{m} (\lambda_k)_i [\nabla c_i(x_{k+1}) - \nabla c_i(x_k)]. && (12.2.34) \end{aligned}$$

A crucial difference is that line

$$s_k^T y_k > 0, \tag{12.2.35}$$

which would be always true for unconstrained optimization. Therefore, for example, we can not directly apply the BFGS update. Powell [268] suggests that y_k be replaced by

$$\bar{y}_k = \begin{cases} y_k, & \text{if } s_k^T y_k \geq 0.2 s_k^T B_k s_k, \\ \theta_k y_k + (1-\theta_k) B_k s_k, & \text{otherwise} \end{cases} \tag{12.2.36}$$

where

$$\theta_k = \frac{0.8 s_k^T B_k s_k}{s_k^T B_k s_k - s_k^T y_k}. \tag{12.2.37}$$

The vector $\bar{y}_k$ defined above satisfies $s_k^T \bar{y}_k > 0$.

The idea of such a choice of $\bar{y}_k$ is to obtain an update vector using the convex combination of y_k and $B_k s_k$. Because $B_k s_k$ can also be viewed as an approximation to y_k, because it satisfies (assuming that B_k is positive definite)

$$s_k^T (B_k s_k) > 0, \tag{12.2.38}$$

it is very natural to use the convex combination of y_k and $B_k s_k$. The geometric interpretation of Powell's formula is as follows. Suppose we normalize the length of the projection of $B_k s_k$ to direction s_k. The rule (12.2.36)-(12.2.37) is in fact to choose $\bar{y}_k$ from the line segment between y_k and $B_k s_k$ that is as close to y_k as possible and whose projection to s_k is at least 0.2. This is shown in Figure 12.2.1.

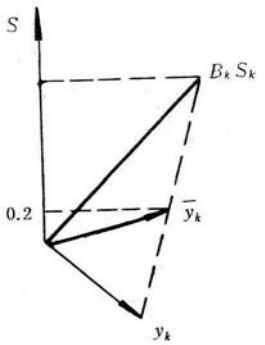


Figure 12.2.1

Having computed the vector $\bar{y}_k$, we can now apply the BFGS formula to update B_{k+1}:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k^T}{s_k^T B_k s_k} + \frac{\bar{y}_k \bar{y}_k^T}{s_k^T \bar{y}_k}. \tag{12.2.39}$$

Another way to modify y_k is to use

$$\hat{y}_k = y_k + 2\rho \sum_{i=1}^{m} -c_i(x_k)\nabla c_i(x_k) \tag{12.2.40}$$

to replace y_k, where $\rho > 0$ is a parameter. Because

$$\hat{y}_k \approx [\nabla^2 L(x_k, \lambda_k) + 2\rho A(x_k) A(x_k)^T] s_k, \tag{12.2.41}$$

updating B_k by using $\hat{y}_k$ can be viewed as making B_{k+1} approximate the Hessian matrix of the augmented Lagrange function. An advantage of this choice is that

$$s_k^T \hat{y}_k > 0 \tag{12.2.42}$$

can usually be satisfied. If $s_k^T \hat{y}_k \leq 0$, we can always make (12.2.42) hold by increasing ρ, unless $\|A(x_k)s_k\| = 0$. Normally, the Hessian matrix of the augmented Lagrange function is positive definite, thus it is very reasonable to use a positive definite matrix B_k to approximate it.

12.3 Superlinear Convergence of SQP Step

In order to prove the superlinear convergence property of the sequential quadratic programming method, i.e.

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0, \tag{12.3.1}$$

we only need to show that the search direction d_k satisfies

$$\lim_{k\to\infty} \frac{\|x_k + d_k - x^*\|}{\|x_k - x^*\|} = 0 \tag{12.3.2}$$

and that the line search condition will allow $\alpha_k = 1$ for all large k if (12.3.2) holds. Thus, the important thing is to show that the search direction generated by the SQP method satisfies (12.3.2). A step d_k satisfying (12.3.2) is called a superlinearly convergent step. In this section, we discuss the conditions for ensuring the sequential quadratic programming method to produce superlinearly convergent steps.

Throughout this section, we make the following assumptions.

Assumption 12.3.1

1) $f(x), c_i(x)$ *are twice continuously differentiable;*

2) $x_k \to x^*$*;*

3) x^* *is a KKT point and*

$$\nabla c_i(x^*), \quad i \in E \cup I(x^*) \tag{12.3.3}$$

are linearly independent. Let $A(x^*)$ *be the* $n \times |E \cup I(x^*)|$ *matrix consisting of the vectors given in (12.3.3). For all nonzero vectors* d *satisfying*

$$A(x^*)^T d = 0, \tag{12.3.4}$$

we have that

$$d^T W(x^*, \lambda^*) d \neq 0, \tag{12.3.5}$$

where $W(x^*, \lambda^*)$ *is defined by (12.1.8), and* λ^* *is the Lagrange multiplier at* x^**.*

The above assumptions are often used for local convergence analyses of algorithms for constrained optimization. For example, relations (12.3.4) and (12.3.5) hold if we assume the second-order sufficient condition

$$d^T W(x^*, \lambda^*) d > 0, \quad \forall d \neq 0, \ A(x^*)^T d = 0. \tag{12.3.6}$$

We also make the assumption that the active set at the solution can be identified when the iterations are very close to the solution. Therefore, when k is sufficiently large, the search direction d_k is actually the solution of an equality constrained quadratic programming subproblem.

Assumption 12.3.2 *For sufficiently large k, d_k is a solution of*

$$\min_{d \in \Re^n} \quad g_k^T d + \frac{1}{2} d^T B_k d \tag{12.3.7}$$

$$s.t. \quad c_i(x_k) + d^T \nabla c_i(x_k) = 0, \quad i \in E \cup I(x^*). \tag{12.3.8}$$

Under Assumption 12.3.2, for all large k there exists $\lambda_k \in \Re^{|E \cup I(x^*)|}$ such that

$$g_k + B_k d_k = A(x_k)\lambda_k, \tag{12.3.9}$$

$$A(x_k)^T d_k = -\hat{c}(x_k), \tag{12.3.10}$$

where $\hat{c}(x)$ is a vector whose elements are $c_i(x)(i \in E \cup I(x^*))$.

Theorem 12.3.3 *Under the conditions of Assumptions 12.3.1 and 12.3.2, d_k is a superlinearly convergent step, namely*

$$\lim_{k \to \infty} \frac{\|x_k + d_k - x^*\|}{\|x_k - x^*\|} = 0 \tag{12.3.11}$$

if and only if

$$\lim_{k \to \infty} \frac{\|P_k(B_k - W(x^*, \lambda^*))d_k\|}{\|d_k\|} = 0, \tag{12.3.12}$$

where P_k is a projection from $\Re^n$ onto the null space of $A(x_k)^T$:

$$P_k = I - A(x_k)(A(x_k)^T A(x_k))^{-1} A(x_k)^T. \tag{12.3.13}$$

Proof. From (12.3.9) and the definition of P_k, we have that

$$\begin{aligned} P_k B_k d_k &= -P_k g_k = -P_k[\nabla f(x_k) - A(x_k)\lambda^*] \\ &= -P_k[\nabla_x L(x_k, \lambda^*) - \nabla_x L(x^*, \lambda^*)] \\ &= -P_k W(x^*, \lambda^*)(x_k - x^*) + O(\|x_k - x^*\|^2). \end{aligned} \quad (12.3.14)$$

Therefore, it follows that

$$\begin{aligned} P_k(B_k - W(x^*, \lambda^*))d_k &= -P_k W(x^*, \lambda^*)[x_k + d_k - x^*] \\ &\quad + O(\|x_k - x^*\|^2). \end{aligned} \quad (12.3.15)$$

Using relation (12.3.10) and

$$\begin{aligned} \hat{c}(x_k) &= \hat{c}(x_k) - \hat{c}(x^*) \\ &= A(x_k)^T(x_k - x^*) + O(\|x_k - x^*\|^2), \end{aligned} \quad (12.3.16)$$

we can show that

$$A(x_k)^T(x_k + d_k - x^*) = O(\|x_k - x^*\|^2). \quad (12.3.17)$$

Equations (12.3.15) and (12.3.17) can be rewritten in matrix form:

$$\begin{aligned} \begin{bmatrix} P_k W(x^*, \lambda^*) \\ A(x_k)^T \end{bmatrix} (x_k + d_k - x^*) &= \begin{bmatrix} -P_k(B_k - W(x^*, \lambda^*))d_k \\ 0 \end{bmatrix} \\ &+ O(\|x_k - x^*\|^2). \end{aligned} \quad (12.3.18)$$

Define the matrix

$$G^* = \begin{bmatrix} P_* W(x^*, \lambda^*) \\ A(x^*)^T \end{bmatrix}, \quad (12.3.19)$$

where $P_* = I - A(x^*)(A(x^*)^T A(x^*))^{-1} A(x^*)^T$. For any $d \in \Re^n$, if $G^* d = 0$ we have that

$$A(x^*)^T d = 0, \quad (12.3.20)$$

$$d^T P_* W(x^*, \lambda^*) d = 0. \quad (12.3.21)$$

From (12.3.20) it follows that $P_* d = d$. Thus,

$$d^T W(x^*, \lambda^*) d = 0. \quad (12.3.22)$$

The above relation and Assumption 12.3.1 show that $d = 0$. Therefore matrix G^* is a full column rank matrix. Hence, from (12.3.18) and the fact that $x_k \to x^*$ we can see that (12.3.11) is equivalent to

$$\lim_{k\to\infty} \frac{\|P_k(B_k - W(x^*, \lambda^*))d_k\|}{\|x_k - x^*\|} = 0. \tag{12.3.23}$$

Using the equivalence between (12.3.23) and (12.3.11) and that between (12.3.11) and

$$\lim_{k\to\infty} \|x_k - x^*\| \Big/ \|d_k\| = 1, \tag{12.3.24}$$

we can show that (12.3.23) is equivalent to (12.3.12). This completes the proof. □

Using relation (12.3.9) and $\lambda_k \to \lambda^*$, we have that

$$\begin{aligned} & W(x^*, \lambda^*)d_k = W(x_k, \lambda_k)d_k + o(\|d_k\|) \\ = \;& \nabla f(x_k + d_k) - A(x_k + d_k)\lambda_k - \nabla f(x_k) + A(x_k)\lambda_k + o(\|d_k\|) \\ = \;& \nabla f(x_k + d_k) - A(x_k + d_k)\lambda_k + B_k d_k + o(\|d_k\|). \end{aligned} \tag{12.3.25}$$

Therefore,

$$\begin{aligned} P_k(B_k - W(x^*, \lambda^*))d_k \;=\; & -P_k[\nabla f(x_k + d_k) - A(x_k + d_k)\lambda_k] \\ & + o(\|d_k\|). \end{aligned} \tag{12.3.26}$$

From the above relation and Theorem 12.3.3, we can get the following result.

Corollary 12.3.4 *Under the assumptions of Theorem 12.3.3, (12.3.11) is equivalent to*

$$\lim_{k\to\infty} \frac{\|P_k[\nabla f(x_k + d_k) - A(x_k + d_k)\lambda_k\|}{\|d_k\|} = 0. \tag{12.3.27}$$

From Theorem 12.3.3, we should choose B_k such that (12.3.12) is satisfied in order to have superlinear convergence, namely B_k should be a good approximation to $W(x^*, \lambda^*)$.

12.4 Maratos Effect

For unconstrained optimization, if x^* is a stationary point at which the second-order sufficient condition holds, namely

$$\nabla^2 f(x^*) \quad \text{positive definite}, \tag{12.4.1}$$

if $x_k \to x^*$, and if d_k is a superlinearly convergent step, then

$$f(x_k + d_k) < f(x_k) \tag{12.4.2}$$

holds for all large k. That is to say, superlinearly convergent steps are acceptable for unconstrained problems. However, this is not always true for constrained problems. Such a phenomenon was first discovered by Maratos [209], so it is called the Maratos Effect.

Consider the equality constrained optimization problem

$$\min_{x=(u,v)\in\Re^2} \quad f(x) = 3v^2 - 2u, \tag{12.4.3}$$

$$\text{s.t.} \quad c(x) = u - v^2 = 0. \tag{12.4.4}$$

It is easy to see that $x^* = (0,0)^T$ is the unique minimizer and condition 3) of Assumption 12.3.1 is satisfied. In fact, the second-order sufficient condition holds at x^*. Consider any points that are close to the solution x^* and that have the form

$$\bar{x}(\epsilon) = (u(\epsilon), v(\epsilon))^T = (\epsilon^2, \epsilon)^T \tag{12.4.5}$$

where $\epsilon > 0$ is a small parameter. Let $B = W(x^*, \lambda^*)$; the quadratic programming subproblem is

$$\min_{d\in\Re^2} \quad d^T \begin{pmatrix} -2 \\ 6\epsilon \end{pmatrix} + \frac{1}{2} d^T \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} d, \tag{12.4.6}$$

$$\text{s.t.} \quad d^T \begin{pmatrix} 1 \\ -2\epsilon \end{pmatrix} = 0. \tag{12.4.7}$$

It is easy to see that the solution of (12.4.6)-(12.4.7) is

$$\bar{d}(\epsilon) = \begin{bmatrix} -2\epsilon^2 \\ -\epsilon \end{bmatrix}. \tag{12.4.8}$$

Therefore, we have that

$$\|\bar{x}(\epsilon) + \bar{d}(\epsilon) - x^*\| = O(\|\bar{x}(\epsilon) - x^*\|^2). \tag{12.4.9}$$

Thus, $\bar{d}(\epsilon)$ is a superlinearly convergent step. Direct calculation indicates that

$$f(\bar{x}(\epsilon) + \bar{d}(\epsilon)) = 2\epsilon^2, \tag{12.4.10}$$
$$c(\bar{x}(\epsilon) + \bar{d}(\epsilon)) = -\epsilon^2. \tag{12.4.11}$$

Because

$$f(\bar{x}(\epsilon)) = \epsilon^2, \tag{12.4.12}$$
$$c(\bar{x}(\epsilon)) = 0, \tag{12.4.13}$$

we have that

$$f(\bar{x}(\epsilon) + \bar{d}(\epsilon)) > f(\bar{x}(\epsilon)), \tag{12.4.14}$$
$$|c(\bar{x}(\epsilon) + \bar{d}(\epsilon))| > |c(\bar{x}(\epsilon))|. \tag{12.4.15}$$

This example shows that even though $\bar{d}(\epsilon)$ is a superlinearly convergent step (namely $\bar{x}(\epsilon) + \bar{d}(\epsilon)$ is much closer to x^* than $\bar{x}(\epsilon)$), the point $\bar{x}(\epsilon) + \bar{d}(\epsilon)$ is "worse" than $\bar{x}(\epsilon)$ from the objective function values and from the constraint violations. In fact, for any penalty functions $P_{\sigma,h}(x)$ having the form of (10.6.2), we would have that

$$P_{\sigma,h}(\bar{x}(\epsilon) + \bar{d}(\epsilon)) > P_{\sigma,h}(\bar{x}(\epsilon)). \tag{12.4.16}$$

Especially, when the merit function is the L_1 exact penalty function, $\bar{x}(\epsilon) + \bar{d}(\epsilon)$ is not acceptable.

The Maratos Effect shows that for many penalty functions a superlinearly convergent step may not be accepted, which, sometimes, prevents the algorithm from fast convergence.

There are mainly three ways to overcome the Maratos Effect. The first one is to relax the line search conditions. Roughly speaking, since the search direction d_k is a superlinearly convergent step, we should choose $\alpha_k = 1$ as often as possible provided that convergence is ensured. The second one is to use a second-order correction step $\hat{d}_k$, where $\hat{d}_k$ satisfies $\|\hat{d}_k\| = O(\|d_k\|^2)$, and $P_\sigma(x_k + d_k + \hat{d}_k) < P_\sigma(x_k)$. In this way, $d_k + \hat{d}_k$ is an acceptable step and it is still a superlinearly convergent step. The third way is to use smooth

exact penalty functions as merit functions. If the penalty function $P_\sigma(x)$ is smooth, we can show that

$$P_\sigma(x_k + d_k) < P_\sigma(x_k) \tag{12.4.17}$$

for all large k as long as (12.3.11) holds.

We will discuss these three techniques in the following sections.

12.5 Watchdog Technique

The nature of the Maratos Effect is the inequality

$$P_\sigma(x_k + d_k) > P_\sigma(x_k), \tag{12.5.1}$$

makes $x_{k+1} \neq x_k + d_k$, therefore the superlinearly convergent property is destroyed. In the Watchdog technique proposed by Chamberlain et. al. [54], the standard linear search which implies that

$$P_\sigma(x_{k+1}) < P_\sigma(x_k), \tag{12.5.2}$$

is used in some iterations, but in the other iterations, the line search conditions are relaxed. The relaxed line search can be either simply $\alpha_k = 1$ or requiring the Lagrange function to be reduced. Assume that the new point obtained in one iteration yields a sufficient reduction on the merit function $P_\sigma(x)$; comparing with the best point in the previous iterations, we can use the relaxed line search in the next iteration.

Define the function

$$P_\sigma(x) = f(x) + \sum_{i=1}^{m_e} \sigma_i |c_i(x)| + \sum_{i=m_e+1}^{m} \sigma_i |\min[0, c_i(x)]|, \tag{12.5.3}$$

and the approximate models

$$\begin{aligned} P_\sigma^{(k)}(x) &= f(x_k) + (x - x_k)^T \nabla f(x_k) + \frac{1}{2}(x - x_k)^T B_k (x - x_k) \\ &+ \sum_{i=1}^{m_e} \sigma_i |c_i(x_k) + (x - x_k)^T \nabla c_i(x_k)| \\ &+ \sum_{i=m_e+1}^{m} \sigma_i |\min[0, c_i(x_k) + (x - x_k)^T \nabla c_i(x_k)]|. \end{aligned} \tag{12.5.4}$$

Assume that $l \leq k$ is the index in which the best point has been found up to the k-th iteration, namely

$$P_\sigma(x_l) = \min_{1 \leq i \leq k} P_\sigma(x_i). \tag{12.5.5}$$

Let $\beta \in \left(0, \frac{1}{2}\right)$ be a given positive constant. If the iterate point obtained in the k-th iteration $x_{k+1} = x_k + \alpha_k d_k$ satisfies

$$P_\sigma(x_{k+1}) \leq P_\sigma(x_l) - \beta[P_\sigma(x_l) - P_\sigma^{(l)}(x_{l+1})], \tag{12.5.6}$$

then we say x_{k+1} (comparing to x_l) yields a "sufficient" reduction on the merit function $P_\sigma(x)$.

The following is an algorithm with the Watchdog technique.

Algorithm 12.5.1 *(Watchdog Method)*

Step 1. *Given $x_1 \in \Re^n$, a positive constant $\bar{n}$.*
Set line search type to be standard; $k := l := 1$;

Step 2. *Compute the search direction d_k;*
Carry out line search using the line search type, obtaining $\alpha_k > 0$;
$x_{k+1} = x_k + \alpha_k d_k$;

Step 3. *if (12.5.6) holds, then set the next line search type to be "relaxed", otherwise to be standard.*

Step 4. *if $P_\sigma(x_{k+1}) \leq P_\sigma(x_l)$, then $l := k + 1$;*

Step 5. *if $k < l + \bar{n}$, then go to Step 6;*
$x_{k+1} := x_l$; $l := k + 1$;

Step 6. *if convergence criterion is satisfied then stop;*
$k := k + 1$; go to Step 2. □

Actually, if the "relaxed" line search conditions are the same as the standard condition, the above algorithm is the original method that is based on the standard line searches. Therefore, the Watchdog method is a generalization of the standard method.

Assume the standard line search condition is

$$P_\sigma(x_{k+1}) \leq P_\sigma(x_k) - \beta[P_\sigma(x_k) - P_\sigma^{(k)}(x_{k+1})]. \tag{12.5.7}$$

From the descriptions of the above algorithm, we know that there exists $k \leq l + \bar{n} + 1$ such that

$$P_\sigma(x_{k+1}) \leq P_\sigma(x_l) - \beta[P_\sigma(x_l) - P_\sigma^{(l)}(x_{l+1})]. \tag{12.5.8}$$

Thus, the watchdog method will reduce the merit function $P_\sigma(x)$ in every $\bar{n}+1$ iterations, even though it can not guarantee the monotonically decreasing of $P_\sigma(x_k)$. Let $l(j)$ be the j-th value of l; from the discussions above we see that

$$l(j) < l(j+1) \leq l(j) + \bar{n} + 2. \tag{12.5.9}$$

If we assume that the sequence $\{x_k\}$ is bounded, then $P_\sigma(x_{l(j)})$ will not tend to negative infinity. Thus, it follows from the inequality

$$P_\sigma(x_{l(j+1)}) \leq P_\sigma(x_{l(j)}) - \beta[P_\sigma(x_{l(j)}) - P_\sigma^{(l(j))}(x_{l(j)+1})] \tag{12.5.10}$$

that

$$\sum_{j=1}^{\infty}[P_\sigma(x_{l(j)}) - P_\sigma^{(l(j))}(x_{l(j)+1})] < +\infty. \tag{12.5.11}$$

The above relation shows that there exists an accumulation point of $\{x_k\}$ that is a KKT point of the constrained optimization problem.

12.6 Second-Order Correction Step

A second-order correction step is a vector $\hat{d}_k$ such that

$$\|\hat{d}_k\| = O(\|d_k\|^2) \tag{12.6.1}$$

and

$$P_\sigma(x_k + d_k + \hat{d}_k) < P_\sigma(x_k) \tag{12.6.2}$$

for all sufficiently large k. Consider that $\hat{d}_k$ is defined as a solution of the following quadratic programming problem:

$$\min_{d \in \Re^n} \quad g_k^T(d_k + d) + \frac{1}{2}(d_k + d)^T B_k(d_k + d), \tag{12.6.3}$$

$$\text{s.t.} \quad c_i(x_k + d_k) + a_i(x_k)^T d = 0, \quad i \in E, \tag{12.6.4}$$

$$c_i(x_k + d_k) + a_i(x_k)^T d \geq 0, \quad i \in I, \tag{12.6.5}$$

where d_k is the solution of (12.2.1)-(12.2.3).

For simplicity, we assume that all the constraints are equality constraints. We also assume that the second-order sufficient conditions hold at x^* and that $x_k \to x^*$. From the KKT condition there exist $\lambda_k \in \Re^m$ and $\hat{\lambda}_k \in \Re^m$ such that

$$B_k d_k = -g_k + A(x_k)\lambda_k, \tag{12.6.6}$$
$$A(x_k)^T d_k = -c(x_k) \tag{12.6.7}$$

and that

$$B_k d_k + B_k \hat{d}_k = -g_k + A(x_k)\hat{\lambda}_k, \tag{12.6.8}$$
$$A(x_k)^T \hat{d}_k = -c(x_k + d_k). \tag{12.6.9}$$

From (12.6.6) and (12.6.8) we see that

$$P_k B_k \hat{d}_k = 0, \tag{12.6.10}$$

where P_k is defined by (12.3.13). We make the following assumptions.

Assumption 12.6.1

1) $x_k \to x^*$;

2) $A(x^*)$ *is full column rank;*

3) there exist positive constants $\bar{m}$ *and* $\bar{M}$ *such that* $\|B_k\| \le \bar{M}$ *and that*

$$d^T B_k d \ge \bar{m}\|d\|_2^2 \tag{12.6.11}$$

holds for all d *satisfying* $A(x_k)^T d = 0$ *for all* k.

From the above assumptions we can show the following lemma.

Lemma 12.6.2 *Under the conditions of Assumption 12.6.1, there exists a positive constant* η *such that*

$$\left\| \begin{pmatrix} P_k B_k \\ A(x_k)^T \end{pmatrix} d \right\|_2 \ge \eta \|d\|_2 \tag{12.6.12}$$

holds for all $d \in \Re^n$ *and all sufficiently large* k.

Proof. Let the QR factorization of $A(x_k)$ be

$$A(x_k) = [Y_k \ Z_k] \begin{bmatrix} R_k \\ 0 \end{bmatrix}. \tag{12.6.13}$$

Because $A(x^*)$ is nonsingular, there exists k_0 such that for $k \geq k_0$ we have

$$\|R_k^{-1}\|_2 \leq \hat{\eta}, \tag{12.6.14}$$

where $\hat{\eta} > 0$ is a constant. Therefore,

$$\|A(x_k)^T d\|_2 = \|R_k^T Y_k^T d\|_2 \geq \frac{1}{\hat{\eta}} \|Y_k^T d\|_2, \tag{12.6.15}$$

for $k \geq k_0$. Using the relation $Y_k Y_k^T + Z_k Z_k^T = I$, we can show that

$$\begin{aligned} \|P_k B_k d\|_2 &= \|Z_k Z_k^T B_k d\|_2 \\ &= \|Z_k Z_k^T B_k Y_k Y_k^T d + Z_k Z_k^T B_k Z_k Z_k^T d\|_2 \\ &\geq \|Z_k Z_k^T B_k Z_k Z_k^T d\|_2 - \|B_k\|_2 \|Y_k^T d\|_2 \\ &\geq \bar{m} \|Z_k^T d\|_2 - \bar{M} \|Y_k^T d\|_2. \end{aligned} \tag{12.6.16}$$

Thus, if

$$\|Y_k^T d\| \geq \frac{\bar{m}}{2\bar{M}} \|Z_k^T d\|, \tag{12.6.17}$$

it follows from (12.6.15) that

$$\begin{aligned} \|A(x_k)^T d\|_2 &\geq \frac{1}{\hat{\eta}} \|Y_k^T d\|_2 \\ &\geq \frac{\frac{\bar{m}}{2\bar{M}}}{\hat{\eta}\sqrt{1 + \left(\frac{\bar{m}}{2\bar{M}}\right)^2}} \|d\|_2. \end{aligned} \tag{12.6.18}$$

If (12.6.17) does not hold, it follows from (12.6.16) that

$$\|P_k B_k d\|_2 \geq \frac{1}{2}\bar{m} \|Z^T d\|_2 \geq \frac{\bar{M}}{\sqrt{1 + \left(\frac{2\bar{M}}{\bar{m}}\right)^2}} \|d\|_2. \tag{12.6.19}$$

Therefore, when $k \geq k_0$, either of (12.6.18) and (12.6.19) must hold. Let

$$\eta = \min\left\{\frac{1}{\hat{\eta}}, \bar{M}\right\} \frac{1}{\sqrt{1 + 4(\bar{M}/\bar{m})^2}}, \tag{12.6.20}$$

then we see that (12.6.12) holds for all $k \geq k_0$ and all $d \in \Re^n$. □

Using (12.6.9)-(12.6.10), we have that

$$\left[\begin{array}{c} P_k B_k \\ A(x_k)^T \end{array} \right] \hat{d}_k = \left[\begin{array}{c} 0 \\ -c(x_k + d_k) \end{array} \right] = O(\|d_k\|_2^2). \tag{12.6.21}$$

Now, from the above relation and Lemma 12.6.2 we can show the following lemma.

Lemma 12.6.3 *Under the conditions of Assumption 12.6.1, there exists a positive constant $\bar{\eta} > 0$ such that*

$$\|\hat{d}_k\|_2 \leq \bar{\eta} \|d_k\|_2^2. \tag{12.6.22}$$

To this end, we have shown that the step defined by (12.6.3)-(12.6.5) is indeed a second-order correction step.

In the following we show that the second-order correction step $\hat{d}_k$ will make the step $d_k + \hat{d}_k$ acceptable. First, using (12.6.9) we see that

$$\begin{aligned} c(x_k + d_k + \hat{d}_k) &= c(x_k + d_k) + A(x_k)^T \hat{d}_k + o(\|\hat{d}_k\|) \\ &= o(\|d_k\|^2) = o(\|x_k - x^*\|^2). \end{aligned} \tag{12.6.23}$$

Define the vector

$$\bar{d}_k = -(A(x_k)^T)^+ c(x_k + d_k) - P_k(x_k + d_k - x^*), \tag{12.6.24}$$

then it follows that

$$\begin{aligned} \|x_k + d_k + \bar{d}_k - x^*\| &= \|(I - P_k)(x_k + d_k - x^*) \\ &\quad -(A(x_k)^T)^+ c(x_k + d_k)\| \\ &= \|(I - P_k)(x_k + d_k - x^*) \\ &\quad -(A(x_k)^T)^+ A(x_k)^T (x_k + d_k - x^*)\| \\ &\quad +o(\|x_k - x^*\|^2) = o(\|x_k - x^*\|^2). \end{aligned} \tag{12.6.25}$$

Furthermore, it follows from (12.6.24) that

$$A(x_k)^T \bar{d}_k = -c(x_k + d_k). \tag{12.6.26}$$

If we assume not only (12.3.12) but also

$$\frac{\|(B_k - W(x^*, \lambda^*))d\|}{\|d\|} \to 0 \tag{12.6.27}$$

holds for $d = d_k + \hat{d}_k$, and $d = d_k + \bar{d}$, then it follows that

$$\begin{aligned} & (g_k - A_k\lambda^*)^T d + \frac{1}{2} d^T B_k d \\ = \; & L(x_k + d, \lambda^*) - L(x_k, \lambda^*) + o(\|d\|^2) + o(\|x_k - x^*\|^2) \\ = \; & L(x_k + d, \lambda^*) - L(x_k, \lambda^*) + o(\|x_k - x^*\|^2) \end{aligned} \tag{12.6.28}$$

holds for $d = d_k + \hat{d}_k$ and $d = d_k + \bar{d}_k$. From the definition of $\hat{d}_k$, we can show that

$$\begin{aligned} g_k^T \hat{d}_k \; + \; & \frac{1}{2}(d_k + \hat{d}_k)^T B_k (d_k + \hat{d}_k) \\ \leq \; & g_k^T \bar{d}_k + \frac{1}{2}(d_k + \bar{d}_k)^T B_k (d_k + \bar{d}_k). \end{aligned} \tag{12.6.29}$$

If follows from (12.6.28) and (12.6.29) that

$$\begin{aligned} L(x_k + d_k + \hat{d}_k, \lambda^*) \; \leq \; & L(x_k + d_k + \bar{d}_k, \lambda^*) + o(\|x_k - x^*\|^2) \\ \leq \; & L(x^*, \lambda^*) + o(\|x_k - x^*\|^2). \end{aligned} \tag{12.6.30}$$

The above inequality and (12.6.23) imply that

$$f(x_k + d_k + \hat{d}_k) \leq f(x^*) + o(\|x_k - x^*\|^2). \tag{12.6.31}$$

It follows from the above relation and (12.6.23) that

$$P_\sigma(x_k + d_k + \hat{d}_k) \leq P_\sigma(x^*) + o(\|x_k - x^*\|^2). \tag{12.6.32}$$

Under the second-order sufficient condition, there exists a positive constant $\delta > 0$ such that

$$P_\sigma(x_k) \geq P_\sigma(x^*) + \delta \|x_k - x^*\|^2. \tag{12.6.33}$$

Therefore, by the above two inequalities we can deduce that

$$P_\sigma(x_k + d_k + \hat{d}_k) < P_\sigma(x_k). \tag{12.6.34}$$

To be more exact, from (12.6.32)-(12.6.33) we can show that

$$\lim_{k \to \infty} \frac{P_\sigma(x_k) - P_\sigma(x_k + d_k + \hat{d}_k)}{P_\sigma(x_k) - P_\sigma(x^*)} = 1. \tag{12.6.35}$$

Therefore, it follows from (12.6.22) and (12.6.34), that

$$\lim_{k \to \infty} \frac{\|x_k + d_k + \hat{d}_k - x^*\|}{\|x_k - x^*\|} = 0, \tag{12.6.36}$$

namely, $d_k + \hat{d}_k$ is a superlinearly convergent step and it is acceptable.

Another way to compute a second-order correction step is to solve the following subproblem:

$$\min_{d \in \Re^n} \quad \tilde{g}_k^T d + \frac{1}{2} d^T B_k d, \tag{12.6.37}$$

$$\text{s.t.} \quad c_i(x_k) + a_i(x_k)^T d = 0, \quad i \in E, \tag{12.6.38}$$

$$c_i(x_k) + a_i(x_k)^T d \geq 0, \quad i \in I, \tag{12.6.39}$$

where

$$\tilde{g}_k = g_k + \frac{1}{2} \sum_{i=1}^{m} (\lambda_k)_i [\nabla c_i(x_k) - \nabla c_i(x_k + d_k)], \tag{12.6.40}$$

and where λ_k is the Lagrange multiplier of the quadratic programming subproblem (12.2.1)-(12.2.3). It can be shown that the search direction defined by (12.6.37)-(12.6.39) is a superlinearly convergent step and is also an acceptable step. For more detailed discussions, please see Mayne and Polak [214] and Fukushima [142].

12.7 Smooth Exact Penalty Functions

The reason for the Maratos Effect to happen is because the merit function used to carry out line search is nonsmooth. If $P(x)$ is a smooth function, if x^* is its minimizer, and if $\nabla^2 P(x^*)$ is positive definite, we can easily see that, for all x sufficiently close to x^*,

$$\bar{M} \|x - x^*\|^2 \geq P(x) - P(x^*) \geq \bar{m} \|x - x^*\|^2, \tag{12.7.1}$$

where $\bar{M} \geq \bar{m}$ are two positive constants. Therefore if

$$\frac{\|x_k + d_k - x^*\|}{\|x_k - x^*\|} \to 0, \tag{12.7.2}$$

it is easy to show that

$$\begin{aligned} P(x_k + d_k) &\leq P(x^*) + \bar{M} \|x_k + d_k - x^*\|^2 \\ &< P(x^*) + \bar{m} \|x_k - x^*\|^2 \leq P(x_k) \end{aligned} \tag{12.7.3}$$

holds for sufficiently large k. Therefore, the Maratos Effect can be avoided if we use a smooth exact penalty function as the merit function.

Consider the equality constrained optimization problem:

$$\min_{x\in\Re^n} \quad f(x), \tag{12.7.4}$$

$$\text{s.t.} \quad c(x) = 0. \tag{12.7.5}$$

We use Fletcher's smooth exact penalty function (10.5.4) as the merit function. Because the derivative of function (10.5.4) needs to compute the second-order derivatives of $f(x)$ and $c(x)$, Powell and Yuan [277] uses an approximate form of (10.5.4):

$$\begin{aligned}\Phi_{k,i}(\alpha\beta_{k,i}) &= f(x_k+\alpha\beta_{k,i}d_k)\\ &- [\lambda(x_k)+\alpha(\lambda(x_k+\beta_{k,i}d_k)-\lambda(x_k))]^T c(x_k+\alpha\beta_{k,i}d_k)\\ &+ \frac{1}{2}\sigma_{k,i}\|c(x_k+\alpha\beta_{k,i}d_k)\|_2^2, \quad 0\le\alpha\le 1,\end{aligned} \tag{12.7.6}$$

where d_k is a solution of the quadratic programming subproblem (12.2.1)-(12.2.3), $\beta_{k,i}$ is the $(i+1)$-th trial step length in the k-th iteration, and $\sigma_{k,i}$ is the current penalty parameter which satisfies that

$$\begin{aligned}\Phi'_{k,i}(0) &\le -\frac{1}{2}[d_k^T B_k d_k + \sigma_{k,i}\|c(x_k)\|_2^2]\\ &\le -\frac{1}{4}\sigma_{k,i}\|c(x_k)\|_2^2.\end{aligned} \tag{12.7.7}$$

The Powell and Yuan's method can be stated as follows:

Algorithm 12.7.1 *(Powell and Yuan's Method)*

Step 1. *Given $x_1\in\Re^n$, $\beta_1\in(0,1)$, $\beta_2\in(\beta_1,1)$, $\mu\in(0,1/2)$, $\sigma_{1,-1}>0$, $B_1\in\Re^{n\times n}$, $\epsilon\ge 0$. $k:=1$;*

Step 2. *Solve (12.2.1)-(12.2.3), giving d_k;*
if $\|d_k\|\le\epsilon$ then stop;
let $i=0$, $\beta_{k,0}=1$;

Step 3. *Choose $\sigma_{k,i}$ such that (12.7.7) holds; if*

$$\Phi_{k,i}(\beta_{k,i}) \le \Phi_{k,i}(0) + \mu\beta_{k,i}\Phi'_{k,i}(0), \tag{12.7.8}$$

then go to Step 4.
$i:=i+1$, $\beta_{k,i}\in[\beta_1,\beta_2]\beta_{k,i-1}$; go to Step 3;

Step 4. $x_{k+1} = x_k + \beta_{k,i} d_k$; $\sigma_{k+1,-1} = \sigma_{k,i}$; *update* B_{k+1}; $k := k + 1$; *go to Step 2.*

For the above algorithm, it can be shown that the following lemma holds.

Lemma 12.7.2 *Assume that* $\{x_k\}$, $\{d_k\}$, $\{B_k\}$ *are bounded. If* $A(x) = \nabla c(x)^T$ *is full column rank for all* $x \in \Re^n$ *and if there exists a constant* $\delta > 0$ *such that*

$$d^T B_k d \geq \delta \|d\|_2^2, \quad \forall A(x_k)^T d = 0 \tag{12.7.9}$$

holds for all k, *then there exists a positive integer* k' *such that*

$$\sigma_{k,i} = \sigma_{k',0} = \bar{\sigma} > 0 \tag{12.7.10}$$

for all $k \geq k'$ *and that*

$$\lim_{k \to \infty} \|d_k\| = 0. \tag{12.7.11}$$

Using this lemma, we can prove the global convergence result of Algorithm 12.7.1

Theorem 12.7.3 *Under the conditions of Lemma 12.7.2, any accumulation point of* $\{x_k\}$ *generated by Algorithm 12.7.1 is a KKT point of (12.7.4)-(12.7.5).*

Now we show that when the iterates are close to a solution, any superlinearly convergent step will be accepted by Algorithm 12.7.1.

Lemma 12.7.4 *Suppose that the assumptions of Lemma 12.7.2 are satisfied, and assume that the sequence* $\{x_k\}$ *generated by Algorithm 12.7.1 converges to* x^*. *For any subsequence* $\{k_i, i = 1, 2, \cdots\}$, *if*

$$\|x_{k_i} + d_{k_i} - x^*\| = o(\|x_{k_i} - x^*\|), \quad k_i \to \infty, \tag{12.7.12}$$

then we have that

$$x_{k_i+1} = x_{k_i} + d_{k_i} \tag{12.7.13}$$

for all large i.

Proof. Without loss of generality, we assume that $k_i \geq k'$. For simplicity of notation, we substitute k_i by j. From the descriptions of the algorithm, we only need to show that

$$\Phi_{j,0}(1) - \Phi_{j,0}(0) - \mu\Phi'_{j,0}(0) < 0. \tag{12.7.14}$$

It follows from (12.7.10) that

$$\Phi_{j,0}(1) = f(x_j + d_j) - \lambda(x_j + d_j)^T c(x_j + d_j) + \frac{1}{2}\bar{\sigma}\|c(x_j + d_j)\|_2^2. \tag{12.7.15}$$

Because $f(x)$ is twice continuously differentiable, we have that

$$\begin{aligned} f(x_j + d_j) &= f(x_j) + \frac{1}{2}d_j^T[g_j + g(x_j + d_j)] + o(\|d_j\|_2^2) \\ &= f(x_j) + \frac{1}{2}d_j^T[g_j + g(x^*)] + o(\|d_j\|_2^2). \end{aligned} \tag{12.7.16}$$

Also, we can obtain similar formulae as (12.7.16) for $c_i(x_j + d_j)$. Substituting all these formulae into (12.7.15), we obtain that

$$\begin{aligned} \Phi_{j,0}(1) - \Phi_{j,0}(0) &= \frac{1}{2}d_j^T[g_j + g(x^*)] \\ &\quad -\lambda(x_j + d_j)^T\left[c_j + \frac{1}{2}A_j^T d_j + \frac{1}{2}A(x^*)^T d_j\right] \\ &\quad -\left[-\lambda_j^T c_j + \frac{1}{2}\bar{\sigma}\|c_j\|_2^2\right] + o(\|d_j\|_2^2) \\ &= \frac{1}{2}\Phi'_{j,0}(0) + \frac{1}{2}d_j^T[g(x^*) - A(x^*)\lambda(x_j + d_j)] \\ &\quad +o(\|d_j)\|_2^2) = \frac{1}{2}\Phi'_{j,0}(0) + o(\|d_j\|_2^2). \end{aligned} \tag{12.7.17}$$

It is not difficult to show there exists a positive constant $\bar{\eta}$ such that

$$\Phi'_{k,i}(0) \leq -\bar{\eta}\|d_k\|_2^2 \tag{12.7.18}$$

holds for all k and i. From (12.7.17), (12.7.18) and $\mu < \frac{1}{2}$, we can see that (12.7.14) holds for sufficiently large $j = k_i$. Thus, the lemma is true. □

A direct corollary of the above result is the superlinear convergence of the algorithm, which we write as follows.

Theorem 12.7.5 *Suppose that the assumptions of Lemma 12.7.2 are satisfied, and assume that the sequence $\{x_k\}$ generated by Algorithm 12.7.1 converges to x^*. If*

$$\lim_{k\to\infty} \frac{\|x_k + d_k - x^*\|}{\|x_k - x^*\|} = 0, \tag{12.7.19}$$

then we have that $x_{k+1} = x_k + d_k$ for all sufficiently large k, which implies that $\{x_k\}$ superlinearly converges to x^*.

12.8 Reduced Hessian Matrix Method

The reduced Hessian matrix method was also developed from the Lagrange-Newton method. A fundamental idea of the reduced Hessian matrix method is that only part of the Hessian matrix of the Lagrangian function is used so that the method requires less storage and computing costs in each iteration.

Consider the equality constrained problem (12.1.1) and (12.1.2). Denoting the Lagrange-Newton step by $(d_k, (\delta\lambda)_k)$, it follows from (12.1.6) that

$$\begin{bmatrix} W(x_k, \lambda_k) & -A(x_k) \\ -A(x_k)^T & 0 \end{bmatrix} \begin{bmatrix} d_k \\ (\delta\lambda)_k \end{bmatrix} = - \begin{bmatrix} \nabla f(x_k) - A(x_k)\lambda_k \\ -c(x_k) \end{bmatrix}. \tag{12.8.1}$$

Using the notations

$$\begin{aligned} W_k &= W(x_k, \lambda_k), && (12.8.2)\\ A_k &= A(x_k) = \nabla c(x_k)^T, && (12.8.3)\\ g_k &= \nabla f(x_k), && (12.8.4)\\ c_k &= c(x_k), && (12.8.5)\\ \hat{\lambda}_k &= \lambda_k + (\delta\lambda)_k, && (12.8.6) \end{aligned}$$

we can rewrite (12.8.1) as

$$\begin{bmatrix} W_k & -A_k \\ -A_k^T & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \hat{\lambda}_k \end{bmatrix} = \begin{bmatrix} -g_k \\ c_k \end{bmatrix}. \tag{12.8.7}$$

Let the QR factorization of A_k be

$$A_k = [Y_k \ Z_k] \begin{bmatrix} R_k \\ 0 \end{bmatrix}, \tag{12.8.8}$$

then linear system (12.8.7) can be written in the following form:

$$\begin{bmatrix} Y_k^T W_k Y_k & Y_k^T W_k Z_k & -R_k \\ Z_k^T W_k Y_k & Z_k^T W_k Z_k & 0 \\ -R_k^T & 0 & 0 \end{bmatrix} \begin{bmatrix} p_k \\ q_k \\ \hat{\lambda}_k \end{bmatrix} = \begin{bmatrix} -Y_k^T g_k \\ -Z_k^T g_k \\ c_k \end{bmatrix}, \tag{12.8.9}$$

where

$$p_k = Y_k^T d_k, \tag{12.8.10}$$
$$q_k = Z_k^T d_k. \tag{12.8.11}$$

It is obvious that p_k and q_k are the projections of d_k to the range space of A_k^T and the null space of A_k^T. Because (12.8.9) has a block triangle form, we can easily solve p_k, q_k and $\hat{\lambda}_k$ in turns:

$$R_k^T p_k = -c_k, \tag{12.8.12}$$
$$(Z_k^T W_k Z_k) q_k = -Z_k^T g_k - Z_k^T W_k Y_k p_k, \tag{12.8.13}$$
$$R_k \hat{\lambda}_k = Y_k^T g_k + Y_k^T W_k (Y_k p_k + Z_k q_k). \tag{12.8.14}$$

If we consider only the last two lines in the linear system (12.8.9), we obtain a linear system independent of λ:

$$\begin{bmatrix} Z_k^T W_k Y_k & Z_k^T W_k Z_k \\ -R_k^T & 0 \end{bmatrix} \begin{bmatrix} p_k \\ q_k \end{bmatrix} = \begin{bmatrix} -Z_k^T g_k \\ c_k \end{bmatrix}, \tag{12.8.15}$$

which is essentially

$$\begin{bmatrix} Z_k^T W_k \\ -A_k^T \end{bmatrix} d_k = \begin{bmatrix} -Z_k^T g_k \\ c_k \end{bmatrix}. \tag{12.8.16}$$

Nocedal and Overton [232] suggests that the matrix $Z_k^T W_k$ be replaced by a quasi-Newton matrix B_k, namely at each iteration the line search direction d_k is obtained by solving the linear system

$$\begin{bmatrix} B_k \\ -A_k^T \end{bmatrix} d = \begin{bmatrix} -Z_k^T g_k \\ c_k \end{bmatrix}, \tag{12.8.17}$$

where $B_k \in \Re^{(n-m)\times n}$ is an approximation to $Z_k^T W_k$. We can apply Broyden's nonsymmetric rank-one formula to update B_k, that is

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k) s_k^T}{s_k^T s_k}, \tag{12.8.18}$$

where

$$s_k = x_{k+1} - x_k, \tag{12.8.19}$$
$$y_k = Z_{k+1}^T g_{k+1} - Z_k^T g_k. \tag{12.8.20}$$

Because $Z_k^T W_k$ is a one-side reduced Hessian matrix, this method is also called the one-side reduced Hessian matrix method. Under certain conditions, Nocedal and Overton [232] proved that the one-side reduced Hessian matrix method using (12.8.18)-(12.8.20) is locally superlinearly convergent.

If we use a symmetric matrix $B_k \in \Re^{(n-m)\times(n-m)}$ to substitute for $Z_k^T W_k Z_k$, and a zero matrix to replace $Z_k^T W_k Y_k$, we can see that (12.8.15) yields that

$$\begin{bmatrix} 0 & B_k \\ -R_k^T & 0 \end{bmatrix} \begin{bmatrix} p_k \\ q_k \end{bmatrix} = \begin{bmatrix} -Z_k^T g_k \\ c_k \end{bmatrix}. \tag{12.8.21}$$

One reason for doing so is a fact discovered by Powell [274] that the SQP method converges 2-step Q-superlinearly

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_{k-1} - x^*\|} = 0 \tag{12.8.22}$$

provided $Y_k^T W_k Z_k$ is bounded. Another reason is that when all iteration points are feasible, we have $p_k = 0$, the value of $Z_k^T W_k Y_k$ does not alter q_k. For linearly constrained problems, all iteration points $x_k (k \geq k_0)$ are feasible if the initial point x_{k_0} is feasible. An advantage of updating $Z_k^T W_k Z_k$ instead of $Z_k^T W_k$ is that $Z_k^T W_k Z_k$ is a square matrix and it is symmetric positive definite near the solution where the second-order sufficient conditions hold. Therefore, we can use positive definite matrices to approximate it, such as the BFGS update:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}, \tag{12.8.23}$$

where

$$s_k = Z_k^T (x_{k+1} - x_k), \tag{12.8.24}$$
$$y_k = Z_{k+1}^T g_{k+1} - Z_k^T g_k. \tag{12.8.25}$$

We can write (12.8.21) in the equivalent form

$$\begin{bmatrix} B_k Z_k^T \\ -A_k^T \end{bmatrix} d_k = \begin{bmatrix} -Z_k^T g_k \\ c_k \end{bmatrix}. \tag{12.8.26}$$

Because the matrix B_k tries to approximate $Z_k^T W_k Z_k$, which is a two-side reduced Hessian matrix, the method using search direction d_k defined by (12.8.26) is called the two-side reduced Hessian matrix method. Such a method is two-step superlinearly convergent near the solution.

Theorem 12.8.1 *Let d_k be defined by (12.8.26). If $x_{k+1} = x_k + d_k$, $x_k \to x^*$, $A(x^*)$ is full column rank, second-order sufficient conditions hold at x^*, and $\|B_k^{-1}\|$ is bounded uniformly and satisfies*

$$\lim_{k\to\infty} \frac{\|[B_k - Z(x^*)^T W(x^*, \lambda^*) Z(x^*)] Z_k^T d_k\|}{\|d_k\|} = 0, \tag{12.8.27}$$

then the sequence converges 2-step Q-superlinearly:

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_{k-1} - x^*\|} = 0. \tag{12.8.28}$$

Proof. It follows from (12.8.26) that

$$\begin{aligned} B_k Z_k^T d_k &= -Z_k^T g_k = -Z_k^T [g_k - A_k \lambda^*] \\ &= -Z_k^T W(x^*, \lambda^*)(x_k - x^*) + O(\|x_k - x^*\|^2). \end{aligned} \tag{12.8.29}$$

Thus, we have that

$$\begin{aligned} [B_k &- Z(x^*)^T W(x^*, \lambda^*) Z(x^*)] Z_k^T d_k \\ &= -Z_k^T W(x^*, \lambda^*)(x_k - x^*) - Z(x^*)^T W(x^*, \lambda^*) d_k \\ &\quad + O(\|x_k - x^*\|^2) + O(\|Y(x^*)^T d_k\|) + o(\|d_k\|) \\ &= -Z_k^T W(x^*, \lambda^*)(x_k + d_k - x^*) \\ &\quad + O(\|x_k - x^*\|^2 + \|Y(x^*)^T d_k\|) + o(\|d_k\|). \end{aligned} \tag{12.8.30}$$

Therefore, based on the assumption (12.8.27), it follows from the above inequality that

$$Z_k^T W(x^*, \lambda^*)(x_k + d_k - x^*) = o(\|x_k - x^*\| + \|d_k\|) + O(\|Y(x^*)^T d_k\|). \tag{12.8.31}$$

The definition of d_k implies that

$$A_k^T (x_k + d_k - x^*) = O(\|x_k - x^*\|^2). \tag{12.8.32}$$

Because $A(x^*)$ is full column rank, we have that

$$\|Y(x^*)^T d_k\| = O(\|c(x_k)\|) = O(\|d_{k-1}\|^2). \tag{12.8.33}$$

Using (12.8.31) and (12.8.32), we see that

$$\begin{bmatrix} Z_k^T W(x^*, \lambda^*) \\ A_k^T \end{bmatrix} (x_k + d_k - x^*) = o(\|x_k - x^*\| + \|d_k\|) + o(\|d_{k-1}\|). \quad (12.8.34)$$

Observing the assumption that B_k^{-1} is uniformly bounded, and using (12.8.26), we can show that

$$\|d_k\| = O(\|x_k - x^*\|), \quad (12.8.35)$$

which indicates that $\|x_k - x^*\| \leq \|x_{k-1} - x^*\| + \|d_{k-1}\| = O(\|x_{k-1} - x^*\|)$. Thus, it follows from (12.8.34) that

$$\begin{bmatrix} Z_k^T W(x^*, \lambda^*) \\ A_k^T \end{bmatrix} (x_k + d_k - x^*) = o(\|x_{k-1} - x^*\|). \quad (12.8.36)$$

Similar to (12.3.19), we can prove that the matrix

$$\begin{bmatrix} Z(x^*)^T W(x^*, \lambda^*) \\ A(x^*)^T \end{bmatrix} \quad (12.8.37)$$

is nonsingular. Therefore, it follows from (12.8.36) that

$$\|x_k + d_k - x^*\| = o(\|x_{k-1} - x^*\|), \quad (12.8.38)$$

which shows that the theorem is true. □

The 2-step superlinearly convergence result of the two-side reduced Hessian matrix method can not be improved. In fact, an example given by Yuan [370] shows that it is possible to show that

$$\|x_{2k+1} - x^*\|_\infty = \|x_{2k} - x^*\|_\infty, \quad (12.8.39)$$
$$\|x_{2k+2} - x^*\|_\infty = \|x_{2k+1} - x^*\|_\infty^2, \quad (12.8.40)$$

which reveals the "one-step fast one-step slow" behaviour of the two-side reduced Hessian matrix method, and it shows that it is impossible to establish a one-step Q-superlinearly convergence result. A similar example was also given by Byrd [40] independently.

Exercises

1. Use the Lagrange-Newton method to solve Rosenbrock's problem:

$$\begin{aligned} \min \quad & (1 - x_1)^2 \\ \text{s.t.} \quad & x_2 - x_1^2 = 0 \end{aligned}$$

with initial point $(0.8, 0.6)^T$ and $\lambda = 1.0$. Give the first three iterations.

2. The damped BFGS update (12.2.39) uses $\bar{y}_k$ which is a linear combination of y_k and $B_k s_k$. Consider the case generating $\bar{y}_k$ by linear combinations of y_k and s_k. What are the advantages and disadvantages of using s_k instead of $B_k s_k$?

3. Prove Corollary 12.3.4.

4. If the QP subproblem in a SQP method is infeasible, one way to overcome this difficulty is to consider the subproblem

$$\begin{array}{ll} \min_{d\in\Re^n,\theta\in[0,1]} & g_k^T d + \frac{1}{2} d^T B_k d + \sigma(1-\theta)^2 \\ \text{s.t.} & a_i(x_k)^T d + \theta c_i(x_k) = 0, \quad i \in E, \\ & a_i(x_k)^T d + \theta c_i(x_k) \geq 0, \quad i \in I. \end{array}$$

Let $(d(\sigma), \theta(\sigma)$ be the solution of the above QP subproblem. Prove that $\theta(\sigma)$ is non-decreasing as σ increases. Discuss the case when $\theta(\sigma) = 0$ for all $\sigma > 0$.

5. Consider application of the SQP method to the following problem:

$$\begin{array}{ll} \min & -x_1 + 10(x_1^2 + x_2^2) \\ \text{s.t.} & x_1^2 + x_2^2 = 1. \end{array}$$

Give the point $\bar{x} = (\cos(\theta), \sin(\theta))^T$ and calculate d by solving the QP subproblem with $B = I$. Assume θ is very small and show that

$$P_\sigma(\bar{x}) < P_\sigma(\bar{x} + d)$$

for any $\sigma \geq 0$, where $P_\sigma(x)$ is the L_1 exact penalty function. Calculate a second-order correction step $\hat{d}$ and verify that

$$P_\sigma(\bar{x}) > P_\sigma(\bar{x} + d + \hat{d}).$$

6. Prove that the Watchdog Technique (Algorithm 12.5.1) can overcome the Maratos Effect.

7. Apply the two-sided reduced Hessian method to the following problem:

$$\begin{aligned}\min \quad & \frac{1}{2}x_2^2 - x_1x_2 + \frac{1}{6(1-x_2)^3}\Big[-4(x_2-x_1)^3 - 6(x_2-x_1)^2(x_1-x_2^2) \\ & -12(x_2-x_1)(x_1-x_2^2)^2 - 17(x_1-x_2^2)^3 + 3\frac{(x_1-x_2^2)^4}{1-x_2}\Big] \\ \text{s.t.} \quad & x_1 + \frac{1}{(1-x_2)^2}[(x_2-x_1)^2 + (x_2-x_1)(x_1-x_2^2) + 2(x_1-x_2^2)^2] = 0\end{aligned}$$

with initial point (ϵ, ϵ), where $\epsilon > 0$ is a very small positive number. You will find the iterates converge to the solution $(0, 0)$ in the one-fast-one-slow pattern.

Chapter 13

Trust-Region Methods for Constrained Problems

13.1 Introduction

Trust-region methods for unconstrained optimization have been discussed in Chapter 6. In this chapter we consider trust-region methods for constrained optimization.

The essential of a trust-region method is that the trial step is within a trust-region. Unlike line search methods where line searches are carried out along a search direction, trust-region algorithms compute a trial step d_k which satisfies

$$\|d_k\| \leq \Delta_k, \tag{13.1.1}$$

where $\Delta_k > 0$ is the trust-region bound at the k-th iteration, and $\|\cdot\|$ is some norm in $\Re^n$. The fundamental belief is that the increment to the variables should not be too large and that it seems not to be a wise idea to search along a not-so-good direction (for example when step of one is not accepted in the search direction). For unconstrained optimization, a line search type algorithm normally obtains its search direction by minimizing an approximation model (for example, a quadratic model in a quasi-Newton method). Minimizing the same approximation model with the trust-region constraint:

$$\|d\| \leq \Delta_k \tag{13.1.2}$$

would give a trial step in the trust-region. Therefore it is obvious that almost all line search algorithms for unconstrained optimization can be modified to

derive corresponding trust-region algorithms.

Unfortunately, the situation for constrained cases are not the same. First, it is easy to see that we can not transform a line search algorithm for constrained optimization into a trust-region algorithm by simply adding a trust-region constraint (13.1.2) to the subproblem of a line search algorithm. Because the subproblems of most line search algorithms for constrained optimization have linear or quadratic constraints, which are approximations to the original constraints, these linear or quadratic constraints may not be consistent with the trust-region condition. For example, if the line search algorithm we have in mind is the Wilson-Han-Powell method discussed in Section 12.2, the undesirable situation is that the linearized constraints (12.2.2)–(12.2.3) may have no solutions in the trust-region (13.1.2). To overcome this infeasibility difficulty, some special considerations have to be made. There are mainly three approaches, which lead to three different types of trust-region subproblems.

The first approach is to scale the constraint violations:

$$\theta_k c_i(x_k) + d^T \nabla c_i(x_k) = 0 \qquad i = 1, 2, \ldots, m_e; \tag{13.1.3}$$

$$\theta_k c_i(x_k) + d^T \nabla c_i(x_k) \geq 0 \qquad i = m_e + 1, \ldots, m \tag{13.1.4}$$

where $\theta_k \in (0, 1]$ is a parameter (see Byrd, Schnabel and Shultz [48] and Vardi [345]). We can see that a smaller θ_k would have smaller constraint violations for the linearized constraints (13.1.3)–(13.1.4), which makes it more likely that its feasible set has a nonempty intersection with the trust-region (13.1.2). Geometrically, the parameter θ_k moves all the feasible points of the linearized constraints (12.2.2)–(12.2.3) towards the origin with the fraction of β_k. Trial steps of the trust-region algorithms that apply null space techniques can also be viewed as solutions of (13.1.2)–(13.1.4).

The second approach is replacing all the linearized constraints by a linear squares constraint. Namely, linear constraints (12.2.2)–(12.2.3) are replaced by a single constraint:

$$\sum_{i=1}^{m_e} (c_i(x_k) + d^T \nabla c_i(x_k))^2 + \sum_{i=m_e+1}^{m} \Big(\min(0, c_i(x_k) + d^T \nabla c_i(x_k))\Big)^2 \leq \xi_k \tag{13.1.5}$$

where $\xi_k \geq 0$ is a parameter. It can be seen that if $\xi_k = 0$, the single constraint on piece-wise linear squares is equivalent to the original linearized constraints (12.2.2)–(12.2.3). The parameter ξ_k should be chosen in such a

way that the constraint (13.1.5) has a nonempty intersection with the trust-region ball (13.1.2).

The third way to overcome the inconsistency of the linearized constraints and the trust-region constraint is replacing the linearized constraints by a penalty term in the subproblem. This approach is essentially applying a trust-region algorithm for nonsmooth optimization to the corresponding penalty function.

A giant monograph on trust-region methods was published by Conn, Gould and Toint [70].

13.2 Linear Constraints

In this section we give a trust-region algorithm for linearly constrained optimization problems. The method uses trust-region conditions to define trial steps and forces all iteration points in the feasible set. The method can be considered as a combination of the feasible point method and a trust-region technique.

Consider the linearly constrained problem

$$\min_{x\in\Re^n} \quad f(x) \tag{13.2.1}$$

$$\text{s. t.} \quad a_i^T x = b_i, \quad i \in E, \tag{13.2.2}$$

$$a_i^T x \geq b_i, \quad i \in I. \tag{13.2.3}$$

Assume that the current iterate point x_k at the k-th iteration is feasible. The trust-region subproblem can be defined by

$$\min_{d\in\Re^n} \quad g_k^T d + \frac{1}{2} d^T B_k d \triangleq \phi_k(d), \tag{13.2.4}$$

$$\text{s.t.} \quad a_i^T d = 0, \ \ i \in E, \tag{13.2.5}$$

$$a_i^T (x_k + d) \geq b_i, \ i \in I, \tag{13.2.6}$$

$$\|d\|_\infty \leq \Delta_k. \tag{13.2.7}$$

It is easy to see that (13.2.4)–(13.2.7) is a quadratic programming problem, which can be solved by methods discussed in Chapter 9. Let d_k be a solution of (13.2.4)–(13.2.7). Define the ratio of actual reduction and predicted reduction by

$$r_k = \frac{f(x_k) - f(x_k + d_k)}{\phi_k(0) - \phi_k(d_k)}. \tag{13.2.8}$$

From the definition of d_k, we can easily see that $d_k = 0$ if and only if x_k is a KKT point of the original problem (13.2.1)–(13.2.3). Because all constraints are considered in the subproblem (13.2.4)–(13.2.7), the zigzagging can not happen. The following is the statement of a trust-region algorithm, assuming that the initial point x_1 is feasible.

Algorithm 13.2.1

Step 1. *Given x_1 satisfying (13.2.2)–(13.2.3); given $B_1 \in \Re^{n\times n}$, $\Delta_1 > 0$, $\varepsilon \geq 0$, $k := 1$.*

Step 2. *Solve (13.2.4)–(13.2.7) giving d_k; if $\|d_k\| \leq \varepsilon$ then stop; Compute (13.2.8);*

$$x_{k+1} = \begin{cases} x_k + d_k, & \text{if} \quad r_k > 0 \\ x_k, & \text{Otherwise.} \end{cases} \tag{13.2.9}$$

Step 3. *If $r_k \geq 0.25$, go to Step 4;*
$\Delta_k := \Delta_k/2$, go to Step 5.

Step 4. *If $r_k < 0.75$ or $\|d_k\|_\infty < \Delta_k$ then go to Step 5;*
$\Delta_k := 2\Delta_k$.

Step 5. *$\Delta_{k+1} := \Delta_k$; Generate B_{k+1};*
$k := k + 1$; go to Step 2.

The matrix B_{k+1} can be updated by quasi-Newton formulae. In the convergence analyses below, we assume that $\{B_k\}$ are uniformly bounded. Namely, there exists a positive constant M such that

$$\|B_k\| \leq M \tag{13.2.10}$$

holds for all k.

Theorem 13.2.2 *Assume that $f(x)$ is continuously differentiable on the feasible set and that (13.2.10) holds. If the sequence $\{x_k\}$ generated by Algorithm 13.2.1 has accumulation points, then there exists an accumulation point which is also a KKT point of the original constrained optimization problem (13.2.1)–(13.2.3).*

Proof. If the theorem is not true, we can show that

$$\lim_{k\to\infty} \Delta_k = 0. \tag{13.2.11}$$

If the above relation does not hold, there exists a positive constant $\delta > 0$, such that

$$\Delta_k \geq \delta \quad \text{and} \quad r_k \geq 0.25 \tag{13.2.12}$$

hold for infinitely many k. Define by K_0 the set of all k such that (13.2.12) hold. Without loss of generality, we assume that

$$\lim_{\substack{k\in K_0 \\ k\to\infty}} x_k = \bar{x}. \tag{13.2.13}$$

From our assumption, $\bar{x}$ is not a KKT point of (13.2.1)–(13.2.2), thus $d = 0$ is not a solution of

$$\begin{aligned} \min \quad & g(\bar{x})^T d + \frac{M}{2}\|d\|_2^2 && (13.2.14)\\ \text{s.t.} \quad & a_i^T d = 0, \ \ i \in E, && (13.2.15)\\ & a_i^T(\bar{x} + d) \geq 0, \ i \in I, && (13.2.16)\\ & \|d\|_\infty \leq \delta/2. && (13.2.17) \end{aligned}$$

Let $\bar{d}$ be a solution of (13.2.14)–(13.2.17), then

$$\eta = g(\bar{x})^T \bar{d} + \frac{1}{2} M \|\bar{d}\|_2^2 < 0. \tag{13.2.18}$$

Thus, it follows from (13.2.12), (13.2.13) and (13.2.18) that

$$\phi_k(0) - \phi_k(d_k) \geq -\frac{1}{2}\eta > 0 \tag{13.2.19}$$

holds for all sufficiently large $k \in K_0$. Using (13.2.19) and the second inequality of (13.2.12) we can see that

$$f(x_k) - f(x_{k+1}) \geq -\frac{1}{8}\eta > 0 \tag{13.2.20}$$

holds for all sufficiently large $k \in K_0$. Because $\lim_{k\to\infty} f(x_k) = f(\bar{x})$, (13.2.20) can not hold for infinitely many k. This contradiction indicates that (13.2.11) must hold if the theorem is not true.

Now we suppose that the theorem is not true. The above analyses imply that (13.2.11) holds. There exists a subsequence K_1 such that

$$r_k < 0.25, \ \forall k \in K_1. \tag{13.2.21}$$

Assume that

$$\lim_{\substack{k \in K_1 \\ k \to \infty}} x_k = \hat{x}. \tag{13.2.22}$$

From our assumption, $\hat{x}$ is not a KKT point. Let $\hat{d}$ be a solution of the subproblem

$$\begin{aligned} \min_{d \in \Re^n} \quad & g(\hat{x})^T d + \frac{1}{2} M \|d\|_2^2, && (13.2.23) \\ \text{s.t.} \quad & a_i^T d = 0, \ \ i \in E, && (13.2.24) \\ & a_i^T (\hat{x} + d) \geq b_i, \ i \in I, && (13.2.25) \\ & \|d\|_\infty \leq 1, && (13.2.26) \end{aligned}$$

then we have that

$$g(\hat{x})^T \hat{d} + \frac{M}{2} \|\hat{d}\|_2^2 = \hat{\eta} < 0. \tag{13.2.27}$$

Thus, because $(\Delta_k \hat{d})$ is a feasible point of the problem

$$\begin{aligned} \min\nolimits_{d \in \Re^n} \quad & g(\hat{x})^T d + \frac{1}{2} M \|d\|_2^2, && (13.2.28) \\ \text{s.t.} \quad & a_i^T d = 0, \ i \in E, && (13.2.29) \\ & a_i^T (\hat{x} + d) \geq b_i, \ i \in I, && (13.2.30) \\ & \|d\|_\infty \leq \Delta_k, && (13.2.31) \end{aligned}$$

we can see that

$$g(\hat{x})^T \hat{d}_k + \frac{1}{2} M \|\hat{d}_k\|_2^2 < \Delta_k \hat{\eta}, \tag{13.2.32}$$

provided that $\Delta_k \leq 1$, where $\hat{d}_k$ is a solution of (13.2.28)–(13.2.31). It follows from (13.2.22) and (13.2.32) that

$$\phi_k(0) - \phi_k(d_k) \geq -\frac{1}{2} \hat{\eta} \Delta_k \tag{13.2.33}$$

holds for all sufficiently large $k \in K_1$. From the continuously differentiable property of $f(x)$ and the uniform boundedness of $\{B_k\}$, we have that

$$Pred_k = Ared_k + o(\|d_k\|). \tag{13.2.34}$$

It can be shown from (13.2.33) and (13.2.34) that

$$\lim_{\substack{k\in K_1\\ k\to\infty}} r_k = 1. \tag{13.2.35}$$

This contradicts (12.2.21). Therefore the theorem is true. □

Similar to our analysis of the trust-region method for unconstrained optimization, Theorem 13.2.2 is still true if the condition (13.2.10) is replaced by

$$\sum_{k=1}^{\infty} \frac{1}{1+\max_{1\le i\le k}\|B_i\|} = +\infty. \tag{13.2.36}$$

From the proof of the above theorem, we can see that it is not necessary to require the trial step d_k to be the exact solution of (13.2.4)–(13.2.7). Define the projected gradient of $f(x)$ (with respect to the feasible set X) by

$$\nabla_X f(x) = \lim_{\alpha\to 0_+} \frac{P(x-\alpha\nabla f(x))-x}{\alpha}, \tag{13.2.37}$$

where

$$P(y) = \arg\min\{\|z-y\|, z\in X\}.$$

It is not difficult to show that x^* is a KKT point of (13.2.1)–(13.2.3) if and only if

$$\nabla_X f(x^*) = 0. \tag{13.2.38}$$

From the proof of Theorem 13.2.2, we can see that Algorithm 13.2.1 remains globally convergent provided that d_k is a feasible point of (13.2.5)–(13.2.7) and satisfies

$$\phi_k(0)-\phi_k(d_k) \ge \bar{\delta}\|\nabla_X f(x_k)\| \min\left\{\Delta_k, \frac{\|\nabla_X f(x_k)\|}{\|B_k\|}\right\}. \tag{13.2.39}$$

As for local convergence analysis, we assume that $x_k \to x^*$ and that there are only equality constraints. We also assume that the second-order sufficient conditions hold at x^* and that the Jacobian matrix $A(x^*) = \nabla c(x^*)^T \in \Re^{n\times m}$ has full column rank. Under these conditions, it is not difficult to show that Algorithm 13.2.1 is superlinearly convergent, namely

$$\lim_{k\to\infty} \frac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|} = 0 \tag{13.2.40}$$

if and only if

$$\lim_{k\to\infty} \frac{\|Z^{*T}(B_k - [\nabla^2 f(x^*) - \sum \lambda_i^* \nabla^2 c_i(x^*)])Z^* Z^{*T} d_k\|}{\|d_k\|} = 0, \qquad (13.2.41)$$

where $Z^* \in \Re^{n\times(n-m)}$ is a matrix satisfying $Z^{*T}A(x^*) = 0$ and $Z^{*T}Z^* = I$.

13.3 Trust-Region Subproblems

The key part of a trust-region algorithm is the calculation of the trial step d_k, which is normally a solution of a trust-region subproblem. Therefore the crucial issue of a trust-region algorithm is the construction of the trust-region subproblem. Because one of the most successful line search type methods is the sequential quadratic programming method, it is natural to consider the combination of quadratic models and trust-region technique. The combined method is usually called the TR-SQP (Trust-Region – Sequential Quadratic Programming) method. Since a trust-region constraint has the form

$$\|d\| \le \Delta_k, \qquad (13.3.1)$$

directly amalgamating (13.3.1) and the quadratic programming subproblem (12.2.1)–(12.2.3) of the sequential quadratic programming method gives the following subproblem:

$$\begin{aligned} \min_{d\in\Re^n} \quad & g_k^T d + \frac{1}{2} d^T B_k d \triangleq \phi_k(d), && (13.3.2)\\ \text{s.t.} \quad & c_i(x_k) + a_i(x_k)^T d = 0, i \in E && (13.3.3)\\ & c_i(x_k) + a_i(x_k)^T d \ge 0, i \in I && (13.3.4)\\ & \|d\| \le \Delta_k. && (13.3.5) \end{aligned}$$

This is not a perfect way, as the constraints (13.3.3)–(13.3.5) might have no solutions. Therefore, the subproblem (13.3.2)–(13.3.5) has to be modified in order to derive a reasonable trust-region subproblem for constrained optimization.

First, we can consider a subproblem of the following type:

$$\begin{aligned} \min_{d\in\Re^n} \quad & g_k^T d + \frac{1}{2} d^T B_k d \triangleq \phi_k(d), && (13.3.6)\\ \text{s.t.} \quad & \theta_k c_i(x_k) + d^T \nabla c_i(x_k) = 0,\ i \in E, && (13.3.7)\\ & \theta_k c_i(x_k) + d^T \nabla c_i(x_k) \ge 0, i \in I, && (13.3.8)\\ & \|d\| \le \Delta_k, && (13.3.9) \end{aligned}$$

where $\theta_k \in (0, 1]$ is a parameter. Subproblem (13.3.7)–(13.3.9) usually has feasible points when θ_k is sufficiently small. Geometrically, multiplying $c_i(x_k)$ by a factor θ_k is to pull the feasible points of the linearized constraints towards the original. In other words, the role of θ_k is to shift the line corresponding to the linearized constraints to a parallel line that intersects the trust-region. This technique of introducing a parameter had already been used in line search algorithms.

If $\theta_k \neq 1$, obviously the trial step d_k obtained by solving subproblem (13.3.6)–(13.3.9) may not be a feasible point of (13.3.3) and (13.3.4). In order to force d_k to be as feasible in the sense of (13.3.3)–(13.3.4) as possible, we should choose θ_k as close to 1 as possible. On the other hand, the larger the parameter θ_k, the smaller the feasible set of (13.3.7)–(13.3.9). To allow certain freedom to the subproblem, we should not choose a too large θ_k.

The minimum-norm solution of the problem

$$\min_{d \in \Re^n} \|(c(x_k) + A(x_k)^T d)^{(-)}\|_2 \tag{13.3.10}$$

is called the Gauss-Newton step, which is denoted by d_k^{GN}. Here $c^{(-)}$ is defined by (10.1.5)–(10.1.6). From the definition of d_k^{GN}, (13.3.7)–(13.3.9) is feasible if and only if

$$\theta_k \|d_k^{GN}\| \leq \Delta_k. \tag{13.3.11}$$

To avoid unnecessary small θ_k, it is reasonable to require

$$\theta_k \|d_k^{GN}\| \geq \delta_1 \Delta_k \tag{13.3.12}$$

if $\theta_k < 1$, where $\delta_1 \in (0, 1)$ is a given constant. For example, we can define θ_k by the formula

$$\theta_k = \begin{cases} 1, & \text{if} \quad 2\|d_k^{GN}\| \leq \Delta_k, \\ \frac{1}{2}\Delta_k / \|d_k^{GN}\|, & \text{otherwise.} \end{cases} \tag{13.3.13}$$

An indirect way to choose the parameter θ_k is regarding $\theta = \theta_k$ as a variable. The idea of forcing θ as large as possible is achieved by a penalty term $\sigma(\theta - 1)^2$. The subproblem can be written as

$$\min_{d \in \Re^n, \theta \in (0,1]} \quad g_k^T d + \frac{1}{2} d^T B_k d + \sigma_k (\theta - 1)^2, \tag{13.3.14}$$

$$\text{s.t.} \quad \theta c_i(x_k) + d^T \nabla c_i(x_k) = 0, \ i \in E, \tag{13.3.15}$$

$$\theta c_i(x_k) + d^T \nabla c_i(x_k) \geq 0, \ i \in I, \tag{13.3.16}$$

$$\|d\| \leq \Delta_k, \tag{13.3.17}$$

where $\sigma_k > 0$ is a penalty parameter.

Another method to overcome the inconsistency of (13.3.3)–(13.3.5) is replacing (13.3.3)–(13.3.4) by a single constraint, which requires the sum of the squares of all linearized constraints being bounded by a certain bound:

$$\|(c_k + A_k^T d)^{(-)}\|_2^2 \leq \xi_k, \tag{13.3.18}$$

where $c_k = c(x_k) = (c_1(x_k), \cdots, c_m(x_k))^T$, $A_k = A(x_k) = \nabla c(x_k)^T$, and $\xi_k > 0$ is a parameter. Thus, the subproblem has the form

$$\min_{d \in \Re^n} \quad g_k^T d + \frac{1}{2} d^T B_k d, \tag{13.3.19}$$

$$\text{s.t.} \quad \|(c_k + A_k^T d)^{(-)}\|_2^2 \leq \xi_k, \tag{13.3.20}$$

$$\|d\|_2^2 \leq \Delta_k^2. \tag{13.3.21}$$

It is easy to see that ξ_k must satisfy

$$\xi_k \geq \min_{\|d\|_2 \leq \Delta_k} \|(c_k + A_k^T d)^{(-)}\|_2^2, \tag{13.3.22}$$

in order to secure the feasibility of (13.3.20)–(13.3.21). Let $\bar{d}_k$ be the negative gradient direction of the function $\|(c_k + A_k^T d)^{(-)}\|_2^2$ at $d = 0$, namely $\bar{d}_k = -A_k c_k^{(-)}$, and let $\bar{\alpha}_k > 0$ be the solution of problem

$$\min_{\substack{\alpha > 0 \\ \|\alpha \bar{d}_k\|_2 \leq \Delta_k}} \|(c_k + A_k^T \alpha \bar{d}_k)^{(-)}\|_2^2. \tag{13.3.23}$$

We call $\bar{\alpha}_k \bar{d}_k$ the Cauchy point or the Cauchy step, which is denoted by d_k^{CP}. In the method of Celis, Dennis and Tapia [52],

$$\xi_k = \|(c_k + A_k^T d_k^{CP})^{(-)}\|_2^2, \tag{13.3.24}$$

while in Powell and Yuan [278], ξ_k can be any number satisfying

$$\min_{\|d\|_2 \leq b_1 \Delta_k} \|(c_k + A_k^T d)^{(-)}\|_2^2 \leq \xi_k \leq \min_{\|d\|_2 \leq b_2 \Delta_k} \|(c_k + A_k^T d)^{(-)}\|_2^2, \tag{13.3.25}$$

where $b_1 \geq b_2$ are two positive constants in (0, 1).

The third type of trust-region subproblem is based on exact penalty functions. For example, based on the exact penalty function

$$P(x, \sigma) = f(x) + \sigma \|c^{(-)}(x)\|, \tag{13.3.26}$$

we can construct trust-region subproblem

$$\min_{d\in\Re^n} \quad g_k^T d + \frac{1}{2} d^T B_k d + \sigma_k \|(c_k + A_k^T d)^{(-)}\|, \tag{13.3.27}$$

$$\text{s.t.} \quad \|d\| \le \Delta_k. \tag{13.3.28}$$

For this kind of subproblems, the norm in (13.3.27) and that in (13.3.28) may not be necessarily the same. For example, if we take the l_1-norm in (13.3.27) and l_∞-norm in (13.3.28), we obtain the subproblem

$$\min_{d\in\Re^n} \quad g_k^T d + \frac{1}{2} d^T B_k d + \sigma_k \sum_{i\in E} |c_i(x_k) + \nabla c_i(x_k)^T d| + \sigma_k \sum_{i\in I} |c_i(x_k) + \nabla c_i(x_k)^T d|^{(-)} \tag{13.3.29}$$

$$\text{s.t.} \quad \|d\|_\infty \le \Delta_k. \tag{13.3.30}$$

Essentially, a trust-region algorithm based on subproblem (13.2.27)–(13.2.28) is the same as a nonsmooth trust-region algorithm for minimizing the exact penalty function (13.3.26).

13.4 Null Space Method

Consider the equality constrained problem

$$\min\nolimits_{x\in\Re^n} \quad f(x), \tag{13.4.1}$$

$$\text{s.t.} \quad c(x) = 0. \tag{13.4.2}$$

The trust-region subproblem (13.3.6)–(13.3.9) can be written as

$$\min\nolimits_{d\in\Re^n} \quad g_k^T d + \frac{1}{2} d^T B_k d = \phi_k(d), \tag{13.4.3}$$

$$\text{s.t.} \quad \theta_k c_k + A_k^T d = 0, \tag{13.4.4}$$

$$\|d\|_2 \le \Delta_k. \tag{13.4.5}$$

We assume that $c_k \in$ Range (A_k^T), it follows from (13.3.11) that θ_k should satisfy

$$\theta_k \|(A_k^T)^+ c_k\|_2 \le \Delta_k. \tag{13.4.6}$$

Let d_k be a solution of (13.4.3)–(13.4.5). It can be seen that d_k is also a solution of the following problem

$$\begin{aligned} \min_{d\in\Re^n} \quad & \phi_k(d), && (13.4.7)\\ \text{s.t.} \quad & A_k^T(d-\hat{d}_k)=0, && (13.4.8)\\ & \|d-\hat{d}_k\|_2 \le \bar{\Delta}_k, && (13.4.9) \end{aligned}$$

where

$$\begin{aligned} \hat{d}_k &= -\theta_k (A_k^T)^+ c_k, && (13.4.10)\\ \bar{\Delta}_k &= \sqrt{\Delta_k^2 - \|\hat{d}_k\|_2^2}. && (13.4.11) \end{aligned}$$

Notice that $\hat{d}_k = \theta_k d_k^{GN}$ where d_k^{GN} is the Gauss-Newton step discussed in the previous section. Define variable $\bar{d} = d - \hat{d}_k$ and let Z_k be a matrix whose columns are an orthonormal base of the null space of A_k^T, namely $A_k^T Z_k = 0$, $Z_k^T Z_k = I$. We can then write

$$\bar{d} = Z_k u,\ u \in \Re^{n-r}, \tag{13.4.12}$$

where r is the rank of A_k. Using the above relation, we can rewrite subproblem (13.4.7)–(13.4.9) in the following equivalent form:

$$\begin{aligned} \min_{u\in\Re^{n-r}} \quad & \bar{g}_k^T u + \frac{1}{2} u^T \bar{B}_k u, && (13.4.13)\\ \text{s.t.} \quad & \|u\|_2 \le \bar{\Delta}_k, && (13.4.14) \end{aligned}$$

where $\bar{g}_k = Z_k^T(g_k + B_k\hat{d}_k)$, $\bar{B}_k = Z_k^T B_k Z_k$. This is already in the form of the trust-region subproblem for unconstrained optimization, which is discussed in Chapter 6. Techniques given there can be used to solve problem (13.4.13)–(13.4.14), giving u_k. Once u_k is computed, the trial step d_k can be obtained by using $d_k = \hat{d}_k + Z_k u_k$.

We use the L_1 exact penalty function

$$P_1(x) = f(x) + \sigma_k \|c(x)\|_1 \tag{13.4.15}$$

as the merit function to decide whether the trial step d_k should be accepted. The actual reduction of the exact penalty function is

$$Ared_k = P_1(x_k) - P_1(x_k + d_k). \tag{13.4.16}$$

We define the predicted reduction by the reduction of the approximate penalty function $\phi_k(d) + \sigma_k \|c_k + A_k^T d\|_1$, namely,

$$Pred_k = \phi_k(0) - \phi_k(d_k) + \sigma_k[\|c_k\|_1 - \|c_k + A_k^T d_k\|_1]. \tag{13.4.17}$$

Assume that $f(x)$ and $c(x)$ are twice continuously differentiable and $\|B_k\|$ is bounded, then we have that

$$Ared_k = Pred_k + O(\|d_k\|_2^2). \tag{13.4.18}$$

From the definition of $\hat{d}_k$, it follows that

$$\hat{d}_k = (A_k^T)^+ A_k^T d_k, \tag{13.4.19}$$

$$d_k - \hat{d}_k = Z_k Z_k^T d_k = (I - (A_k^T)^+ A_k^T) d_k. \tag{13.4.20}$$

The step $\hat{d}_k$ is a vector in the range space of A_k, hence it is called the range space step. While the step $d_k - \hat{d}_k$ is in the null space of A_k^T, it is called the null space step. Geometrically, it is often that the range space step is vertical and the null space step is horizontal when we sketch an illustrated diagram (for example, see Figure 13.4.1). Therefore, the range space step and the null space step are called the vertical step and the horizontal step respectively.

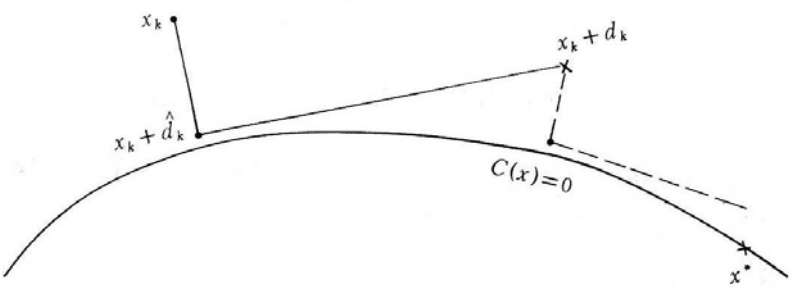


Figure 13.4.1

Using the vertical step and the horizontal step, we can decompose the predicted reduction into two parts:

$$Vpred_k = \phi_k(0) - \phi_k(\hat{d}_k) + \sigma_k(\|c_k\|_1 - \|c_k + A_k^T \hat{d}_k\|_1), \tag{13.4.21}$$

$$Hpred_k = \phi_k(\hat{d}_k) - \phi_k(d_k). \tag{13.4.22}$$

We can assume that θ_k satisfies the "not too small" condition:

$$\theta_k \|(A_k^T)^+ c_k\|_2 \geq \delta_1 \Delta_k, \text{ if } \theta_k < 1. \tag{13.4.23}$$

Assume again that we choose a sufficiently large σ_k so that

$$\sigma_k \geq \left\| A_k^+ \left(g_k + \frac{1}{2} B_k \hat{d}_k \right) \right\|_\infty + \rho. \tag{13.4.24}$$

With all these, we can show that

$$Vpred_k \geq \rho \min[\|c_k\|_1, \delta_1 \Delta_k / \|(A_k^T)^+\|_2]. \tag{13.4.25}$$

For the null space, the situation is essentially the same as that of unconstrained optimization. Applying Lemma 6.1.3, we have that

$$Hpred_k \geq \frac{1}{2} \|\bar{g}_k\|_2 \min[\bar{\Delta}_k, \|\bar{g}_k\|_2 / \|\bar{B}_k\|_2]. \tag{13.4.26}$$

Thus, we have established that there exist positive constants ρ_1, ρ_2 such that

$$\begin{aligned} Pred_k \quad \geq \quad & \rho_1 \min[\|c_k\|_1, \Delta_k / \|(A_k^T)^+\|_2] \\ & + \rho_2 \|\bar{g}_k\|_2 \min[\bar{\Delta}_k, \|\bar{g}_k\|_2 / \|\bar{B}_k\|_2]. \end{aligned} \tag{13.4.27}$$

In practice, we can first compute $\hat{d}_k$ using the Gauss-Newton step, and then obtain u_k by solving (13.4.13)–(13.4.14) inexactly. The vector $d_k = \hat{d}_k + Z_k u_k$ satisfies the sufficient reduction condition (13.4.27).

The following is a trust-region algorithm based on null space technique.

Algorithm 13.4.1

Step 1. *Given* $x_1 \in \Re^n$, $\Delta_1 > 0$, $\epsilon \geq 0$.
$0 < \beta_3 < \beta_4 < 1 < \beta_1$, $0 \leq \beta_0 \leq \beta_2 < 1$,
$\beta_2 > 0, \sigma_1 > 0, k := 1$;

Step 2. *If* $\|c_k\|_2 + \|Z_k^T g_k\|_2 \leq \epsilon$ *then stop;*
If (13.4.24) is satisfied then go to Step 3. Set

$$\sigma_k = \left\| A_k^+ \left(g_k + \frac{1}{2} B_k \hat{d}_k \right) \right\| + 2\rho;$$

Step 3. *Compute a trial step* d_k *satisfying (13.4.27).*

Step 4. Compute $Ared_k$ and $Pred_k$ by (13.4.16)–(13.4.17); Set $r_k = Ared_k/Pred_k$;

$$x_{k+1} = \begin{cases} x_k + d_k, & \text{if } r_k > \beta_0, \\ x_k, & \text{otherwise.} \end{cases} \tag{13.4.28}$$

Choose Δ_{k+1} such that

$$\Delta_{k+1} \in \begin{cases} (\beta_3\|d_k\|_2,\ \beta_4\Delta_k), & \text{if } r_k < \beta_2, \\ (\Delta_k, \beta_1\Delta_k), & \text{otherwise.} \end{cases} \tag{13.4.29}$$

Step 5. Generate B_{k+1}; Set $\sigma_{k+1} := \sigma_k$ and $k := k + 1$; go to Step 2.

In order to establish the global convergence of the above algorithm, we need the following lemma.

Lemma 13.4.2 *If d_k satisfies (13.4.27), the inequality*

$$Pred_k \geq \tau \min[\varepsilon_k, 1] \min[\Delta_k, \varepsilon_k/(1 + \|B_k\|_2)] \tag{13.4.30}$$

holds, where $\tau = \min[\rho_1/[2\max(1, \|A_k^+\|_2)], \rho_2/4]$ and

$$\varepsilon_k = \|c_k\|_2 + \|Z_k^T g_k\|_2. \tag{13.4.31}$$

Proof. If

$$\|c_k\|_2 > \frac{\Delta_k}{2\|A_k^+\|_2}, \tag{13.4.32}$$

it follows directly from (13.4.27) that

$$Pred_k \geq \rho_1\Delta_k/2\|A_k^+\|_2. \tag{13.4.33}$$

Thus we see that (13.4.30) holds.

Therefore, for the rest of the proof we can assume that (13.4.32) is not true. This implies that

$$\bar{\Delta}_k = \sqrt{\Delta_k^2 - \|\hat{d}_k\|_2^2} \geq \sqrt{\Delta_k^2 - (\|A_k^+\|_2\|c_k\|_2^2)^2} \geq \frac{1}{2}\Delta_k. \tag{13.4.34}$$

If

$$(1 + 2\|B_k\|_2\|A_k^+\|_2)\|c_k\|_2 \leq \|Z_k^T g_k\|_2, \tag{13.4.35}$$

we can show that

$$\begin{aligned} \|\bar{g}_k\|_2 &\geq \|Z_k^T g_k\|_2 - \|B_k\|_2 \|\hat{d}_k\|_2 \\ &\geq \|Z_k^T g_k\|_2 - \|B_k\|_2 \|A_k^+\|_2 \|c_k\|_2 \\ &\geq \frac{1}{2}(\|Z_k^T g_k\|_2 + \|c_k\|_2) = \frac{1}{2}\varepsilon_k. \end{aligned} \tag{13.4.36}$$

This inequality, (13.4.34) and (13.4.27) indicate that (13.4.30) holds when $\tau = \rho_2/4$.

Now, we assume that inequality (13.4.35) does not hold, which implies that

$$\|c_k\|_2 \geq \frac{\varepsilon_k}{2(1 + \|B_k\|_2 \|A_k^+\|_2)}. \tag{13.4.37}$$

Consequently, we can use (13.4.27) to show that

$$\begin{aligned} Pred_k &\geq \rho_1 \min[\varepsilon_k/[2(1 + \|B_k\|_2 \|A_k^+\|_2)],\ \Delta_k/\|A_k^+\|_2] \\ &\geq \frac{\rho_1}{2\max[1, \|A_k^+\|_2]} \min[\Delta_k, \varepsilon_k/(1 + \|B_k\|_2)], \end{aligned} \tag{13.4.38}$$

which says that (13.4.30) holds when $\tau = \rho_1/(2\max[1, \|A_k^+\|_2])$. □

The following lemma says that the norm of the trial step can not converge to zero faster than the reciprocal of the norm of quasi-Newton matrices B_k, if the iteration points are bounded away from KKT points.

Lemma 13.4.3 *Assume that $f(x)$ and $c(x)$ are twice continuously differentiable and that $\{x_k, k = 1, 2, ...\}$ are generated by Algorithm 13.4.1. If $\|A_k\|_2$ is bounded above uniformly and*

$$\|c_k\|_2 + \|Z_k^T g_k\|_2 \geq \delta > 0 \tag{13.4.39}$$

for all k, then there exists a positive constant β_5 such that

$$\|d_k\|_2 \geq \beta_5/M_k,\ k = 1, 2, \cdots \tag{13.4.40}$$

holds for all k, where

$$M_k = \max_{1 \leq i \leq k} \|B_i\|_2 + 1. \tag{13.4.41}$$

Proof. If $\|d_k\|_2 < \Delta_k$, we can easily see that

$$\|\bar{d}_k\|_2 < \bar{\Delta}_k. \tag{13.4.42}$$

The definition of $\bar{d}_k$ and the above inequality imply that

$$\bar{g}_k + \bar{B}_k \bar{d}_k = 0. \tag{13.4.43}$$

Thus, we have

$$\|\bar{d}_k\|_2 \geq \frac{\|\bar{g}_k\|_2}{\|\bar{B}_k\|_2} \geq \frac{\|Z_k^T g_k\|_2}{\|B_k\|_2} \geq \frac{\|Z_k^T g_k\|_2}{M_k}. \tag{13.4.44}$$

From the definition of $\hat{d}_k$, we see that

$$\|\hat{d}_k\|_2 \geq \min\left[\delta_1 \Delta_k, \ \frac{\|c_k\|_2}{\|A_k\|_2}\right]. \tag{13.4.45}$$

Relations (13.4.44) and (13.4.45) indicate that either

$$\|d_k\|_2 \geq \delta_1 \Delta_k, \tag{13.4.46}$$

or

$$\|d_k\|_2 \geq \frac{1}{2(1 + \|A_k\|_2)} \frac{\|c_k\|_2 + \|Z_k^T g_k\|_2}{M_k}. \tag{13.4.47}$$

The boundedness of $\|A_k\|$ shows that (13.4.40) holds if (13.4.46) fails.

For the rest of the proof, we assume that (13.4.46) holds. If the lemma is not true there exists a subsequence $\{k_i\}$ such that

$$\|d_{k_i}\|_2 \geq \delta_1 \Delta_{k_i} \tag{13.4.48}$$

and

$$\lim_{i \to \infty} \Delta_{k_i} M_{k_i} = 0. \tag{13.4.49}$$

A direct consequence of the above limit is $\Delta_{k_i} \to 0$. Because M_k is monotonically increasing, we can assume that $\Delta_{k_i} < \Delta_{k_i - 1}$ for all i. Denote $\bar{i} = k_i - 1$, (13.4.29) implies that $\|d_{\bar{i}}\|_2 \geq \Delta_{k_i}/\beta_3$, which, together with (13.4.49) and $M_{\bar{i}} \leq M_{k_i}$, shows that

$$\lim_{i \to \infty} \|d_{\bar{i}}\| M_{\bar{i}} = 0. \tag{13.4.50}$$

This shows that

$$\|d_{\bar{i}}\| \geq \delta_1 \Delta_{\bar{i}}. \tag{13.4.51}$$

This inequality guarantees the existence of a positive number $\bar{\tau}$ such that

$$Pred_{\bar{i}} \geq \bar{\tau} \|d_{\bar{i}}\|_2 \tag{13.4.52}$$

for sufficiently large i. This inequality and (13.4.18) show that

$$\lim_{i\to\infty} r_{k_i-1} = \lim_{i\to\infty} \frac{Ared_{\bar{i}}}{Pred_{\bar{i}}} = 1. \tag{13.4.53}$$

Therefore, for sufficient large i, we have that

$$\Delta_{k_i} \geq \Delta_{k_i-1}. \tag{13.4.54}$$

This contradicts the assumption that $\Delta_{k_i-1} < \Delta_{k_i}$. Thus, the lemma is true. □

The following lemma is due to Powell [264], which is a very powerful tool for convergence analysis of trust-region algorithms.

Lemma 13.4.4 *Suppose that $\{\Delta_k\}$ and $\{M_k\}$ are two sequences of positive numbers. If there exist positive constants $\tau > 0$, $\beta_1 > 0$, $\beta_4 \in (0,1)$, and a subset I of $\{1,2,3,\cdots\}$ such that*

$$\begin{aligned} \Delta_{k+1} &\leq \beta_1\Delta_k, && \forall k \in I; \\ \Delta_{k+1} &\leq \beta_4\Delta_k, && \forall k \notin I; \\ \Delta_k &\geq \tau/M_k, && \forall k; \\ M_{k+1} &\geq M_k, && \forall k; \\ \textstyle\sum_{k\in I} 1/M_k &< +\infty, \end{aligned} \tag{13.4.55}$$

then

$$\sum_{k=1}^{\infty} \frac{1}{M_k} < +\infty. \tag{13.4.56}$$

Proof. Let p be a positive integer satisfying

$$\beta_1 \cdot \beta_4^{p-1} < 1. \tag{13.4.57}$$

Define the set

$$I_k = I \cap \{1,2,\cdots,k\}. \tag{13.4.58}$$

Denote the number of elements of I_k by $|I_k|$. Define the set

$$J := \{k \mid k \leq p|I_k|\}. \tag{13.4.59}$$

From the monotone property of M_k and the above definition, we have that

$$\sum_{k\in J} \frac{1}{M_k} \leq p \sum_{k\in I} \frac{1}{M_k} < +\infty. \tag{13.4.60}$$

For $k \notin J$, we have that $|I_k| < k/p$, which gives $|I_{k-1}| \leq |I_k| \leq (k-1)/p$. Thus,

$$\begin{aligned} \Delta_k &\leq \beta_1^{|I_{k-1}|}\beta_4^{k-1-|I_{k-1}|}\Delta_1 \\ &\leq (\beta_1\beta_4^{p-1})^{(k-1)/p}\Delta_1 \end{aligned} \tag{13.4.61}$$

holds for all $k \notin J$. Consequently, we have that

$$\begin{aligned} \sum_{k\notin J}\frac{1}{M_k} &\leq \sum_{k=1}^{\infty}(\beta_1\beta_4^{p-1})^{(k-1)/p}\Delta_1/\tau \\ &= \frac{\Delta_1}{\tau[1-(\beta_1\beta_4^{p-1})^{1/p}]}. \end{aligned} \tag{13.4.62}$$

Now, we can see that (13.4.56) follows from (13.4.60) and (13.4.62). □

Using the above lemmas, we can prove the global convergence of Algorithm 13.4.1.

Theorem 13.4.5 *Assume that $f(x)$ and $c(x)$ are twice continuously differentiable, that all the iteration points $\{x_k\}$ generated by Algorithm 13.4.1 are in an open set S, and that $\nabla f(x)$, $\nabla^2 f(x)$, $A(x)$, $\nabla A(x)$ are bounded above on S. If $\sigma_k = \bar{\sigma}$ for all sufficiently large k, $P_1(x_k)$ is bounded below, and $\{\|A_k\|_2, \|A_k^+\|_2\}$ are uniformly bounded, and*

$$\sum_{k=1}^{\infty}\frac{1}{1+\max_{1\leq i\leq k}\|B_i\|_2} = \infty, \tag{13.4.63}$$

then

$$\liminf_{k\to\infty}[\|c_k\|_2 + \|Z_k^T g_k\|_2] = 0. \tag{13.4.64}$$

Furthermore, under additional assumptions that $\|B_k\|_2$ is uniformly bounded and $\beta_0 > 0$, we have that

$$\lim_{k\to\infty}[\|c_k\|_2 + \|Z_k^T g_k\|_2] = 0. \tag{13.4.65}$$

Proof. If the theorem is not true, the sequence $\{P_{\bar{\sigma}}(x_k) = f(x_k) + \bar{\sigma}\|c(x_k)\|_1\}$ is bounded below and there exists a positive constant δ such that (13.4.39) holds for all k. Define the set

$$I = \{k | r_k \geq \beta_2\}, \tag{13.4.66}$$

then it follows from Lemmas 13.4.2 and 13.4.3 that

$$\begin{aligned} +\infty &> \sum_{k=1}^{\infty}[P_{\bar{\sigma}}(x_k) - P_{\bar{\sigma}}(x_{k+1})] \geq \sum_{k\in I}[P_{\bar{\sigma}}(x_k) - P_{\bar{\sigma}}(x_{k+1})] \\ &\geq \sum_{k\in I}\beta_2 Pred_k \geq \sum_{k\in I}\frac{1}{2}\beta_2\delta\min[\Delta_k, \delta/M_k] \\ &\geq \sum_{k\in I}\frac{1}{2}\beta_2\delta\min[\beta_5, \delta]/M_k. \end{aligned} \tag{13.4.67}$$

This inequality and the previous lemma imply that

$$\sum_{k=1}^{\infty}\frac{1}{M_k} < +\infty, \tag{13.4.68}$$

which contracts (13.4.63). Therefore the theorem is true. □

13.5 CDT Subproblem

Consider the subproblem (13.3.19)–(13.3.21) for the case when there are only equality constraints ($m_e = m$). It can be written as

$$\min_{d\in\Re^n} \quad g^T d + \frac{1}{2}d^T B d = \phi(d), \tag{13.5.1}$$

$$\text{s.t.} \quad \|A^T d + c\|_2 \leq \xi, \tag{13.5.2}$$

$$\|d\|_2 \leq \Delta, \tag{13.5.3}$$

here we omit the subscript for convenience. Such a subproblem was proposed by Celis, Dennis and Tapia [52], and is generally called the CDT subproblem.

Obviously, only when

$$\xi \geq \xi_{\min} = \min_{\|d\|_2\leq\Delta} \|A^T d + c\|_2, \tag{13.5.4}$$

there exist feasible points for (13.5.2)–(13.5.3).

First, we consider the case when $\xi = \xi_{\min}$. It is easy to deduce from the convexity of $\|d\|_2$ that either there is only one feasible solution of (13.5.2)–(13.5.3) or that

$$\xi = \xi^*_{\min} = \min_{d\in\Re^n} \|A^T d + c\|_2. \tag{13.5.5}$$

The case when there is only one feasible solution of (13.5.2)–(13.5.3) requires no further consideration since this feasible point must be the solution of the CDT subproblem. This feasible point must have the following form:

$$d = -(AA^T + \lambda I)^+ c, \tag{13.5.6}$$

where $\lambda \geq 0$, and $\lambda = 0$ if $\|d\|_2 < \Delta$. $AA^T + \lambda I$ is nonsingular unless $\|d\|_2 = \Delta$. For the case when (13.5.5) holds, we have that

$$\|\hat{d}\|_2 \leq \Delta, \tag{13.5.7}$$

where $\hat{d} = -(A^T)^+ c$ is the minimal norm solution (also called the Gauss-Newton Step). Let Z be a matrix whose columns are a basis of the null space of A^T. By the variable substitution $d = \hat{d} + Zu$ as given in the previous section, problem (13.5.1)–(13.5.3) can be transformed as

$$\min_{u \in \Re^{\bar{n}}} \quad \bar{g}^T u + \frac{1}{2} u^T \bar{B} u, \tag{13.5.8}$$

$$\text{s.t.} \qquad \|u\|_2 \leq \bar{\Delta}, \tag{13.5.9}$$

which is already in the form of the trust-region subproblem for unconstrained optimization discussed in Chapter 6.

Therefore in this section, we concentrate our attention on the case when

$$\xi > \xi_{\min}. \tag{13.5.10}$$

First we have the following necessary result.

Theorem 13.5.1 *Let d^* be a global solution of the subproblem (13.5.1)–(13.5.3). Assume that (13.5.10) holds. Then there exist nonnegative constants λ^*, μ^* such that*

$$(B + \lambda^* I + \mu^* AA^T) d^* = -(g + \mu^* Ac), \tag{13.5.11}$$

where λ^* and μ^* satisfy the complementarity conditions

$$\lambda^* [\Delta - \|d^*\|_2] = 0, \tag{13.5.12}$$

$$\mu^* [\xi - \|c + A^T d^*\|_2] = 0. \tag{13.5.13}$$

Furthermore, the matrix

$$H(\lambda^*, \mu^*) = B + \lambda^* I + \mu^* AA^T \tag{13.5.14}$$

has at most one negative eigenvalue if the multipliers λ^* and μ^* are unique.

Proof. Assumption (13.5.10) implies that the feasible region X of (13.5.2)–(13.5.3) is convex and has a nonempty interior, and we can easily prove that $LFD(d^*, X) = SFD(d^*, X)$. From the results in Chapter 8, there exist nonnegative numbers λ^* and μ^* such that (13.5.11)–(13.5.13) hold. To complete the proof, we only need to prove that the matrix $H(\lambda^*, \mu^*)$ has no more than one negative eigenvalue if the multipliers λ^*, μ^* are unique.

If at most one of the constraints (13.5.2)–(13.5.3) is active at the solution d^*, the second-order sufficient condition given in Chapter 8 shows that the matrix $H(\lambda^*, \mu^*)$ has at most one negative eigenvalue.

For the rest of the proof, we assume that both constraints are active. Define the vector

$$y^* = A(c + A^T d^*). \tag{13.5.15}$$

If d^* and y^* are linearly dependent, there exists $\eta \in \Re$ such that

$$y^* = \eta d^*. \tag{13.5.16}$$

The assumption $\xi > \xi_{\min}$ implies $\eta > 0$. It follows from the uniqueness of λ^* and μ^* that $\lambda^* = \mu^* = 0$. Thus, d^* is a stationary point of $\phi(d)$. From (13.5.16) and $\eta > 0$ we see that d is a feasible direction provided that $d^T d^* < 0$. This shows that $d^T B d \geq 0$ holds for all d satisfying $d^T d^* \leq 0$, which implies that B is a semi-definite matrix.

If d^* and y^* are linearly independent, the second-order necessary condition shows that the matrix $H(\lambda^*, \mu^*)$ is positive semi-definite in the $n-2$ dimensional subspace orthogonal to d^* and y^*. Assume that $H(\lambda^*, \mu^*)$ has two negative eigenvalues, then there exist linearly independent vectors $z_1, z_2 \in \Re^n$ such that $H(\lambda^*, \mu^*)$ is negative definite on $Span(z_1, z_2)$. The intersection of $Span(z_1, z_2)$ and the $n-2$ dimensional subspace mentioned above is empty except the original. Therefore the matrix

$$\begin{pmatrix} z_1^T d^* & z_2^T d^* \\ z_1^T y^* & z_2^T y^* \end{pmatrix} \tag{13.5.17}$$

is nonsingular. The nonsingularity of the above matrix implies the existence of a nonzero vector $\bar{d} \in \text{Span}(z_1, z_2)$ such that

$$\|d^* + \bar{d}\|_2 = \Delta, \ \|c + A^T(d^* + \bar{d})\|_2 = \xi. \tag{13.5.18}$$

Relation (13.5.18) and the negative definiteness of $H(\lambda^*, \mu^*)$ on $\text{Span}(z_1, z_2)$ give that $\phi(d^* + \bar{d}) < \phi(d^*)$. This contradicts the optimality of d^*. Hence the lemma is true. □

The following is a sufficient condition.

Theorem 13.5.2 *Let d^* be a feasible point of (13.5.2)–(13.5.3). If there exist $\lambda^* \geq 0$ and $\mu^* \geq 0$ such that (13.5.11)–(13.5.13) hold, and that $H(\lambda^*, \mu^*)$ is positive semi-definite, then d^* is a global solution of (13.5.1)–(13.5.13).*

Proof. Let d be any vector satisfying (13.5.2)–(13.5.3). We have that

$$\begin{aligned}
\phi(d) &= \phi(d) + \frac{1}{2}\lambda^*\|d\|_2^2 + \frac{1}{2}\mu^*\|c + A^T d\|_2^2 \\
&\quad -\frac{1}{2}[\lambda^*\|d\|_2^2 + \mu^*\|c + A^T d\|_2^2] \\
&\geq \phi(d^*) + \frac{1}{2}\lambda^*\|d^*\|_2^2 + \frac{1}{2}\mu^*\|c + A^T d^*\|_2^2 \\
&\quad -\frac{1}{2}[\lambda^*\|d\|_2^2 + \mu^*\|c + A^T d\|_2^2] \\
&= \phi(d^*) + \frac{1}{2}\lambda^*[\Delta^2 - \|d\|_2^2] + \frac{1}{2}\mu^*[\xi^2 - \|c + A^T d\|_2^2] \\
&\geq \phi(d^*). \qquad (13.5.19)
\end{aligned}$$

Thus, we can see that d^* is a global solution of (13.5.1)–(13.5.3). □

A direct consequence of the above theorem is the following.

Corollary 13.5.3 *Assume that B is positive semi-definite. A feasible point d^* of (13.5.2)–(13.5.3) is a solution of (13.5.1)–(13.5.3) if and only if there exist $\lambda^* \geq 0$, $\mu^* \geq 0$ such that (13.5.11)-(13.5.13) hold.*

Therefore, when B is positive definite, the solution of (13.5.1)–(13.5.3) must have the form

$$d(\lambda, \mu) = -H(\lambda, \mu)^{-1}[g + \mu A c]. \qquad (13.5.20)$$

From Corollary 13.5.3 we can easily see that the following lemma holds.

Lemma 13.5.4 *Assume that B is positive definite. Then $d(\lambda, \mu)$ defined by (13.5.20) is a solution of (13.5.1)–(13.5.3) if and only if it is a feasible point of (13.5.2)–(13.5.3), and one of the following holds:*

1. $\lambda = \mu = 0$;
2. $\lambda > 0$, $\mu = 0$, $\|d(\lambda, \mu)\|_2 = \Delta$;
3. $\lambda = 0$, $\mu > 0$, $\|c + A^T d(\lambda, \mu)\|_2 = \xi$;

4. $\lambda > 0, \mu > 0$, $\|d(\lambda,\mu)\|_2 = \Delta$, $\|c + A^T d(\lambda,\mu)\|_2 = \xi$.

From the above statements, solving a convex CDT subproblem is equivalent to finding $\lambda^*, \mu^* \geq 0$ such that $d(\lambda^*, \mu^*)$ is feasible and one of the four possibilities in Lemma 13.5.4 holds.

For the case $\lambda^* = \mu^* = 0$, the solution is $d = -B^{-1}g$.

For $\mu^* = 0$ and $\lambda^* > 0$, we can solve $\bar{\psi}(\lambda, 0) = 0$ to obtain λ^*, where

$$\bar{\psi}(\lambda,\mu) = \frac{1}{\|d(\lambda,\mu)\|_2} - \frac{1}{\Delta}. \tag{13.5.21}$$

The reason for considering $\bar{\psi}(\lambda, 0) = 0$ instead of $\|d(\lambda, 0)\|_2 = \Delta$ is similar to that in Chapter 6, namely $\bar{\psi}(\lambda, 0)$ behaves more like a linear function. $\bar{\psi}(\lambda,\mu)$ as a function of λ is concave and increasing, thus we can apply Newton's iteration:

$$\lambda_+ = \lambda - \frac{\bar{\psi}(\lambda, 0)}{\bar{\psi}'_\lambda(\lambda, 0)}. \tag{13.5.22}$$

It is not difficult to show that iteration process (13.5.22) with any initial $\lambda \in [0, \lambda^*]$ will generate a monotone increasing sequence converging to λ^*.

When $\lambda^* = 0$ and $\mu^* > 0$, we define

$$\hat{\psi}(\lambda,\mu) = \frac{1}{\|c + A^T d(\lambda,\mu)\|_2} - \frac{1}{\xi}. \tag{13.5.23}$$

Similarly, we can apply Newton's method to $\hat{\psi}(0, \mu) = 0$, that is,

$$\mu_+ = \mu - \frac{\hat{\psi}(0, \mu)}{\hat{\psi}'_\mu(0, \mu)}. \tag{13.5.24}$$

When $\lambda^* > 0$ and $\mu^* > 0$, we need to solve

$$\bar{\psi}(\lambda,\mu) = 0, \tag{13.5.25}$$

$$\hat{\psi}(\lambda,\mu) = 0. \tag{13.5.26}$$

The Newton iteration for the above system is

$$\begin{pmatrix} \lambda_+ \\ \mu_+ \end{pmatrix} = \begin{pmatrix} \lambda \\ \mu \end{pmatrix} - J(\lambda,\mu)^{-1} \begin{pmatrix} \bar{\psi}(\lambda,\mu) \\ \hat{\psi}(\lambda,\mu) \end{pmatrix}, \tag{13.5.27}$$

where $J(\lambda,\mu)$ is the Jacobi matrix:

$$J(\lambda,\mu) = \begin{bmatrix} \bar{\psi}'_\lambda(\lambda,\mu) & \bar{\psi}'_\mu(\lambda,\mu) \\ \hat{\psi}'_\lambda(\lambda,\mu) & \hat{\psi}'_\mu(\lambda,\mu) \end{bmatrix}. \tag{13.5.28}$$

An algorithm based on the above analyses is given as follows.

Algorithm 13.5.5

Step 1. *Given $g \in \Re^n$, B positive definite, $\Delta > 0$, $\xi > \xi_{\min}$.*

Step 2. *Compute $d(0,0)$. If $d(0,0)$ is feasible then stop; If $\|d(0,0)\| \leq \Delta$ then go to Step 4;*

Step 3. *Applying (13.5.22) to solve $\bar{\psi}(\lambda, 0) = 0$ giving λ^*; If $d(\lambda^*, 0)$ is feasible then stop;*

Step 4 *Applying (13.5.24) to solve $\hat{\psi}(0, \mu) = 0$ giving μ^*; If $d(0, \mu^*)$ is feasible then stop;*

Step 5. *Applying (13.5.27) to solve (13.5.25)–(13.5.26) giving λ^*, μ^*; stop.*

The above algorithm is in fact an enumeration of the four cases given in Lemma 13.5.4. A more direct way is to solve the system

$$\begin{pmatrix} \bar{\psi}(\lambda,\mu) \\ \hat{\psi}(\lambda,\mu) \end{pmatrix} \geq 0, \ (\lambda,\mu)^T \begin{bmatrix} \bar{\psi}(\lambda,\mu) \\ \hat{\psi}(\lambda,\mu) \end{bmatrix} = 0 \tag{13.5.29}$$

in the nonnegative orthant $\Re^2_+ = \{\lambda \geq 0, \mu \geq 0\}$. Such an approach to identify the Lagrange multipliers λ^* and μ^* is equivalent to solving the dual problem of (13.5.1)–(13.5.3). A truncated Newton's method based on the dual of (13.5.1)–(13.5.3) is given by Yuan [373], which is basically the Newton-Raphson method for the nonlinear system (13.5.29). The approach given by Zhang [381] is to reformulate (13.5.29) as a univariate problem. Basically it is to solve the problem

$$\hat{\psi}(\lambda(\mu), \mu) = 0 \tag{13.5.30}$$

where $\lambda(\mu)$ is defined by $\bar{\psi}(\lambda, \mu) = 0$.

13.6 Powell-Yuan Algorithm

Consider the constrained optimization problem (13.4.1)–(13.4.2). The trial step d_k is obtained by solving

$$\min_{d \in \Re^n} \quad g_k^T d + \frac{1}{2} d^T B_k d = \phi_k(d), \tag{13.6.1}$$

$$\text{s. t.} \quad \|c_k + A_k^T d\|_2 \leq \xi_k, \tag{13.6.2}$$

$$\|d\|_2 \leq \Delta_k, \tag{13.6.3}$$

where Δ_k is the trust-region radius, ξ_k is a parameter satisfying (13.3.25). The merit function is Fletcher's differentiable function:

$$P_k(x) = f(x) - \lambda(x)^T c(x) + \sigma_k \|c(x)\|_2^2, \tag{13.6.4}$$

where $\sigma_k > 0$ is a penalty parameter, $\lambda(x)$ is the minimum norm solution of

$$\min_{\lambda \in \Re^m} \|g(x) - A(x)\lambda\|_2. \tag{13.6.5}$$

The actual reduction is

$$Ared_k = P_k(x_k) - P_k(x_k + d_k), \tag{13.6.6}$$

and the predicted reduction is defined by

$$\begin{aligned} Pred_k &= -(g_k - A_k\lambda_k)^T d_k - \frac{1}{2} d_k^T B_k \bar{d}_k \\ &\quad + [\lambda(x_k + d_k) - \lambda_k]^T \Big(c_k + \frac{1}{2} A_k^T d_k \Big) \\ &\quad + \sigma_k (\|c_k\|_2^2 - \|c_k + A_k^T d_k\|_2^2), \end{aligned} \tag{13.6.7}$$

where $\bar{d}_k$ is the orthogonal projection of d_k to the null space of A_k^T, namely

$$\bar{d}_k = \bar{P}_k d_k, \tag{13.6.8}$$

$$\bar{P}_k = I - A(x_k)A(x_k)^+. \tag{13.6.9}$$

If $\|c_k\|_2 - \|c_k + A_k^T d_k\|_2 > 0$, from (13.6.7) and by increasing σ_k (if needed), we have that

$$Pred_k \geq \frac{1}{2}\sigma_k (\|c_k\|_2^2 - \|c_k + A_k^T d_k\|_2^2). \tag{13.6.10}$$

If $\|c_k\|_2 - \|c_k + A_k^T d_k\|_2 = 0$, d_k is the minimizer of $\phi_k(d)$ in the intersection of the trust-region and the null space of A_k^T and $Pred_k = \phi_k(0) - \phi_k(d_k)$. Thus, $Pred_k = 0$ if and only if $g_k - A_k\lambda_k = 0$.

The following algorithm is given by Powell and Yuan(1991):

Algorithm 13.6.1

Step 1. Given $x_1 \in \Re^n$, $\Delta_1 > 0$, $\epsilon > 0$.
$0 < \tau_3 < \tau_4 < 1 < \tau_1$, $0 \leq \tau_0 \leq \tau_2 < 1$, $\tau_2 > 0$; $k := 1$.

Step 2. If $\|c_k\|_2 + \|g_k - A_k\lambda_k\|_2 \leq \epsilon$ *then Stop. Otherwise solve the problem (13.6.1)-(13.6.3) which gives* d_k;

Step 3. Calculate $Pred_k$ by formula (13.6.7); If (13.6.10) is satisfied then go to Step 4; Set

$$\sigma_k := 2\sigma_k + \max\Big\{0, \frac{-2Pred_k}{\|c_k\|_2^2 - \|c_k + A_k^T d_k\|_2^2}\Big\}. \quad (13.6.11)$$

Recalculate $Pred_k$ by (13.6.7).

Step 4. Calculate the ratio $r_k = Ared_k/Pred_k$. Set the values

$$x_{k+1} = \begin{cases} x_k + d_k, & \text{if } r_k > 0, \\ x_k, & \text{otherwise;} \end{cases} \quad (13.6.12)$$

and

$$\Delta_{k+1} = \begin{cases} \max[4\|d_k\|_2, \Delta_k], & \text{if } r_k > 0.9, \\ \Delta_k, & 0.1 \le r_k \le 0.9, \\ \min[\Delta_k/4, \|d_k\|_2/2], & r_k < 0.1. \end{cases} \quad (13.6.13)$$

Step 5. Generate B_{k+1}. Set $\sigma_{k+1} := \sigma_k$. Set $k := k + 1$ and go to Step 2.

In order to establish the convergence results of the above algorithm, we make the following assumptions.

Assumption 13.6.2

1. *There exists a bounded convex closed set $\Omega \in \Re^n$ such that $\{x_k\}$, $\{x_k + d_k\}$ are all in Ω for all k.*
2. *$A(x)$ has full column rank for all $x \in \Omega$.*
3. *The matrices $\{B_k | k = 1, 2, \cdots\}$ are uniformly bounded.*

The following two lemmas provide a lower bound on the predicted reduction $Pred_k$.

Lemma 13.6.3 *The inequality*

$$\|c_k\|_2 - \|c_k + A_k^T d_k\|_2 \ge \min\Big[\|c_k\|_2, \frac{b_2 \Delta_k}{\|A_k^+\|_2}\Big] \quad (13.6.14)$$

holds for all k, where b_2 is introduced in (13.3.25).

Proof. If $b_2\Delta_k \geq \|(A_k^T)^+ c_k\|_2$, we have $\xi_k = 0$. Thus,

$$\|c_k\|_2 - \|c_k + A_k^T d_k\|_2 = \|c_k\|_2, \tag{13.6.15}$$

which implies (13.6.14).

In the case when $b_2\Delta_k < \|(A_k^T)^+ c_k\|_2$, it follows from (13.3.25) and constraint condition (13.6.2) that

$$\begin{aligned} \|c_k\|_2 &- \|c_k + A_k^T d_k\| \geq \|c_k\|_2 - \xi_k \\ &\geq \|c_k\|_2 - \Big\|c_k - A_k^T\Big[\frac{b_2\Delta_k}{\|(A_k^T)^+ c_k\|_2}\Big](A_k^T)^+ c_k\Big\|_2 \\ &= \|c_k\|_2 \frac{b_2\Delta_k}{\|(A_k^T)^+ c_k\|_2} \geq \frac{b_2\Delta_k}{\|A_k^+\|_2}. \end{aligned} \tag{13.6.16}$$

Thus the lemma is true. □

Lemma 13.6.4 *There exists a positive constant δ_1 such that the inequality*

$$\begin{aligned} Pred_k &- \frac{1}{2}\sigma_k(\|c_k\|_2^2 - \|c_k + A_k^T d_k\|_2^2) + \delta_1\|d_k\|_2\|c_k\|_2 \\ &\geq \frac{1}{4}\|\bar{P}_k\bar{g}_k\|_2 \min\Big[\bar{\Delta}_k, \frac{\|\bar{P}_k\bar{g}_k\|_2}{2\|B_k\|_2}\Big] \\ &\quad + \frac{1}{2}\sigma_k\|c_k\|_2 \min\Big[\|c_k\|_2, \frac{b_2\Delta_k}{\|A_k^+\|_2}\Big] \end{aligned} \tag{13.6.17}$$

holds for all k, where we use the notation

$$\bar{g}_k = g_k + B_k\hat{d}_k, \tag{13.6.18}$$

$$\hat{d}_k = d_k - \bar{P}_k d_k = d_k - \bar{d}_k, \tag{13.6.19}$$

$$\bar{\Delta}_k = \sqrt{\Delta_k^2 - \|\hat{d}_k\|_2^2}. \tag{13.6.20}$$

Proof. The definition of $\hat{d}_k$ and $\|c_k + A_k^T d_k\|_2 \leq \|c_k\|_2$ imply the bound

$$\begin{aligned} \|\hat{d}_k\|_2 &= \|A_k A_k^+ d_k\|_2 = \|(A_k^+)^T[(c_k + A_k^T d_k) - c_k]\|_2 \\ &\leq 2\|A_k^+\|_2\|c_k\|_2. \end{aligned} \tag{13.6.21}$$

From its definition, $\bar{d}_k$ is a solution of the subproblem

$$\min_{d\in\Re^n} \quad \bar{g}_k^T d + \frac{1}{2}d^T B_k d, \tag{13.6.22}$$

$$\text{s. t.} \quad A_k^T d = 0, \tag{13.6.23}$$

$$\|\hat{d}_k + d\|_2 \leq \Delta_k. \tag{13.6.24}$$

It is easy to see that $\bar{d}_k$ also solves the calculation

$$\min_{d\in\Re^n} \quad (\bar{P}_k\bar{g}_k)^T d + \frac{1}{2}(\bar{P}_k d)^T B_k(\bar{P}_k d), \tag{13.6.25}$$

$$\text{s. t.} \quad \|d\|_2 \le \bar{\Delta}_k. \tag{13.6.26}$$

Similar to the proof of Lemma 6.1.3, we can show that

$$\bar{g}_k^T\bar{d}_k \le -\frac{1}{2}\|\bar{P}_k\bar{g}_k\|_2 \min\Big[\bar{\Delta}_k, \frac{\|\bar{P}_k\bar{g}_k\|_2}{2\|B_k\|_2}\Big]. \tag{13.6.27}$$

Hence the definitions of $\lambda_k, \bar{d}_k, \bar{g}_k$, the fact that expression (13.6.25) increases monotonically between $d = \bar{d}_k$ and $d = 0$, and the inequalities (13.6.21) and (13.6.27) imply the bound

$$\begin{aligned}
(g_k - A_k\lambda_k)^T d_k &+ \frac{1}{2}d_k^T B_k d_k = \Big(g_k + \frac{1}{2}B_k d_k\Big)^T \bar{d}_k \\
&= \frac{1}{2}[g_k^T\bar{d}_k + \bar{d}_k^T B_k\bar{d}_k + \bar{g}_k^T\bar{d}_k] \le \frac{1}{2}g_k^T\bar{d}_k \\
&\le \frac{1}{2}\bar{g}_k^T\bar{d}_k + \frac{1}{2}\|B_k\hat{d}_k\|_2\|\bar{d}_k\|_2 \\
&\le -\frac{1}{4}\|\bar{P}_k\bar{g}_k\|_2 \min\Big[\bar{\Delta}_k, \frac{\|\bar{P}_k\bar{g}_k\|_2}{2\|B_k\|_2}\Big] \\
&\quad +\|A_k^+\|_2\|B_k\|_2\|d_k\|_2\|c_k\|_2.
\end{aligned} \tag{13.6.28}$$

Moreover, due to the definition of $\lambda(x)$ and Assumption 13.6.2, there exists a positive constant $\delta_2 > 0$ such that the condition

$$\|\lambda(x_k) - \lambda(x_k + d_k)\|_2 \le \delta_2\|d_k\|_2 \tag{13.6.29}$$

holds for all k. The convexity of $\|c_k + A_k^T d\|_2$ shows that

$$\Big\|c_k + \frac{1}{2}A_k^T d_k\Big\|_2 \le \frac{1}{2}(\|c_k\|_2 + \|c_k + A_k^T d_k\|_2) \le \|c_k\|_2. \tag{13.6.30}$$

Therefore, the inequality (13.6.17) now follows from (13.6.7) and (13.6.14) and (13.6.28)-(13.6.30) if we let $\delta_1 = \delta_2 + \sup_{k\ge1}\{\|B_k\|_2\|A_k^+\|_2\}$, which is finite due to Assumption 13.6.2. □

A direct corollary of the above lemma is that (13.6.10) is satisfied if $\|c_k\|_2/\Delta_k$ is sufficiently small.

Corollary 13.6.5 *There exist positive constants δ_3 and δ_4, such that, on the iterations that satisfy the condition*

$$\|c_k\|_2 \leq \delta_3 \Delta_k, \tag{13.6.31}$$

we have the inequality

$$Pred_k \geq \frac{1}{2}\sigma_k[\|c_k\|_2^2 - \|c_k + A_k^T d_k\|_2^2] + \delta_4 \Delta_k. \tag{13.6.32}$$

Proof. From Assumption 13.6.2, there exists $\bar{M}$ such that $\Delta_k \leq \bar{M}$. If $\delta_3 < \frac{\epsilon}{3\bar{M}}$, (13.6.31) implies $\|c_k\|_2 \leq \epsilon/3$. Unless the algorithm terminates, we have that

$$\|g_k - A_k\lambda_k\|_2 \geq 2\epsilon/3. \tag{13.6.33}$$

If $\delta_3 < \epsilon/(6\bar{M}\sup_k \|B_k\|_2\|A_k^+\|_2)$, (13.6.31) yields

$$\|c_k\|_2 \leq \frac{\epsilon}{6\sup_{1\leq k}\|B_k\|_2\|A_k^+\|_2}, \tag{13.6.34}$$

which implies that

$$\begin{aligned} \|g_k - A_k\lambda_k\|_2 &= \|\bar{P}_k g_k\|_2 \leq \|\bar{P}_k\bar{g}_k\|_2 + \|\bar{P}_k B_k\hat{d}_k\|_2 \\ &\leq \|\bar{P}_k\bar{g}_k\|_2 + 2\|A_k^+\|_2\|B_k\|_2\|c_k\|_2 \\ &\leq \|\bar{P}_k\bar{g}_k\|_2 + \frac{\epsilon}{3}. \end{aligned} \tag{13.6.35}$$

Thus, provided that

$$\delta_3 < \frac{\epsilon}{3\bar{M}}\min\Big[1, \frac{1}{2\sup\|B_k\|_2\|A_k^+\|_2}\Big], \tag{13.6.36}$$

we have, using (13.6.33) and (13.6.35), that

$$\|\bar{P}_k\bar{g}_k\|_2 \geq \frac{\epsilon}{3}. \tag{13.6.37}$$

Consequently, it follows from Lemma 13.6.4 that

$$\begin{aligned} Pred_k &- \frac{1}{2}\sigma_k[\|c_k\|_2^2 - \|c_k + A_k^T d_k\|_2^2] + \delta_1\|d_k\|_2\|c_k\|_2 \\ &\geq \frac{\epsilon}{12}\min\Big[\bar{\Delta}_k, \frac{\epsilon}{6\|B_k\|}\Big]. \end{aligned} \tag{13.6.38}$$

If δ_3 satisfies

$$\delta_3 \leq 0.3/\sup_k \|A_k^+\|, \tag{13.6.39}$$

we have $\bar{\Delta}_k > 0.8\Delta_k$ from (13.6.31). When

$$\delta_3 < \frac{\epsilon}{24\delta_1} \min\Big[\frac{0.8}{\bar{M}}, \frac{\epsilon}{6\bar{M}^2 \sup_k \|B_k\|_2}\Big], \tag{13.6.40}$$

it follows from (13.6.31) that

$$\delta_1 \|c_k\|_2 \|d_k\|_2 \leq \frac{\epsilon}{24} \min\Big[0.8\Delta_k, \frac{\epsilon}{6\|B_k\|_2}\Big]. \tag{13.6.41}$$

Now, inequalities (13.6.38) and (13.6.41) give that

$$\begin{aligned} Pred_k &- \frac{1}{2}\sigma_k[\|c_k\|_2^2 - \|c_k + A_k^T d_k\|_2^2] \\ &\geq \frac{\epsilon}{24} \min\Big[0.8\Delta_k, \frac{\epsilon}{6\|B_k\|}\Big]. \end{aligned} \tag{13.6.42}$$

The corollary follows from the above inequality, and the assumptions that $\{\Delta_k\}$, $\{\|B_k\|\}$ are bounded. □

Now, using the above results, we can easily prove the boundedness of the sequence $\{\sigma_k\}$, which is important in establishing the convergence properties of the algorithm.

Lemma 13.6.6 *The sequence $\{\sigma_k | k = 1, 2, \cdots\}$ remains bounded. In other words, because any increase in σ_k is by at least a factor of 2, there exists $\bar{k}$, such that*

$$\sigma_k = \sigma_{\bar{k}}, \quad \forall k \geq \bar{k}. \tag{13.6.43}$$

Proof. Corollary 13.6.5 shows that (13.6.10) fails only if $\|c_k\|_2 > \delta_3\Delta_k$. In this case, using $\Delta_k \geq \|d_k\|_2$ too, Lemma 13.6.4 provides the bound

$$\begin{aligned} Pred_k &- \frac{1}{2}\sigma_k(\|c_k\|_2^2 - \|c_k + A_k^T d_k\|_2^2) \\ &\geq \|d_k\|_2\|c_k\|_2\Big[\frac{1}{2}\sigma_k \min(\delta_3, b_2/\delta_5) - \delta_1\Big], \end{aligned} \tag{13.6.44}$$

where δ_5 is an upper bound on $\{\|A_k^+\|_2, k = 1, 2, \cdots\}$. Hence condition (13.6.10) holds if $\sigma_k > 2\delta_1 \max[1/\delta_3, \delta_5/b_2]$. Therefore the number of increase in σ_k is finite. □

We now assume without loss of generality that $\sigma_k \equiv \sigma$ for all k. The next lemma shows that both the trust-region bound and the constraints converges to zero, if the algorithm does not terminate after finitely many iterations.

Lemma 13.6.7 *If the algorithm does not terminate, we have the limits*

$$\lim_{k\to\infty} \Delta_k = 0, \tag{13.6.45}$$

$$\lim_{k\to\infty} \|c_k\|_2 = 0. \tag{13.6.46}$$

Proof. To prove (13.6.45), we assume that the number

$$\eta = \limsup_{k\to\infty} \Delta_k \tag{13.6.47}$$

is positive and deduce a contradiction. Define K to be the set of integers k satisfying

$$r_k \geq 0.1, \quad \Delta_k \geq \eta/8. \tag{13.6.48}$$

The set K contains infinitely many elements because of (13.6.47). Since the monotonically decreasing sequence $\{P(x_k)\}$ is convergent, we have that

$$\lim_{\substack{k\in K\\ k\to\infty}} Pred_k = 0. \tag{13.6.49}$$

Therefore, (13.6.32) does not hold for sufficiently large $k \in K$. It follows from Corollary 13.6.5 and (13.6.48) that

$$\|c_k\|_2 > \delta_3\eta/8 \tag{13.6.50}$$

holds for all sufficiently large $k \in K$. Thus, Lemma 13.6.3 implies that

$$\begin{aligned} Pred_k &\geq \frac{1}{2}\sigma[\|c_k\|_2^2 - \|c_k + A_k^T d_k\|_2^2] \\ &\geq \frac{1}{2}\sigma\|c_k\|_2 \min\Big[\|c_k\|_2, \frac{b_2\Delta_k}{\|A_k^+\|_2}\Big]. \end{aligned} \tag{13.6.51}$$

Using the above inequality, relations (13.6.48) and (13.6.49), we can deduce that

$$\lim_{\substack{k\in K\\ k\to\infty}} \|c_k\|_2 = 0, \tag{13.6.52}$$

which contradicts (13.6.50). Therefore (13.6.45) is true.

As for (13.6.46), we deduce a contradiction from the assumption that

$$\bar{\eta} = \limsup_{k\to\infty} \|c_k\|_2 > 0. \tag{13.6.53}$$

Define $\bar{K} = \{k | \|c_k\|_2 > \bar{\eta}/2\}$. It follows from (13.6.51) and (13.6.45) that there exists a constant $\bar{\delta} > 0$ such that

$$Pred_k \geq \bar{\delta}\Delta_k, \quad \forall k \in \bar{K}. \tag{13.6.54}$$

The above inequality and (13.6.45) imply that

$$\lim_{\substack{k \in K \\ k \to \infty}} r_k = 1, \tag{13.6.55}$$

which, together with (13.6.54), shows that

$$\sum_{k \in \bar{K}} \Delta_k < +\infty. \tag{13.6.56}$$

From the definition of $\bar{K}$, inequality (13.6.56) and the continuity of $c(x)$, we can show that

$$\lim_{k \to \infty} \|c_k\|_2 = \bar{\eta}. \tag{13.6.57}$$

Thus, $k \in \bar{K}$ for all sufficiently large k. This observation and relation (13.6.55) imply that $\Delta_{k+1} \geq \Delta_k$ for all sufficiently large k. This contradicts (13.6.45). Therefore, (13.6.46) is true. □

Having established the above results, we can easily show the global convergence of the algorithm.

Theorem 13.6.8 *Under Assumption 13.6.2, Algorithm 13.6.1 will terminate after finitely many iterations. In other words, if we remove the convergence test from Step 2, then $d_k = 0$ for some k or the limit*

$$\liminf_{k \to \infty} [\|c_k\|_2 + \|\bar{P}_k g_k\|_2] = 0 \tag{13.6.58}$$

is obtained, which ensures that $\{x_k, k = 1, 2, ...\}$ is not bounded away from stationary points of the problem (13.4.1)-(13.4.2).

Proof. First we assume that $\epsilon > 0$. If the algorithm does not terminate, then the inequality

$$\|c_k\|_2 + \|\bar{P}_k g_k\|_2 \geq \epsilon \tag{13.6.59}$$

holds for all k. It follows from (13.6.46) that

$$\|\bar{P}_k g_k\|_2 \geq \epsilon/2 \tag{13.6.60}$$

holds for all sufficiently large k. Using (13.6.60), (13.6.45), (13.6.46) and (13.6.17) we can show that there exists a positive constant δ such that

$$Pred_k \geq \delta \Delta_k \tag{13.6.61}$$

is true for all sufficiently large k. The above inequality implies that

$$\lim_{k \to \infty} r_k = 1, \tag{13.6.62}$$

which leads to the inequality $\Delta_{k+1} \geq \Delta_k$ (for all sufficiently large k). This contradicts (13.6.45). The contradiction indicates that for any positive $\epsilon > 0$ Algorithm 13.6.1 will terminate after finitely many iterations.

If $\epsilon = 0$, then the algorithm terminates if and only if $d_k = 0$. If $d_k = 0$, then x_k is a KKT point of the optimization problem (13.4.1)–(13.4.2). Assume that the algorithm dost not terminate, then $d_k \neq 0$ for all k. Let

$$\eta = \inf_k [\|c_k\|_2 + \|\bar{P}_k g_k\|_2]. \tag{13.6.63}$$

If $\eta > 0$, we see that the algorithm does not terminate for $\epsilon = \eta/2$, which contradicts the proof given above. This shows that we must have $\eta = 0$, which implies (13.6.58). □

Under second-order sufficient conditions and other mild conditions, locally superlinear convergence of the algorithm can be proved (see, Powell and Yuan [278]).

Exercises

1. Prove that the trust-region subproblem

$$\min_{d \in \Re^n} g_k^T d + \frac{1}{2} d^T B_k d + \sigma_k \|(c_k + A_k^T d)^{(-)}\|_\infty$$

subject to

$$\|d\|_\infty \leq \Delta_k$$

can be reformulated as a quadratic programming problem.

2. Extend the null space trust-region method for equality constrained optimization to handle also inequality constraints.

3. Consider the CDT subproblem when

$$g = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \quad B = \begin{pmatrix} -2 & 0 \\ 0 & 2 \end{pmatrix}, \quad A = I, \quad c = \begin{pmatrix} -2 \\ 0 \end{pmatrix},$$

$\Delta = 2$ and $\xi = 1$. Verify that the Hessian of the Lagrange function can have one negative eigenvalue even when only one of the constraints are active at the solution.

4. Construct an example to show that the Hessian of the Lagrange of the CDT subproblem at the solution may have two negative eigenvalues.

5. Let $C, D \in \Re^{n \times n}$ be two symmetric matrices and let A and B be two closed sets in $\Re^n$ such that $A \cup B = \Re^n$. If we have that

$$x^T C x \geq 0, \forall x \in A, \qquad x^T D x \geq 0, \forall x \in B,$$

prove that there exists a $t \in [0, 1]$ such that the matrix $tC + (1 - t)D$ is positive semi-definite.

6. Discuss the local convergence properties of Powell-Yuan's trust-region algorithm.

Chapter 14

Nonsmooth Optimization

14.1 Generalized Gradients

In this book, nonsmooth functions are those functions which need not be differentiable. Therefore they are also called nondifferentiable functions.

The nonlinear programming problem (8.1.1)–(8.1.3) is said to be a nonsmooth optimization problem, provided that either the objective function $f(x)$ or at least one of the constraint functions $c_i(x), (i = 1, \cdots, m)$ is a nonsmooth function.

To conclude the book, we would like to give an initial and readable introduction to nonsmooth optimization. To study the optimality condition of nonsmooth optimization and construct some numerical methods for solving nonsmooth optimization problems, we first introduce the fundamental conceptions and properties of nonsmooth functions.

Let X be a Banach space with a norm $\|\cdot\|$ defined on X. Let Y be a subset of X. A function $f : Y \to R$ is Lipschitz on Y if $f(x)$ satisfies

$$|f(x) - f(y)| \leq K\|x - y\|, \ \forall x, y \in Y \subseteq X, \tag{14.1.1}$$

where K is called the Lipschitz constant. The inequality (14.1.1) is also referred to as a Lipschitz condition.

We define a generalized sphere

$$B(x, \epsilon) = \{y \mid \|x - y\| \leq \epsilon\}. \tag{14.1.2}$$

We say that f is Lipschitz near x if, for some $\epsilon > 0$, f satisfies a Lipschitz condition on $B(x, \epsilon)$.

It is easy to see that a function having a Lipschitz property near a point need not be differentiable there, nor need admit a directional derivative in the classical sense.

The directional derivative of f at x in the direction d is

$$f'(x;d) = \lim_{t \downarrow 0} \frac{f(x+td) - f(x)}{t}. \tag{14.1.3}$$

The upper Dini directional derivative of f at x in the direction d is

$$f^{(D)}(x;d) = \limsup_{t \downarrow 0} \frac{f(x+td) - f(x)}{t}. \tag{14.1.4}$$

Let f be Lipschitz near a given point x, and let d be any other vector in X. The generalized directional derivative of f at x in the direction d is defined as follows:

$$f^o(x;d) = \limsup_{\substack{y \to x \\ t \downarrow 0}} \frac{f(y+td) - f(y)}{t}, \tag{14.1.5}$$

where of course y is a vector in X and t is a positive scalar, and $t \downarrow 0$ denotes that t tends to zero monotonically and downward. Since the generalized directional derivative is due to Clarke [60], it is also referred to as a Clarke directional derivative.

For a locally Lipschitz function, the directional derivative may not exist but the Dini and the Clarke directional derivatives always exist. Obviously, we always have the relation

$$f^{(D)}(x;d) \leq f^o(x;d) \tag{14.1.6}$$

for all x and d. If the directional derivative exists, then it is equal to the upper Dini directional derivative. If $f'(x;d)$ exists at x for all d, then f is said to be directionally differentiable at x. If f is directionally differentiable at x and

$$f'(x;d) = f^o(x;d), \tag{14.1.7}$$

then f is said to be regular at x. The function f is said to be a regular function if it is regular everywhere.

Lemma 14.1.1 *Let $f(x)$ be Lipschitz near x. Then*

1. *The function $d \to f^o(x;d)$ is positive homogeneous and subadditive on X, and satisfies*

$$|f^o(x;d)| \leq K\|d\|. \tag{14.1.8}$$

2. $f^o(x;d)$ is Lipschitz on X as a function of d.

3. $f^o(x;d)$ is upper semicontinuous as a function of $(x;d)$.

4. $f^o(x;-d) = (-f)^o(x;d)$.

Proof. 1) In view of (14.1.1), (14.1.5) and the fact that $f(x)$ is Lipschitz near x, we immediately have (14.1.8). The fact that

$$f^o(x;\lambda d) = \lambda f^o(x;d)$$

for any $\lambda > 0$ is immediate from the definition (14.1.5). Now we turn to the subadditivity.

From (14.1.5), we have

$$\begin{aligned} f^o(x;d_1+d_2) &= \limsup_{\substack{y\to x\\ t\downarrow 0}} \frac{f(y+t(d_1+d_2))-f(y)}{t} \\ &\le \limsup_{\substack{y\to x\\ t\downarrow 0}} \frac{f(y+td_1+td_2)-f(y+td_2)}{t} \\ &\quad + \limsup_{\substack{y\to x\\ t\downarrow 0}} \frac{f(y+td_2)-f(y)}{t} \\ &\le f^o(x;d_1)+f^o(x;d_2). \end{aligned} \tag{14.1.9}$$

2) Let any $d_1, d_2 \in X$ be given. We have from the Lipschitz condition that

$$f(y+td_1)-f(y) \le f(y+td_2)-f(y)+Kt\|d_1-d_2\| \tag{14.1.10}$$

holds for y near x, $t > 0$ sufficiently small. Dividing by t and taking upper limits as $y \to x, t \downarrow 0$, gives

$$f^o(x;d_1) \le f^o(x;d_2) + K\|d_1-d_2\|. \tag{14.1.11}$$

Similarly, we obtain

$$f^o(x;d_2) \le f^o(x;d_1) + K\|d_1-d_2\|. \tag{14.1.12}$$

The above two inequalities give

$$|f^o(x;d_1)-f^o(x;d_2)| \le K\|d_1-d_2\|. \tag{14.1.13}$$

Then we complete 2).

3) Now let $\{x_i\}$ and $\{d_i\}$ be arbitrary sequences with $x_k \to x$ and $d_k \to d$ respectively. For each i, by definition of upper limit, there exist $y_k \in X$ and $t_k > 0$ such that

$$\|y_k - x_k\| + t_k < \frac{1}{k}, \tag{14.1.14}$$

$$\begin{aligned} & f^o(x_k; d_k) - \frac{1}{k} \\ \leq & \frac{f(y_k + td_k) - f(y_k)}{t_k} \\ \leq & \frac{f(y_k + t_k d_k) - f(y_k + t_k d)}{t_k} + \frac{f(y_k + t_k d) - f(y_k)}{t_k}. \end{aligned} \tag{14.1.15}$$

Upon taking upper limits (as $k \to \infty$), we derive

$$\limsup_{k\to\infty} f^o(x_k; d_k) \leq f^o(x; d), \tag{14.1.16}$$

which establishes the upper semicontinuity.

4) Finally, we calculate

$$\begin{aligned} f^o(x; -d) &= \limsup_{\substack{y\to x \\ t\downarrow 0}} \frac{f(y - td) - f(y)}{t} \\ &= \limsup_{\substack{u\to x \\ t\downarrow 0}} \frac{(-f)(u + td) - (-f)(u)}{t} \\ &= (-f)^o(x; d), \end{aligned} \tag{14.1.17}$$

where $u = y - td$. Hence, we complete the proof. □

The Hahn-Banach Theorem (for example, see Cryer [72], Theorem 7.4) asserts that any positive homogeneous and subadditive functional on X majorizes some linear functional on X. Under the condition of Lemma 14.1.1, therefore, there is at least one linear functional $\xi : X \to R$ such that, for all $d \in X$, one has

$$f^o(x; d) \geq \xi(d).$$

It follows also that ξ is bounded, and hence belongs to the dual space X^* of continuous linear functionals on X, for which we adopt the convention of using $\langle \xi, d \rangle$ or $\langle d, \xi \rangle$ for $\xi(d)$.

We then give the following definition:

Definition 14.1.2 *Let $f(x)$ be Lipschitz near x. Then we say that the generalized differential (or Clarke differential) of f at x is the set*

$$\partial f(x) = \{\xi \in X^* \mid f^o(x;d) \geq \langle \xi, d\rangle, \forall d \in X\}. \tag{14.1.18}$$

The ξ is said to be the generalized gradient.

The norm $\|\xi\|_*$ in conjugate space X^* is defined as

$$\|\xi\|_* = \sup\{\langle \xi, d\rangle \ : \ d \in X, \|d\| \leq 1\}. \tag{14.1.19}$$

The following summarizes some basic properties of the generalized gradient.

Lemma 14.1.3 *Let $f(x)$ be Lipschitz near x. Then*

1) $\partial f(x)$ is a nonempty, convex, weak-compact subset of X^* and $\|\xi\|_* \leq K$ for every $\xi \in \partial f(x)$.*

2) For every $d \in X$, one has

$$f^o(x;d) = \max_{\xi \in \partial f(x)} \{\langle \xi, d\rangle\}. \tag{14.1.20}$$

Proof. Assertion 1) is immediate from the preceding remarks and Lemma 14.1.1. (The weak*-compactness follows from Alaoglu's Theorem.)

Assertion 2) is simply a restatement of the fact that $\partial f(x)$ is by definition the weak*-closed convex set whose support function is $f^o(x;\cdot)$. To see this independently, suppose that for some d, $f^o(x;d)$ exceeded the given maximum (it cannot be less, by definition of $\partial f(x)$). According to a common version of the Hahn-Banach Theorem there is a linear functional ξ majorized by $f^o(x,\cdot)$ and agreeing with it at d. It follows that $\xi \in \partial f(x)$, whence $f^o(x;d) > \langle \xi; d\rangle = f^o(x;d)$. This contradiction establishes the assertion 2). □

Note that if $f(x)$ is convex, the conceptions of generalized directional derivative and generalized gradient coincide with that of directional derivative and subgradient defined for convex functions due to Rockafellar [288].

As an example, we calculate the generalized differential of the absolute-value function in the case of $X = R$.

Consider the problem

$$f(x) = |x|.$$

Obviously, f is Lipschitz by the triangle inequality. If $x > 0$, we calculate

$$f^o(x;d) = \limsup_{\substack{y \to x \\ t \downarrow 0}} \frac{y + td - y}{t} = d,$$

so that

$$\partial f(x) = \{\xi \mid d \geq \xi d,\ \forall d \in R\}$$

reduces to the singleton $\{1\}$.

Similarly, we have

$$\partial f(x) = \{-1\} \text{ if } x < 0.$$

The remaining case is $x = 0$. We find

$$f^o(0;d) = \begin{cases} d, & \text{if } d \geq 0, \\ -d, & \text{if } d < 0, \end{cases}$$

that is

$$f^o(0,d) = |d|.$$

Thus $\partial f(0)$ consists of those ξ satisfying $|d| \geq \xi d$ for all d; that is $\partial f(0) = \{-1, 1\}$. Therefore, we conclude

$$\partial f(x) = \begin{cases} \{1\}, & x > 0, \\ \{-1\}, & x < 0, \\ \{-1, 1\}, & x = 0. \end{cases}$$

We introduce an important conception as follows.

The support function of a nonempty subset Ω of X is a function $\sigma_\Omega(\xi) : X^* \to R \cup \{+\infty\}$ defined by

$$\sigma_\Omega(\xi) := \sup_{x \in \Omega} \{\langle \xi, x \rangle\}. \tag{14.1.21}$$

It is easy to see that $f^o(x;\cdot)$ is the support function of $\partial f(x)$.

By (14.1.21) and Definition 14.1.2, the following lemma is obvious.

Lemma 14.1.4 *Let $f(x)$ be Lipschitz near x. Then*

$$\xi \in \partial f(x) \text{ if and only if } f^o(x;d) \geq \langle \xi; d \rangle \ \forall d \in X. \tag{14.1.22}$$

Furthermore, $\partial f(x)$ has the following properties:

a)

$$\partial f(x) = \cap_{\delta>0} \cup_{y\in x+B(0,\delta)} \partial f(y), \tag{14.1.23}$$

where

$$B(0,\delta) = \{x \,|\, \|x\| \leq \delta,\, x \in X\}.$$

If X is finite-dimensional, then the ∂f is upper semi-continuous.

b) If $f_i\,(i = 1, \cdots, m)$ are finitely many Lipschitz functions near x, then $\sum_{i=1}^m f_i$ is also Lipschitz near x and

$$\partial(\sum_{i=1}^m f_i)(x) \subset \sum_{i=1}^m \partial f_i(x). \tag{14.1.24}$$

c) If $f(x) = g(h(x))$, where $h(x) = (h_1(x), \cdots, h_n(x))^T$, each $h_i(x)$ is Lipschitz near x, and $g(x)$ is Lipschitz near $h(x)$, then $f(x)$ is Lipschitz near x and

$$\partial f(x) \subset \overline{co}\left\{\sum_{i=1}^n \alpha_i\xi_i :\ \xi_i \in \partial h_i(x), \alpha \in \partial g(h)|_{h=h(x)}\right\}, \tag{14.1.25}$$

where $\overline{co}$ denotes a weak*-compact convex hull (see Theorem 2.3.9 in Clarke [60]).

Below, we turn to the optimal condition for minimization of a Lipschitz function. By Lemma 14.1.4, we can immediately deduce the first-order necessary condition.

Theorem 14.1.5 *If $f(x)$ attains a local minimum or maximum at x^* and $f(x)$ is Lipschitz near x^*, then*

$$0 \in \partial f(x^*). \tag{14.1.26}$$

Proof. If x^* is a local minimizer of $f(x)$, then it follows from the definition (14.1.5) that for any $d \in X$ we have

$$f^o(x^*; d) \geq 0. \tag{14.1.27}$$

Thus, by Lemma 14.1.4, we have $0 \in \partial f(x^*)$.

If x^* is a local maximizer of $f(x)$, then x^* is a local minimizer of $(-f)(x)$. It suggests that $0 \in \partial(-f)(x^*)$. It is not difficult to show that for any scalar

s, one has $\partial(sf)(x) = s\partial f(x)$. Therefore $0 \in \partial(-f)(x^*) = -\partial f(x^*)$ which means $0 \in \partial f(x^*)$. Hence we complete the proof. $\square$

A point x^* is called a stationary point of f if f is directionally differentiable at x^* and for all d,

$$f'(x^*, d) \geq 0. \tag{14.1.28}$$

A point x^* is called a Dini stationary point of f if for all d,

$$f^{(D)}(x^*; d) \geq 0. \tag{14.1.29}$$

A point x^* is called a Clarke stationary point of f if for all d,

$$f^o(x^*; d) \geq 0, \tag{14.1.30}$$

i.e.,

$$0 \in \partial f(x^*). \tag{14.1.31}$$

A local minimizer x^* of a local Lipschitzian function f is always a Dini stationary point of f. If f is directionally differentiable at x^*, then x^* is also a stationary point. A Dini stationary point is always a Clarke stationary point but not vice versa.

Now we state the sufficient condition which is based on a lemma below.

Lemma 14.1.6 *Let $f(x)$ be convex and Lipschitz near x^*, then the generalized differential $\partial f(x)$ coincides with the subdifferential at x, and the generalized directional derivative $f^o(x; d)$ coincides with the directional derivative $f'(x; d)$ for each d.*

Proof. It is known from convex analysis that $f'(x; d)$ exists for each d and $f'(x; d)$ is the support function of the subdifferential at x. It suffices therefore to prove that for any d, $f^o(x; d) = f'(x; d)$. Note that

$$f^o(x; d) = \lim_{\epsilon \downarrow 0} \sup_{\|x' - x\| < \epsilon\delta} \sup_{0 < t < \epsilon} \frac{f(x' + td) - f(x')}{t}, \tag{14.1.32}$$

where δ is any fixed positive number. It follows from the definition of convex function that the function

$$t \to \frac{f(x' + td) - f(x')}{t}$$

is non-decreasing, whence

$$f^o(x;d) = \lim_{\epsilon \downarrow 0} \sup_{\|x'-x\|<\epsilon\delta} \frac{f(x'+\epsilon d) - f(x')}{\epsilon}.$$

Now by the Lipschitz condition, for any x' in $x + B(0, \epsilon\delta)$, one has

$$\left| \frac{f(x'+\epsilon d) - f(x')}{\epsilon} - \frac{f(x+\epsilon d) - f(x)}{\epsilon} \right| \leq 2\delta K,$$

so that

$$f^o(x;d) \leq \lim_{\epsilon \downarrow 0} \frac{f(x+\epsilon d) - f(x)}{\epsilon} + 2\delta K = f'(x;d) + 2\delta K.$$

Since δ is arbitrary, we deduce $f^o(x;d) \leq f'(x;d)$. Therefore the equality follows. The proof is complete. □

We now can state the sufficient condition.

Theorem 14.1.7 *Let $f(x)$ be convex and Lipschitz near x^*, and*

$$0 \in \partial f(x^*), \tag{14.1.33}$$

then x^* is a local minimizer of $f(x)$.

Proof. For a convex and Lipschitzian function, from Lemma 14.1.6, the generalized differential and the subdifferential

$$\{\xi \in X^* \,|\, f(z) - f(x) \geq \langle \xi, z - x \rangle, \ \forall z \in X\} \tag{14.1.34}$$

are equivalent. Then, by (14.1.33) and (14.1.34), we have that x^* is a local minimizer of $f(x)$. □

Hence, for a convex and Lipschitzian function, (14.1.33) is a sufficient and necessary condition for x^* to be a local minimizer of $f(x)$. This is also equivalent to

$$f^o(x^*;d) \geq 0, \ \forall d \in X. \tag{14.1.35}$$

For a convex and Lipschitzian function, the generalized directional derivative $f^o(x;d)$ coincides with the directional derivative $f'(x;d)$:

$$f'(x;d) = \lim_{t \downarrow 0} \frac{f(x+td) - f(x)}{t}. \tag{14.1.36}$$

(We would like to mention that, from convex analysis, convex functions are Lipschitz except in the pathological case).

Furthermore, we can state a sufficient condition for a strict (strong) minimizer.

Theorem 14.1.8 *Let $f(x)$ be convex and Lipschitz near x^*. If*

$$f'(x^*; d) > 0, \ \forall d \neq 0, d \in X, \tag{14.1.37}$$

then x^* is a strict (strong) minimizer of $f(x)$, i.e., there exists $\delta > 0$ such that

$$f(x) - f(x^*) \geq \delta \|x - x^*\| \tag{14.1.38}$$

holds for all x sufficiently close to x^*.

Proof. Define a set

$$S = \{d \,|\, d \in X, \|d\| = 1\}.$$

Obviously, S is compact and closed. By (14.1.37), it follows that $f'(x^*, d)$ is positive on S. Then, from the continuity of $f'(x^*, d)$ (in fact, $f'(x^*; d)$ is a positive homogeneous and convex function of d), there exists $\delta > 0$ such that

$$f'(x^*; d) \geq 2\delta, \ \forall d \in S. \tag{14.1.39}$$

Then for any $d \in S$, there exists $t(d) > 0$ such that

$$f(x^* + td) - f(x^*) \geq td, \ \forall t \in [0, t(d)]. \tag{14.1.40}$$

By convexity and continuity of $f(x)$, we can show that there is an $\epsilon > 0$ such that

$$t(d) \geq \epsilon, \ \forall d \in S. \tag{14.1.41}$$

Hence, for all x with $\|x - x^*\| \leq \epsilon$, we have

$$f(x) - f(x^*) \geq \delta \|x - x^*\| \tag{14.1.42}$$

which indicates (14.1.38). □

However, for a non-convex function, the above sufficiency result is not true. In fact, let us consider an example below: for $f : R^1 \to R^1$,

$$f(x) = \begin{cases} (-1)^{k+1} \left[\frac{1}{2^{k+1}} - 3x\right], & x \in \left[\frac{1}{2^{k-1}}, \frac{1}{2^k}\right], \\ 0, & x = 0, \end{cases} \tag{14.1.43}$$

$$f(x) = f(-x), \ \forall x \in [-1, 0). \tag{14.1.44}$$

Clearly, $f(x)$ is Lipschitz on $[-1, 1]$, and

$$f^o(x^*; \pm 1) = 3 > 0 \tag{14.1.45}$$

at $x^* = 0$, which means there are two generalized directional derivatives equal to 3. But $x^* = 0$ is not the extreme point.

14.2 Nonsmooth Optimization Problem

Consider unconstrained optimization problem

$$\min_{x\in X} f(x), \tag{14.2.1}$$

where $f(x)$ is a nondifferentiable function defined in Banach space and satisfies a Lipschitz condition. From the discussion in §14.1, if x^* is a solution of (14.2.1), then

$$0 \in \partial f(x^*), \tag{14.2.2}$$

i.e., x^* is a stationary point of (14.2.1).

As to solution for nonsmooth optimization problem (14.2.1), there are two main difficulties if one is using a method suitable for differentiable problems. First, it is not easy to give a termination criteria. It is well-known that when x approaches the minimizer of a continuously differentiable function $f(x)$, the $\|\nabla f(x)\|$ is very small. Hence the common termination criteria

$$\|\nabla f(x)\| \le \epsilon \tag{14.2.3}$$

is used. However, for a nonsmooth function, there are no similar results. For example, consider the simple problem that $f : R^1 \to R^1$ and $f(x) = |x|$. Then, for any x that is not a solution, $f(x)$ is differentiable and

$$|\partial f(x)| = |\nabla f(x)| = 1. \tag{14.2.4}$$

Hence, in this case, we cannot use (14.2.3) as a termination criteria.

Second, as indicated by Wolfe [354], when $f(x)$ is nondifferentiable, if one uses the steepest descent method with line search to solve (14.2.1), it is possible to generate a sequence $\{x_k\}$ converging to a non-stationary point. For example, let $f : R^2 \to R^1, x = (u, v)^T$ and

$$f(x) = \max\left[\frac{1}{2}u^2 + (v-1)^2, \frac{1}{2}u^2 + (v+1)^2\right]. \tag{14.2.5}$$

Suppose that x_k has the form

$$x_k = \begin{pmatrix} 2(1+|\epsilon_k|) \\ \epsilon_k \end{pmatrix}, \tag{14.2.6}$$

where $\epsilon_k \neq 0$. Then we can calculate

$$\nabla f(x_k) = \begin{pmatrix} 2(1+|\epsilon_k|) \\ 2(1+|\epsilon_k|)t_k \end{pmatrix} = 2(1+|\epsilon_k|)\begin{pmatrix} 1 \\ t_k \end{pmatrix}, \tag{14.2.7}$$

where $t_k = \text{sign}(\epsilon_k)$. If we employ the negative gradient direction $-\nabla f(x_k)$, then we have

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k(-\nabla f(x_k)) = \begin{bmatrix} 2(1+|\epsilon_k|/3) \\ -\epsilon_k/3 \end{bmatrix} \\ &= \begin{bmatrix} 2(1+|\epsilon_{k+1}|) \\ \epsilon_{k+1} \end{bmatrix}, \end{aligned} \tag{14.2.8}$$

where $\epsilon_{k+1} = -\epsilon_k/3 \neq 0$. Then we can prove $\epsilon_k \to 0$. So, for a given initial point as $(2+2|\delta|, \delta)^T$, where $\delta \neq 0$, the sequence generated by the steepest descent method with exact line search converges to $(2,0)^T$. It is obvious that $(2,0)^T$ is not the stationary point.

A nonsmooth constrained optimization problem has the form

$$\min_{x \in Y} f(x), \tag{14.2.9}$$

where $Y \subseteq X$ is a set, or a feasible region. Define a distance function

$$\text{dist}(x, Y) = \min_{y \in Y} \|y - x\|. \tag{14.2.10}$$

By the theory of penalty function, under suitable conditions, (14.2.9) is equivalent to

$$\min_{x \in X} f(x) + \sigma \text{dist}(x, Y), \tag{14.2.11}$$

where $f(x) + \sigma \text{dist}(x, Y)$ is a non-differentiable function. Hence, the nonsmooth constrained optimization problem is transformed to an equivalent nonsmooth unconstrained problem. This interprets why one always is interested in studying nonsmooth unconstrained optimization problems.

There are many examples of nonsmooth optimization problems, for example, the minimax problem

$$\min_{x \in X} \max_{1 \leq i \leq m} f_i(x). \tag{14.2.12}$$

In addition, in order to solve nonlinear equations

$$f_i(x) = 0, i = 1, \cdots, m, \tag{14.2.13}$$

we often find the solution of the minimization problem

$$\min_{x \in X} f(x) = \min_{x \in X} \|\bar{f}(x)\| \tag{14.2.14}$$

under some norm $\|\cdot\|$, where $f(x) = \|\bar{f}(x)\|$, $\bar{f}(x) = (f_1(x), \cdots, f_m(x))$ is a vector function from X to R^n. Clearly, the problem (14.2.14) is a nonsmooth optimization problem. In particular, if $\|\cdot\| = \|\cdot\|_1$, it is a L_1 minimization problem; if $\|\cdot\| = \|\cdot\|_\infty$, it is Chebyshev approximation problem.

Note that the exact penalty function (10.6.2) is also a nonsmooth function. Therefore, the minimization to the exact penalty function is also a nonsmooth optimization problem.

14.3 The Subgradient Method

The subgradient method is a direct generalization of the steepest descent method, which generates a sequence $\{x_k\}$ by use of $-g_k$ as a direction, where $g_k \in \partial f(x_k)$.

Let $f(x)$ be a convex function on R^n and the minimization problem be $\min_{x\in R^n} f(x)$. We have seen that the convex function is differentiable almost everywhere, and

$$\partial f(x) = \operatorname{conv} \Omega(x), \tag{14.3.1}$$

where convΩ denotes the convex hull of Ω,

$$\Omega(x) = \{g \,|\, g = \lim \nabla f(x_i), x_i \to x, \nabla f(x_i) \text{ exists}\}. \tag{14.3.2}$$

The subgradient method is described as follows.

Algorithm 14.3.1 *(The subgratient method)*

Step 1. Given an initial point $x_1 \in R^n$, $k := 1$.

Step 2. Compute $f(x_k), g_k \in \partial f(x_k)$.

Step 3. Choose stepsize $\alpha_k > 0$ and set

$$x_{k+1} = x_k - \alpha_k g_k / \|g_k\|_2, \tag{14.3.3}$$

$k := k + 1$, go to Step 2. □

As shown in the above section, in the subgradient method, the exact line search may cause convergence to a non-stationary point.

In smooth optimization, inexact line search is to find the stepsize α_k such that

$$f(x_k + \alpha_k d_k) \le f(x_k) + \alpha_k c_1 d_k^T \nabla f(x_k), \tag{14.3.4}$$

where $c_1 \in (0,1)$ is a constant. For the steepest descent method, the above rule becomes

$$f(x_k - \alpha_k \nabla f(x_k)) \leq f(x_k) - \alpha_k c_1 \|\nabla f(x_k)\|^2. \tag{14.3.5}$$

However, when $f(x)$ is nonsmooth, then for any $c_1 \in (0,1)$ and $g_k \in \partial f(x_k)$, the inequality

$$f(x_k - \alpha g_k) \leq f(x_k) - \alpha c_1 \|g_k\|^2 \tag{14.3.6}$$

may not hold for any $\alpha > 0$. Therefore, inexact line search is also not practicable in nonsmooth optimization.

Note that a constant stepsize is unsuitable because the function may be nondifferentiable at the solution and then $\{g_k\}$ does not necessarily tend to zero, even if $\{x_k\}$ converges to the optimal point.

Therefore, the rules for determining α_k for the subgradient method are entirely different from that for the steepest descent method.

Although the exact and inexact line search for smooth optimization cannot be simply generalized to the nonsmooth case, the negative subgradient direction is a "good" direction such that the new iterate is closer to the solution.

Lemma 14.3.2 *Let $f(x)$ be a convex function and the set*

$$S^* = \{x \,|\, f(x) = f^* = \min_{x \in R^n} f(x)\} \tag{14.3.7}$$

be nonempty. If $x_k \notin S^*$, then for any $x^* \in S^*$ and $g_k \in \partial f(x_k)$, there must exist $T_k > 0$ such that

$$\left\| x_k - \alpha \frac{g_k}{\|g_k\|_2} - x^* \right\|_2 < \|x_k - x^*\|_2 \tag{14.3.8}$$

holds for all $\alpha \in (0, T_k)$.

Proof. For any x_k,

$$\begin{aligned} \left\| x_k - \alpha \frac{g_k}{\|g_k\|_2} - x^* \right\|_2^2 &= \|x_k - x^*\|_2^2 \\ &\quad + 2\alpha \left(\frac{g_k}{\|g_k\|_2} \right)^T (x^* - x_k) + \alpha^2. \end{aligned} \tag{14.3.9}$$

Since $g_k \in \partial f(x_k)$ and $x_k \notin S^*$, then we have

$$g_k^T(x^* - x_k) \leq f(x^*) - f(x_k) < 0. \tag{14.3.10}$$

Define

$$T_k = -2g_k^T(x^* - x_k)/\|g_k\|_2 > 0, \tag{14.3.11}$$

then (14.3.9) becomes

$$\|x_k - \alpha \frac{g_k}{\|g_k\|_2} - x^*\|_2^2 = \|x_k - x^*\|_2^2 + \alpha(\alpha - T_k). \tag{14.3.12}$$

If $0 < \alpha < T_k$, then $\alpha(\alpha - T_k) < 0$ and further (14.3.8) holds. □

By use of the above property of subgradient direction, we can take a sufficiently small step, such that the sequence $\{x_k\}$ is closer and closer to the solution. From the above lemma we can deduce easily the following result due to Shor [309].

Theorem 14.3.3 *Let $f(x)$ be convex and the set S^* be nonempty. For any $\delta > 0$ there exists $r > 0$ such that if the subgradient Algorithm 14.3.1 is used with $\alpha_k \equiv \alpha \in (0, r)$ then we have*

$$\liminf_{k\to\infty} f(x_k) \leq f^* + \delta. \tag{14.3.13}$$

Note that the choice of constant stepsize $\alpha_k \equiv \alpha$ may cause the algorithm not to converge. Ermoliev [118] and Polyak [254] suggest choosing α_k which would satisfy

$$\alpha_k > 0, \ \lim_{k\to\infty} \alpha_k = 0, \tag{14.3.14}$$

$$\sum_{k=1}^{\infty} \alpha_k = \infty, \tag{14.3.15}$$

and establish the following convergence theorem.

Theorem 14.3.4 *Let $f(x)$ be convex, and the set S^* be nonempty and bounded. If α_k satisfies (14.3.14) and (14.3.15), then the sequence $\{x_k\}$ generated by Algorithm 14.3.1 satisfies*

$$\lim_{k\to\infty} dist(x_k, S^*) = 0, \tag{14.3.16}$$

where $dist(x, S)$ is defined by (14.2.10).

Proof. Since $f(x)$ is convex, there exists continuous function $\delta(\epsilon)$ such that

$$f(x) \leq f^* + \epsilon \tag{14.3.17}$$

holds for any

$$\text{dist}(x, S^*) \leq \delta(\epsilon), \tag{14.3.18}$$

where $\delta(\epsilon) > 0\,(\forall \epsilon > 0)$. For each k, we define

$$\epsilon_k = f(x_k) - f^* \geq 0. \tag{14.3.19}$$

If $\epsilon_k > 0$, then

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &= \|x_k - x^*\|^2 + \alpha_k^2 - 2\alpha_k (x_k - x^*)^T g_k / \|g_k\|_2 \\ &= \|x_k - x^*\|^2 + \alpha_k^2 - 2\delta(\epsilon_k)\alpha_k \\ &\quad -2\alpha_k \left[x_k - x^* - \delta(\epsilon_k) \frac{g_k}{\|g_k\|_2} \right]^T g_k / \|g_k\|_2 \\ &\leq \|x_k - x^*\|^2 + \alpha_k^2 - 2\delta(\epsilon_k)\alpha_k. \end{aligned} \tag{14.3.20}$$

Hence

$$[\text{dist}(x_{k+1}, S^*)]^2 - [\text{dist}(x_k, S^*)]^2 \leq -\alpha_k [2\delta(\epsilon_k) - \alpha_k]. \tag{14.3.21}$$

Define $\delta(0) = 0$, then the above expression holds for every k. Summing both sides of (14.3.21) gives

$$\liminf_{k \to \infty} \delta(\epsilon_k) = 0. \tag{14.3.22}$$

Thus,

$$\liminf_{k \to \infty} \text{dist}(x_k, S^*) = 0. \tag{14.3.23}$$

Suppose to the contrary that the theorem is not true. Then there exist a positive constant $\delta' > 0$ and infinitely many k such that

$$\text{dist}(x_{k+1}, S^*) > \text{dist}(x_k, S^*) \tag{14.3.24}$$

and

$$\epsilon_k > \delta' \tag{14.3.25}$$

hold. From (14.3.24) and (14.3.21), we deduce that

$$2\delta(\epsilon_k) < \alpha_k \tag{14.3.26}$$

holds for sufficiently large k. Clearly, (14.3.26) contradicts (14.3.25). This contradiction shows the theorem. □

The above theorem indicates that Algorithm 14.3.1 converges if α_k satisfies (14.3.14)–(14.3.15). However, for such chosen α_k, the algorithm does not converge rapidly. In fact, we have

$$\|x_k - x^*\| + \|x_{k+1} - x^*\| \geq \|x_k - x_{k+1}\| = \alpha_k. \tag{14.3.27}$$

Then, by (14.3.27) and (14.3.15), we have immediately that

$$\sum_{k=1}^{\infty} \|x_k - x^*\| = +\infty, \tag{14.3.28}$$

which shows that the sequence cannot converge R-linearly.

To make the algorithm converge R-linearly, Shor [310] takes

$$\alpha_k = \alpha_0 q^k, \ 0 < q < 1. \tag{14.3.29}$$

But, such an α_k does not satisfy (14.3.15). For any given α_0 and q, as long as

$$\text{dist}(x_1, S^*) > \frac{\alpha_0}{1-q}, \tag{14.3.30}$$

the sequence generated from the algorithm is not possible to close S^*.

The convergence result of the algorithm with step rule (14.3.29) is stated as follows.

Theorem 14.3.5 *Let $f(x)$ be convex and let there exist positive constant $\delta_1 > 0$ such that for all x,*

$$(x - x^*)^T g \geq \delta_1 \|g\| \|x - x^*\|, \ \forall g \in \partial f(x), \tag{14.3.31}$$

then there must exist constants $\bar{q} \in (0, 1)$ and $\bar{\alpha} > 0$ such that, provided that

$$q \in (\bar{q}, 1), \ \alpha_0 > \bar{\alpha}, \tag{14.3.32}$$

then the sequence $\{x_k\}$ generated by Algorithm 14.3.1 satisfies

$$\|x_k - x^*\| \leq M(\delta, \alpha_0) q^k, \tag{14.3.33}$$

where $x^* \in S^*$, $\bar{q}$ and $\bar{\alpha}$ are constants related to $\|x_1 - x^*\|$ and δ_1, $M(\delta_1, \alpha_0) > 0$ is a constant irrelative to k and related to δ_1 and α_0.

However, the rule (14.3.29) to determine stepsize is almost infeasible in practice, because, in general, it is impossible to know the values of $\bar{\alpha}$ and $\bar{q}$. If the given α_0 is too small, then (14.3.32) is not satisfied; if α_0 is too big, then the algorithm converges very slowly.

When f^* is known in advance, let us set

$$\alpha_k = \lambda \frac{f(x_k) - f^*}{\|g_k\|}, \ 0 < \lambda < 2. \tag{14.3.34}$$

The convergence theorem of Algorithm 14.3.1 with stepsize rule (14.3.34) is due to Polyak [255] and stated as follows.

Theorem 14.3.6 *Let $f(x)$ be convex and the set S^* be nonempty. If there exist positive numbers $\bar{c}$ and $\hat{c}$ such that*

$$\|g\| \leq \bar{c}, \ \forall g \in \partial f(x), \tag{14.3.35}$$

$$f(x) - f^* \geq \hat{c}\, dist(x, S^*) \tag{14.3.36}$$

hold for all x satisfying $dist(x, S^*) \leq dist(x_1, S^*)$, then the sequence generated by Algorithm 14.3.1 with stepsize (14.3.34) converges to some $x^* \in S^*$, and there exists a positive constant M such that

$$\|x_k - x^*\| \leq Mq^k, \tag{14.3.37}$$

where $q = (1 - \lambda(2 - \lambda)\hat{c}^2/\bar{c}^2)^{1/2} < 1$.

The above discussion has shown that the improvements only in the stepsize rule cannot, in general, significantly accelerate convergence. Indeed, slow convergence is due to the fact that the gradient is almost perpendicular to the direction towards the minimum. There is a simple way of changing the angles between the gradient and the direction towards the minimum. This can be done by performing a space dilation technique, which, in fact, is a generalization of the variable metric method.

Now we describe the space dilation method as follows:

Algorithm 14.3.7 *(The space dilation method)*

Step 1. Given initial point $x_1, \alpha > 0, H_1 = \alpha I; k := 1$.

Step 2. Evaluate $g_k \in \partial f(x_k)$; find the stepsize $\alpha_k > 0$; set

$$x_{k+1} = x_k - \alpha_k H_k g_k / (g_k^T H_k g_k)^{1/2}. \tag{14.3.38}$$

Step 3. Choose $r_k > 0$ *and* $\beta_k < 1$. *Set*

$$H_{k+1} = r_k \left(H_k - \beta_k \frac{H_k g_k g_k^T H_k}{g_k^T H_k g_k} \right). \tag{14.3.39}$$

$k := k + 1$, *go to Step 2.* □

It is not difficult to see that the matrix sequence $\{H_k\}$ generated by (14.3.39) are positive definite. There are various ways to choose α_k, β_k and r_k, for example,

$$\alpha_k = \frac{1}{n+1},\ \beta_k = \frac{2}{n+2},\ r_k = \frac{n^2}{n^2-1}. \tag{14.3.40}$$

Below, we state the convergence of the space dilation method without proof. The interested reader can consult Shor [311].

Theorem 14.3.8 *Let* $f(x)$ *be convex and the set* S^* *be nonempty. If*

$$dist(x_1, S^*) \ \leq \alpha,$$

then the sequence $\{x_k\}$ *generated by Algorithm 14.3.7 with (14.3.40) satisfies*

$$\liminf_{k\to\infty} \frac{f(x_k) - f^*}{q^k} < +\infty, \tag{14.3.41}$$

where

$$q = \left(1 - \frac{2}{n+1}\right)^{\frac{1}{2n}} \frac{n}{\sqrt{n^2-1}}. \tag{14.3.42}$$

There are other generalizations to the subgradient method, for example, ellipsoid algorithm, finite difference approximation etc. We refer the readers to Zowe [386] and Shor [313] for details.

14.4 Cutting Plane Method

The cutting plane method for convex programming was presented independently by Kelley [186] and Cheney and Goldstein [58] respectively. The underlying idea of the cutting plane method is to find the minimum of a function on a convex polyhedral set in each iteration. After each iteration, a cutting plane is introduced, and a point, which does not satisfy the new

hyperplane, is cut off from the feasible region, and hence the polyhedral set is reduced. At last, the iterates converge to a solution. The procedure is performed by solving a sequence of approximating linear programming.

For convex function $f(x)$, obviously, we have

$$f(x) = \sup_y \sup_{g \in \partial(y)} [f(y) + g^T(x - y)]. \tag{14.4.1}$$

Therefore, the minimization of $f(x)$ is equivalent to the following problem

$$\min \quad v \tag{14.4.2}$$

$$\text{s.t.} \quad v \geq f(y) + g^T(x - y), \ \forall y \in R^n, g \in \partial f(y). \tag{14.4.3}$$

The cutting plane method is just, at each iteration, to solve an approximation problem to (14.4.2)–(14.4.3). Let $x_i \, (i = 1, \cdots, k)$ be existing iterates. At each iteration, we would like to solve the subproblem

$$\min \quad v \tag{14.4.4}$$

$$\text{s.t.} \quad v \geq f(x_i) + g_i^T(x - x_i), \ i = 1, \cdots, k. \tag{14.4.5}$$

Obviously, the linear programming problem (14.4.4)–(14.4.5) is an approximation to problem (14.4.2)–(14.4.3).

We can state the cutting plane method as follows.

Algorithm 14.4.1 *(Cutting plane method)*

Step 1. Given an initial point $x_1 \in S$, where S is a given polyhedral set. Set $k := 1$.

Step 2. Compute $g_k \in \partial f(x_k)$.

Step 3. Solve the linear program (14.4.4)–(14.4.5) for v_{k+1} and x_{k+1}. Set $k := k + 1$, go to Step 2. □

As indicated above, at each iteration, the algorithm adds a new constraint, which means, in geometry, that a part in S which does not contain the solution, will be cut off by a hyperplane.

The convergence of the cutting plane method can be stated below.

Theorem 14.4.2 *Let $f(x)$ be convex and bounded below. Then the sequences $\{x_k\}$ and $\{v_k\}$ generated by Algorithm 14.4.1 satisfy*

1) $v_2 \leq v_3 \leq \cdots \leq v_k \to f^*$.

2) Any accumulation point of $\{x_k\}$ is a minimizer of $f(x)$ in S.

Suppose that $f(x)$ is differentiable and the algorithm converges to a solution, then for k sufficiently large, $g_k = \nabla f(x_k)$ is very small, and hence the constraint condition (14.4.5) will be ill-conditioned. The other disadvantage of the cutting plane method is that when k is sufficiently large, there are too many constraints in problem (14.4.4)–(14.4.5) such that the cost is prohibitively expensive, since cutting plane constraints are always added to the existing set of constraints but are never deleted. Because of these disadvantages, the cutting plane methods have never been attractive, although it is one of the earliest methods for general convex programming. Therefore, some modified versions of the cutting plane methods are needed.

14.5 The Bundle Methods

The bundle method is a class of methods extended from the conjugate subgradient method. This is a descent method with $f(x_{k+1}) \leq f(x_k)$ for each k.

The conjugate subgradient method was presented by Wolfe [354]. At the k-th iteration, there is an index set $I_k \subset \{1, \cdots, k\}$. The search direction is determined by

$$d_k = -\sum_{i \in I_k} \lambda_i^{(k)} g_i,\ g_i \in \partial f(x_k), \tag{14.5.1}$$

where $\lambda_i^{(k)}$ $(i \in I_k)$ are obtained by solving the subproblem

$$\min \quad \left\| \sum_{i \in I_k} \lambda_i g_i \right\|_2^2 \tag{14.5.2}$$

$$\text{s.t.} \quad \sum_{i \in I_k} \lambda_i = 1,\ \lambda_i \geq 0. \tag{14.5.3}$$

When $f(x)$ is a convex quadratic function and $I_k = \{1, 2, \cdots, k\}$, under exact line search, the direction generated from (14.5.1)–(14.5.3) is the same as that of the conjugate gradient method. So, this method is said to be a conjugate subgradient method. We now state the algorithm as follows.

Algorithm 14.5.1 *(Conjugate Subgradient Method)*

Step 1. Given initial point $x_1 \in R^n$, compute $g_1 \in \partial f(x_1)$. Choose $0 < m_2 < m_1 < \frac{1}{2}$, $0 < m_3 < 1$; $\epsilon > 0$, $\eta > 0$, $k := 1$; $I_1 = \{1\}$.

Step 2. *Compute the direction d_k by (14.5.1)–(14.5.3). If $\|d_k\| \le \eta$ stop.*

Step 3. *Compute $y_k = x_k + \alpha_k d_k$ such that*

$$f(y_k) \le f(x_k) - m_2\alpha_k\|d_k\|_2^2, \tag{14.5.4}$$

or

$$\|y_k - x_k\| \le m_3\epsilon. \tag{14.5.5}$$

Step 4. *If there is $g_{k+1} \in \partial f(y_k)$ such that*

$$g_{k+1}^T d_k \ge -m_1\|d_k\|_2^2, \tag{14.5.6}$$

then set $x_{k+1} := y_k$, otherwise set $x_{k+1} := x_k$.

Step 5. *Set $I_{k+1} := I_k \cup \{k+1\} \setminus T_k$, where T_k is an index set*

$$T_k = \{i \,|\, \|x_i - x_{k+1}\| > \epsilon\}.$$

Step 6. *$k := k+1$, go to Step 2.* □

The following convergence theorem was given by Wolfe [354].

Theorem 14.5.2 *Let $f(x)$ be convex and $\|\partial f(x)\|$ be bounded on some open set containing the set $\{x \,|\, f(x) \le f(x_1)\}$. Let the sequence $\{x_k\}$ generated by Algorithm 14.5.1 make $f(x_k)$ bounded below. Then the algorithm must terminate in finitely many iterations.*

Now we consider an extension of the conjugate subgradient method. Suppose that we have performed several steps of the conjugate subgradient method. A certain number of points have been generated, at which the value of f has been computed together with some subgradient. We symbolize this information by the bundle $x_1, \cdots, x_k; f_1, \cdots, f_k; g_1, \cdots, g_k$; where $f_i = f(x_i)$ and $g_i \in \partial f(x_i)$.

Suppose that at k-th iteration we have weighted factors $t_i^{(k)} \ge 0$ $(i = 1, \cdots, k)$. Consider the following subproblem

$$\min \quad \left\| \sum_{i=1}^{k} \lambda_i g_i \right\| \tag{14.5.7}$$

$$\text{s.t.} \quad \sum_{i=1}^{k} \lambda_i = 1,\ \lambda_i \geq 0, \tag{14.5.8}$$

$$\sum_{i+1}^{k} \lambda_i t_i^{(k)} \leq \bar{\epsilon}, \tag{14.5.9}$$

where $\bar{\epsilon} > 0$ is a given constant. Write $\lambda_i^{(k)}$ as a solution of (14.5.7)–(14.5.9). Then the search direction of the bundle method is

$$d_k = -\sum_{i=1}^{k} \lambda_i^{(k)} g_i. \tag{14.5.10}$$

It is not difficult to see that if $t_i^{(k)} = 0\,(i \in I_k)$ and $t_i^{(k)} = +\infty\,(i \notin I_k)$, then (14.5.7)–(14.5.9) is equivalent completely to (14.5.2)–(14.5.3).

Algorithm 14.5.3 *(Bundle Method)*

Step 1. *Given initial point $x_1 \in R^n$, compute $g_1 \in \partial f(x_1)$. Choose $0 < m_2 < m_1 < \frac{1}{2}$, $0 < m_3 < 1$, $\epsilon > 0, \eta > 0, k := 1$ and $t_1^{(1)} = 1$.*

Step 2. *Solve (14.5.7)–(14.5.9) for $\lambda_i^{(k)}$.*
Compute d_k by (14.5.10).
If $\|d_k\| \leq \eta$ stop.

Step 3. *Compute $y_k = x_k + \alpha_k d_k$ such that (14.5.4) holds or*

$$f(y_k) - \alpha_k g_{k+1}^T d_k \geq f(x_k) - \epsilon, \tag{14.5.11}$$

where $g_{k+1} \in \partial f(y_k)$.
If (14.5.4) does not hold, then go to Step 5.

Step 4. $x_{k+1} := y_k, t_{k+1}^{(k+1)} = 1,$
$t_j^{(k+1)} = t_j^{(k)} + f(x_{k+1}) - f(x_k) - \alpha_k g_j^T d_k,\ j = 1, \cdots, k.$
Set $k := k + 1$, go to Step 2.

Step 5. $x_{k+1} := x_k, t_j^{(k+1)} = t_j^{(k)}\ (j = 1, \cdots, k)$
$t_{k+1}^{(k+1)} = f(x_k) - f(y_k) + \alpha_k g_{k+1}^T d_k.$
Set $k := k + 1$, go to Step 2. □

The convergence of the bundle method was established by Lemarechal [196] and stated below.

Theorem 14.5.4 *Under the assumptions of Theorem 14.5.2, Algorithm 14.5.3 will terminate in finitely many iterations, i.e., there exists $k \in IN$ such that $f(x_k) \leq f^* + \epsilon$, where IN is an index set of positive integers.*

14.6 Basic Property of a Composite Nonsmooth Function

In the following two sections of the chapter, we will discuss a problem with the special form

$$\min_{x \in R^n} h(f(x)), \tag{14.6.1}$$

and develop the trust-region method for solving this class of problems. In (14.6.1), $f(x) = (f_1(x), \cdots, f_m(x))^T$ is a continuously differentiable function, and $h(f) : R^m \to R^1$ is convex but nonsmooth. The objective function in (14.6.1) is a composite function, and the problem (14.6.1) is referred to as composite nonsmooth optimization (for brief, composite NSO) or composite nondifferentiable optimization (for brief, composite NDO).

There are many examples of composite NSO in discrete approximation and data fitting. The following is a simple example.

Consider linear equations

$$Ax = b, \tag{14.6.2}$$

where $A \in R^{m \times n}$ and $b \in R^m$. If $m > n$, the equations (14.6.2) in general have no solution. However, we can take x such that the error between Ax and b is as small as possible. This means that we need to solve the minimization problem

$$\min_{x \in R^n} \|Ax - b\|, \tag{14.6.3}$$

where $\|\cdot\|$ is a norm on R^m. Obviously, (14.6.3) is a form of (14.6.1). If we take $\|\cdot\|_2$ in (14.6.3), the problem is just the classical least-squares problem.

In addition, note that a general smooth constrained optimization problem can be transformed to a composite NSO problem via an L_1 exact penalty function. This is the other reason that the composite NSO attracts us.

A prerequisite for describing algorithms for composite NSO is a study of optimality conditions for composite NSO, which is a direct use of the result

in §14.1. For simplicity, we introduce the following conception:

$$\chi(x,d) = h(f(x)) - h(f(x) + A(x)^T d), \tag{14.6.4}$$

$$\psi_t(x) = \max_{\|d\| \le t} \chi(x,d), \tag{14.6.5}$$

$$DF(x,d) = \sup_{\lambda \in \partial h(f(x))} d^T A(x)\lambda, \tag{14.6.6}$$

where $\partial h(f(x))$ denotes the subgradient of $h(\cdot)$ at $f(x)$, $A(x) = \nabla f(x)^T$ is an $n \times m$ matrix.

Since $h(\cdot)$ is a convex function, by use of the chain rule of the subgradient of a composite function, it is not difficult to get the following lemma.

Lemma 14.6.1 *For composite function $\tilde{f}(x) = h(f(x))$, the fact*

$$0 \in \partial \tilde{f}(x) \tag{14.6.7}$$

is equivalent to

$$DF(x,d) \ge 0, \ \forall d \in R^n. \tag{14.6.8}$$

Then the stationary point of nonsmooth optimization satisfies (14.6.8). From the convexity of $h(f)$, we can also obtain the following results:

Lemma 14.6.2 *Let $\chi(x,d)$, $\psi_t(x)$, $DF(x,d)$ be defined in (14.6.4)–(14.6.6). Then*

1) $DF(x,d)$ exists for all x and d ;

2) $\chi(x,d)$ is a concave function with respect to d, its directional derivative at $d^* = 0$ in the direction d is $-DF(x,d)$.

3) $\psi_t(x) \ge 0$, $\forall t \ge 0$; $\psi_1(x) = 0$ if and only if x is a stationary point;

4) $\psi_t(x)$ is a concave function of t;

5) $\psi_t(x)$ is a continuous function of x for any given $t \ge 0$.

By the above results, we can show that the following statements are equivalent:

1) The sequence $\{x_k\}$ has an accumulation point x^* which is a stationary point.

2)

$$\liminf_{k \to \infty} \psi_1(x_k) = 0. \tag{14.6.9}$$

From the necessity theorem in §14.1, it follows that if x^* is a minimizer of $h(f(x))$, then it is a stationary point. For a special form of composite nonsmooth function, it can be written in the following equivalent form.

Theorem 14.6.3 *If x^* is a local minimizer of composite NSO problem (14.6.1), then there exists $\lambda^* \in \partial h(f(x^*))$ such that*

$$A(x^*)\lambda^* = 0, \tag{14.6.10}$$

where $A(x) = \nabla f(x)^T$.

Proof. It is enough to prove that (14.6.10) and

$$DF(x^*, d) \geq 0, \ \forall d \in R^n \tag{14.6.11}$$

are equivalent.

If (14.6.10) holds, then it follows from the definition (14.6.6) that (14.6.11) holds.

Now let us assume that (14.6.11) holds. Suppose to the contrary that (14.6.10) does not hold. Then the set

$$\bar{S} = \{A(x^*)\lambda \mid \lambda \in \partial h(f(x^*))\} \tag{14.6.12}$$

does not contain the origin. Since $\partial h(f(x^*))$ is a closed convex set, then $\bar{S}$ is too. Hence by applying the separation theorem of convex sets, we know there must exist $\bar{d} \in R^n$ such that

$$\bar{d}^T A(x^*)\lambda < 0, \ \forall \lambda \in \partial h(f(x^*)). \tag{14.6.13}$$

Since $\partial h(f(x^*))$ is closed, the above expression (14.6.13) contradicts the fact that $DF(x^*, \bar{d}) \geq 0$. This contradiction shows the equivalence between (14.6.11) and (14.6.10). □

Although the function $\tilde{f}(x) = h(f(x))$ may not be convex, we can obtain the following first-order sufficient conditions.

Theorem 14.6.4 *(First order sufficient conditions) If*

$$DF(x^*, d) > 0 \tag{14.6.14}$$

holds for all nonzero vectors d, then x^* is a strictly local minimizer of $h(f(x))$.

Proof. By (14.6.14), there exists $\delta > 0$ such that

$$DF(x^*, d) \geq \delta, \ \forall \|d\|_2 = 1. \tag{14.6.15}$$

Suppose that the theorem is not true, then there exists $x_k \to x^*$ with $h(f(x_k)) \le h(f(x^*))$. Let us suppose that

$$x_k = x^* + \alpha_k d_k,\ \|d_k\|_2 = 1,\ \alpha_k > 0,\ \alpha_k \to 0_+.$$

Then

$$\begin{aligned} & h(f(x_k)) - h(f(x^*)) \\ = \ & h(f(x^*) + A(x^*)^T(x_k - x^*)) - h(f(x^*)) + o(\alpha_k) \\ \ge \ & \alpha_k DF(x^*, d_k) + o(\alpha_k) \\ \ge \ & \alpha_k \delta + o(\alpha_k), \end{aligned} \tag{14.6.16}$$

which contradicts the fact that $h(f(x_k)) \le h(f(x^*))$. The contradiction proves the theorem. □

In fact, it also follows from (14.6.16) that, under assumption (14.6.14), there exist $\bar{\delta}$ and $\bar{\epsilon}$ such that

$$h(f(x)) - h(f(x^*)) \ge \bar{\delta}\|x - x^*\| \tag{14.6.17}$$

holds for all x with $\|x - x^*\| \le \bar{\epsilon}$.

14.7 Trust Region Method for Composite Nonsmooth Optimization

For composite nonsmooth optimization (14.6.1), the subproblem of the trust-region method has the form

$$\min_{d \in R^n} \quad h(f(x_k) + A(x_k)^T d) + \frac{1}{2} d^T B_k d \triangleq \phi_k(d) \tag{14.7.1}$$

$$\text{s.t.} \quad \|d\| \le \Delta_k, \tag{14.7.2}$$

where $A(x) = \nabla f(x)^T \in R^{n \times m}$, $B_k \in R^{n \times n}$ is a symmetric matrix, and $\Delta_k > 0$ is a radius of the trust-region which is adjusted adaptively to be as large as possible subject to adequate agreement between $\phi_k(d)$ and $h(f(x_k + d))$ being maintained. The norm $\|\cdot\|$ in (14.7.2) is arbitrary but $\|\cdot\|_2$ is used in this section without special specification.

Let d_k be a solution of subproblem (14.7.1)–(14.7.2). Similar to Theorem 14.6.3, we can prove that there must exist

$$\lambda_k \in \partial h(f(x_k) + A(x_k)^T d_k), \tag{14.7.3}$$

$$\mu_k \in \partial \|d_k\|, \tag{14.7.4}$$

and $\bar{\mu}_k \geq 0$ such that

$$A(x_k)\lambda_k + B_k d_k + \bar{\mu}_k \mu_k = 0, \tag{14.7.5}$$

$$\bar{\mu}_k[\Delta_k - \|d_k\|] = 0. \tag{14.7.6}$$

The trust-region algorithm for composite nonsmooth optimization due to Fletcher [129] is as follows.

Algorithm 14.7.1 *(Trust-region algorithm for composite NSO)*

Step 1. *Given* $x_1 \in R^n, \lambda_0 \in R^m$, $\Delta_1 > 0$, $\epsilon \geq 0$, $k := 1$.

Step 2. *Compute*

$$B_k = \sum_{i=1}^{m} (\lambda_{k-1})_i \nabla^2 f_i(x_k); \tag{14.7.7}$$

Solve the subproblem (14.7.1)–(14.7.2) for d_k*;*
If $\|d_k\| \leq \epsilon$*, stop.*

Step 3. *Calculate*

$$r_k = \frac{h(f(x_k)) - h(f(x_k + d_k))}{\phi_k(0) - \phi_k(d_k)}. \tag{14.7.8}$$

If $r_k < 0.25$ *set* $\Delta_{k+1} := \|d_k\|/4$*;*
if $r_k > 0.75$ *and* $\|d_k\| = \Delta_k$*, set* $\Delta_{k+1} = 2\Delta_k$*;*
otherwise, set $\Delta_{k+1} = \Delta_k$.

Step 4. *If* $r_k > 0$ *go to Step 5;*
else $x_{k+1} := x_k$, $\lambda_k := \lambda_{k-1}$, *go to Step 6.*

Step 5. *Set* $x_{k+1} := x_k + d_k$, λ_k *is defined by (14.7.5).*

Step 6. $k := k + 1$, *go to Step 2.* □

To analyze the convergence of Algorithm 14.7.1, we assume that the sequence $\{x_k\}$ from the algorithm is bounded, which is implied if any level set $\{x \mid h(f(x)) \leq h(f(x_1))\}$ is bounded. The boundedness of $\{x_k\}$ suggests that there exists a bounded, closed convex set Ω such that

$$x_k \in \Omega,\ x_k + d_k \in \Omega,\ \forall k = 1, 2, \cdots. \tag{14.7.9}$$

Since $h(\cdot)$ is convex and well-defined on all of R^m, then there exists constant $L > 0$ such that

$$|h(f_1) - h(f_2)| \leq L\|f_1 - f_2\| \tag{14.7.10}$$

holds for all $f_1, f_2 \in f(\Omega) = \{v = f(x), x \in \Omega\}$. From the continuous differentiability of f and the boundedness of Ω, it follows that there is a constant $M > 0$ such that

$$\|A(x)\| \leq M \tag{14.7.11}$$

holds for all $x \in \Omega$.

Theorem 14.7.2 *Let $f_i(x)$ $(i = 1, \cdots, m)$ be twice continuously differentiable, if the sequence $\{x_k\}$ generated by Algorithm 14.7.1 is bounded, then there exists an accumulation point x^* of Algorithm 14.7.1, which is a stationary point of optimization problem (14.6.1).*

As to the proof of the theorem, please consult Fletcher (1981). Further, we have the following corollary.

Corollary 14.7.3 *Under the assumption of Theorem 14.7.2, if, instead of (14.7.7), $\|B_k\|$ is uniformly bounded, then $\{x_k\}$ has an accumulation point x^*, which is a stationary point.*

Now, the uniform boundedness of $\|B_k\|$ is relaxed to

$$\|B_k\| \leq c_5 + c_6 \sum_{i=1}^{k} \Delta_i. \tag{14.7.12}$$

Also, the adjustment of trust-region radius can be extended to the general case:

$$\|d_k\| \leq \Delta_{k+1} \leq \min[c_1\Delta_k, \bar{\Delta}], \quad \text{if } r_k \geq c_2, \tag{14.7.13}$$

$$c_3\|d_k\| \leq \Delta_{k+1} \leq c_4\Delta_k, \qquad \text{if } r_k < c_2, \tag{14.7.14}$$

where $c_i (i = 1, \cdots, 6)$ are positive constants and satisfy $c_1 > 1 > c_4 > c_3$, $c_2 < 1$; $\bar{\Delta}$ is a constant given in advance, an upper bound of the trust-region radius.

Under the extended conditions, we also can establish the convergence. We first give a lemma.

Lemma 14.7.4 *Let d_k be a solution of (14.7.1)–(14.7.2), then*

$$h(f(x_k)) - \phi_k(d_k) \geq \frac{1}{2}\psi_{\Delta_k}(x_k) \min\left[1, \frac{\psi_{\Delta_k}(x_k)}{\|B_k\|\Delta_k^2}\right], \tag{14.7.15}$$

where $\psi_t(x)$ is defined by (14.6.4)–(14.6.5).

Proof. It follows from the definition of d_k that

$$h(f(x_k)) - \phi_k(d_k) \geq h(f(x_k)) - \phi_k(d) \tag{14.7.16}$$

holds for any d with $\|d\| \leq \Delta_k$. By the definition (14.6.5) of $\psi_t(x)$, there exists $\|\bar{d}_k\| \leq \Delta_k$ such that

$$\psi_{\Delta_k}(x_k) = h(f(x_k)) - h(f(x_k) + A(x_k)^T \bar{d}_k). \tag{14.7.17}$$

Then, by using the convexity of $h(\cdot)$, we obtain that

$$\begin{aligned} h(f(x_k)) - \phi_k(d_k) &\geq h(f(x_k)) - \phi_k(\alpha\bar{d}_k) \\ &= \chi(x_k, \alpha\bar{d}_k) - \frac{1}{2}\alpha^2 \bar{d}_k^T B_k \bar{d}_k \\ &\geq \alpha\chi(x_k, \bar{d}_k) - \frac{1}{2}\alpha^2 \|B_k\| \|\bar{d}_k\|^2 \\ &\geq \alpha\psi_{\Delta_k}(x_k) - \frac{1}{2}\alpha^2 \|B_k\| \Delta_k^2 \end{aligned} \tag{14.7.18}$$

holds for all $\alpha \in [0, 1]$. Therefore

$$\begin{aligned} h(f(x_k)) - \phi_k(d_k) &\geq \max_{0\leq\alpha\leq 1}\left[\alpha\psi_{\Delta_k}(x_k) - \frac{1}{2}\alpha^2\|B_k\|\Delta_k^2\right] \\ &\geq \frac{1}{2}\min\left[\psi_{\Delta_k}(x_k), \frac{[\psi_{\Delta_k}(x_k)]^2}{\|B_k\|\Delta_k^2}\right]. \end{aligned} \tag{14.7.19}$$

We complete the proof. □

It is now possible to establish an extended conclusion of Theorem 14.7.2.

Theorem 14.7.5 *Let $f_i(x)\,(i = 1, \cdots, m)$ be twice continuously differentiable. Suppose that B_k in Algorithm 14.7.1 is not given by (14.7.7) but instead by (14.7.12) and that the sequence $\{x_k\}$ of the algorithm is bounded, then there must exist an accumulation point x^* of $\{x_k\}$ which is a stationary point of the problem (14.6.1).*

Proof. Suppose that the theorem does not hold, so there exists a positive constant $\delta > 0$ such that

$$\psi_1(x_k) \geq \delta, \ \forall k. \tag{14.7.20}$$

By use of 5) of Lemma 14.6.2, Lemma 14.7.4, inequality (14.7.20) and boundedness of Δ_k, we deduce that

$$\begin{aligned} h(f(x_k)) - \phi_k(d_k) &\geq c_7 \min\left[\Delta_k, \frac{1}{\|B_k\|}\right] \\ &\geq c_7 \min\left[\Delta_k, \frac{1}{c_5 + c_6 \sum_{i=1}^{k} \Delta_i}\right], \end{aligned} \tag{14.7.21}$$

where c_7 is a positive constant. Define a set

$$S = \{k \,|\, r_k \geq c_2\}, \tag{14.7.22}$$

then we have

$$\begin{aligned} h(f(x_1)) - \min_{x\in\Omega} h(f(x)) &\geq \sum_{k=1}^{\infty} [h(f(x_k)) - h(f(x_{k+1}))] \\ &\geq \sum_{k\in S} [h(f(x_k)) - h(f(x_{k+1}))] \\ &\geq c_2 \sum_{k\in S} [h(f(x_k)) - \phi_k(d_k)]. \end{aligned} \tag{14.7.23}$$

By (14.7.23), (14.7.21), (14.7.12) and $\Delta_k \leq \bar{\Delta}$, it follows that

$$\sum_{k\in S} \Delta_k / \left(c_5 + c_6 \sum_{i=1}^{k} \Delta_i\right) < +\infty. \tag{14.7.24}$$

In view of definition of Δ_{k+1}, we have

$$\Delta_{k+1} \leq c_4 \Delta_k, \ \forall k \notin S, \tag{14.7.25}$$

which gives

$$\sum_{i=1}^{k} \Delta_i \leq \left(1 + \frac{c_1}{1 - c_4}\right) \left[\sum_{\substack{i=1 \\ i\in S}}^{k} \Delta_i + \Delta_1\right]. \tag{14.7.26}$$

Combining (14.7.24) and (14.7.26) yields that $\sum_{i\in S} \Delta_i$ converges, and further that $\sum_{k=1}^{\infty} \Delta_k$ converges by (14.7.26) again. Hence $\|B_k\|$ is uniformly

bounded. So, by Corollary 14.7.3, we know that (14.7.20) cannot hold for all k. The contradiction proves the theorem. □

Similar to the analysis of the trust-region method for unconstrained optimization, the condition (14.7.12) can further be weakened to

$$\|B_k\| \leq c_8 + c_9 k. \tag{14.7.27}$$

However, for the nonsmooth trust-region method, no matter what choices of B_k, there is only linear convergence. Several modifications are available to avoid the Maratos effect and enable the second-order rate to be established. The interested reader can consult Fletcher [131] and Yuan [369] for details.

14.8 Nonsmooth Newton's Method

Qi and Sun [283] extended the classical Newton's method to a non-smooth case by using the generalized Jacobian instead of the classical Jacobian. In this section, following Qi and Sun [283], we discuss the non-smooth Newton's method.

First, we introduce the generalized Jacobian and semismooth function. Suppose that $F : R^n \to R^m$ is a locally Lipschitzian function. Rademacher's theorem says that F is differentiable almost everywhere. Denote the set of points at which F is differentiable by D_F. We write $JF(x)$ for the usual $m \times n$ Jacobian matrix of partial derivatives whenever x is a point at which the necessary partial derivatives exist.

The generalized Jacobian of F at x, denoted by $\partial F(x)$, is a convex hull of all $m \times n$ matrices V obtained as the limit of a sequence of the form $JF(x_i)$, where $x_i \to x$ and $x_i \in D_F$. Then, we have

$$\partial F(x) = \text{co}\,\{\lim JF(x_i) \mid x_i \to x,\, x_i \in D_F\}. \tag{14.8.1}$$

Let F be Lipschitz on an open convex set U in R^n, and let x and y be points in U. Then, by Proposition 2.6.5 of Clarke [60], one has

$$F(y) - F(x) \in \partial F([x, y])(y - x). \tag{14.8.2}$$

Assume that for any $h \in R^n$,

$$\lim_{\substack{V \in \partial F(x+th) \\ t \downarrow 0}} \{Vh\} \tag{14.8.3}$$

exists. Then the classical directional derivative

$$F'(x;h) = \lim_{t\downarrow 0} \frac{F(x+th) - F(x)}{t} \tag{14.8.4}$$

exists, and

$$F'(x;h) = \lim_{\substack{V\in\partial F(x+th)\\ t\downarrow 0}} \{Vh\}. \tag{14.8.5}$$

In fact, by (14.8.2), we have

$$\frac{F(x+t_jh) - F(x)}{t_j} \in \operatorname{co}\partial F([x, x+t_jh])h.$$

By the Carathèodory theorem, there exist $t_j^{(k)} \in [0, t_j]$, $\lambda_j^{(k)} \in [0,1]$, $V_j^{(k)} \in \partial F([x, x+t_j^{(k)}h])$, for $k = 0, 1, \cdots, m$, $\sum_{k=0}^m \lambda_j^{(k)} = 1$, such that

$$\frac{F(x+t_jh) - F(x)}{t_j} = \sum_{k=0}^m \lambda_j^{(k)} V_j^{(k)} h.$$

By passing to a subsequence, we can assume that $\lambda_j^{(k)} \to \lambda_j$ as $j \to \infty$. We have $\lambda_j \in [0,1]$ for $k = 0, \cdots, m$ and $\sum_{k=0}^m \lambda_j = 1$. Then there are $t_j \downarrow 0$ such that

$$\begin{aligned} F'(x;h) &= \lim_{j\to\infty} \frac{F(x+t_jh) - F(x)}{t_j} = \lim_{j\to\infty} \{\sum_{k=0}^m \lambda_j^{(k)} V_j^{(k)} h\} \\ &= \sum_{k=0}^m \lim_{j\to\infty} \lambda_j^{(k)} \lim_{j\to\infty} \{V_j^{(k)} h\} = \sum_{k=0}^m \lambda_j \lim_{\substack{V\in\partial F(x+th)\\ t\downarrow 0}} \{Vh\} \\ &= \lim_{\substack{V\in\partial F(x+th)\\ t\downarrow 0}} \{Vh\}. \end{aligned}$$

F is called semismooth at x if F is locally Lipschitzian at x and

$$\lim_{\substack{V\in\partial F(x+th')\\ h'\to h, t\downarrow 0}} \{Vh'\} \tag{14.8.6}$$

exists for any $h \in R^n$. It implies that

$$\lim_{\substack{h'\to h\\ t\downarrow 0}} \frac{F(x+th') - F(x)}{t} = \lim_{\substack{V\in\partial F(x+th')\\ h'\to h, t\downarrow 0}} \{Vh'\}. \tag{14.8.7}$$

Lemma 14.8.1 *suppose that $F : R^n \to R^m$ is locally Lipschitzian and $F'(x;h)$ exists for any h at x. Then*
(1) $F'(x;h)$ is Lipschitzian;
(2) for any h, there exists a $V \in \partial F(x)$ such that

$$F'(x;h) = Vh. \tag{14.8.8}$$

Proof. For any $h, h' \in R^n$,

$$\begin{aligned} \|F'(x;h) - F'(x;h')\| &= \left\| \lim_{t\downarrow 0} \frac{F(x+th) - F(x+th')}{t} \right\| \\ \leq \lim_{t\downarrow 0} \frac{\|F(x+th) - F(x+th')\|}{t} &\leq L\|h - h'\|, \end{aligned}$$

where L is the Lipschitzian constant near x. This proves (1).

By (14.8.2) and (14.8.4), there are a sequence $\{t_k\}$ and a sequence $\{V_k\}$ such that $t_k \downarrow 0, V_k \in \operatorname{co} \partial F([x, x + t_k h])$,

$$F'(x;h) = \lim_{k\to\infty} \{V_k h\}.$$

Because of the local Lipschitzian property of F, $\{V_k\}$ is bounded. By passing to a subsequence, we may assume that $V_k \to V$. Also since ∂F is closed, $V \in \partial F(x)$. So, (2) is proved. □

If F is semismooth, then for any $V \in \partial F(x + h)$ and $h \to 0$,

$$Vh - F'(x;h) = o(\|h\|) \tag{14.8.9}$$

and

$$\lim_{\substack{x+h\in D_f \\ h\to 0}} \frac{F'(x+h;h) - F'(x;h)}{\|h\|} = 0. \tag{14.8.10}$$

In fact, if F is semismooth, we have a conclusion that the right-hand side of (14.8.7) is uniformly convergent for all h. Suppose that this conclusion does not hold. Then there exist $\epsilon > 0$, $\{h_k \in R^n \mid \|h_k\| = 1, k = 1, 2, \cdots\}$, $\|\bar{h}_k - h_k\| \to 0$, $t_k \downarrow 0$, $V_k \in \partial F(x + t_k \bar{h}_k)$ such that

$$\|V_k \bar{h}_k - F'(x;h_k)\| \geq 2\epsilon, \tag{14.8.11}$$

for $k = 1, 2, \cdots$. By passing to a subsequence, we may assume that $h_k \to h$. Thus, $\bar{h}_k \to h$ too. By Lemma 14.8.1 (1) and (14.8.11), we can get

$$\|V_k \bar{h}_k - F'(x;h)\| \geq \epsilon \tag{14.8.12}$$

for all sufficiently large k. This contradicts the semismoothness assumption.

The uniform convergence of the right-hand side of (14.8.7) implies the uniform convergence of the right-hand side of (14.8.5), which further implies (14.8.9).

Also, it immediately follows from (14.8.9) and (14.8.8) that (14.8.10) holds.

The Fréchet derivative $F'(x)$ is said to be strong if

$$\lim_{\substack{y\to x\\ z\to x}} \frac{F(z)-F(y)-F'(x)(z-y)}{\|z-y\|}=0. \tag{14.8.13}$$

Clearly, if F has strong Fréchet derivative at x, then F is semismooth at x.

If for any $V\in\partial F(x+h)$ and $h\to 0$,

$$Vh-F'(x;h)=O(\|h\|^{1+p}),$$

where $0<p\le 1$, then we call F p-order semismooth at x. Obviously, p-order semismoothness $(0<p\le 1)$ implies semismoothness.

Note that, if F is semismooth at x, then for any $h\to 0$,

$$F(x+h)-F(x)-F'(x;h)=o(\|h\|). \tag{14.8.14}$$

If F is p-order semismooth at x, then for any $h\to 0$,

$$F(x+h)-F(x)-F'(x;h)=O(\|h\|^{1+p}). \tag{14.8.15}$$

Now, we are in a position to give the nonsmooth Newton's method.

It is well-known that for smooth function $F: R^n\to R^n$, the Newton's method for solving the nonlinear equation

$$F(x)=0 \tag{14.8.16}$$

is

$$x_{k+1}=x_k-[F'(x_k)]^{-1}F(x_k). \tag{14.8.17}$$

Now, suppose that F is not a smooth function, but a locally Lipschitzian function. Then the formula (14.8.17) cannot be used. Let $\partial F(x_k)$ be the generalized Jacobian of F at x_k. Instead of (14.8.17), we may use

$$x_{k+1}=x_k-V_k^{-1}F(x_k), \tag{14.8.18}$$

where $V_k\in\partial F(x_k)$, to solve the nonsmooth equation

$$F(x)=0. \tag{14.8.19}$$

Lemma 14.8.2 *If all $V \in \partial F(x)$ are nonsingular, then there is a neighborhood $N(x)$ of x and a constant C such that for any $y \in N(x)$ and any $V \in \partial F(y)$, V is nonsingular and*

$$\|V^{-1}\| \leq C. \tag{14.8.20}$$

Proof. By contradiction. If the lemma is not true, there is a sequence $y_k \to x$, $V_k \in \partial F(y_k)$ such that either all V_k are singular or $\|V_k^{-1}\| \to \infty$. Since F is locally Lipschitzian, ∂F is bounded in a neighborhood of x. By passing to a subsequence, we may assume that $V_k \to V$. Then V must be singular, a contradiction to the assumption for this proposition. This completes the proof. □

Theorem 14.8.3 *(Local Convergence) Suppose that x^* is a solution of nonsmooth equation (14.8.19), F is locally Lipschitzian and semismooth at x^*, and all $V \in \partial F(x^*)$ are nonsingular. Then the iterative method (14.8.18) is well-defined and convergent to x^* in a neighborhood of x^*. If in addition F is p-order semismooth at x^*, then the convergence of (14.8.18) is of order $1 + p$.*

Proof. By Lemma 14.8.2, the iteration (14.8.18) is well-defined in the neighborhood of x^*. By (14.8.18), (14.8.9) and (14.8.14), we have

$$\begin{aligned} \|x_{k+1} - x^*\| &= \|x_k - x^* - V_k^{-1}F(x_k)\| \\ &\leq \|V_k^{-1}[F(x_k) - F(x^*) - F'(x^*, x_k - x^*)]\| \\ &\quad + \|V_k^{-1}[V_k(x_k - x^*) - F'(x^*; x_k - x^*)]\| \\ &= o(\|x_k - x^*\|). \end{aligned} \tag{14.8.21}$$

The case that F is p-order semismooth at x is similar. □

Finally we give the global convergence of nonsmooth Newton's method.

Theorem 14.8.4 *(Global Convergence) Suppose that F is locally Lipschitzian and semismooth on $S = \{x \in R^n : \|x - x_0\| \leq r\}$. Also suppose that for any $V \in \partial F(x)$ and $x, y \in S$, V is nonsingular,*

$$\|V^{-1}\| \leq \beta, \ \|V(y - x) - F'(x; y - x)\| \leq \gamma\|y - x\|,$$

and

$$\|F(y) - F(x) - F'(x; y - x)\| \leq \delta\|y - x\|,$$

where $\alpha = \beta(\gamma+\delta) < 1$ and $\beta\|F(x_0)\| \le r(1-\alpha)$. Then the iterates (14.8.18) remain in S and converge to the unique solution x^* of (14.8.19). Moreover, the error estimate

$$\|x_k - x^*\| \le [\alpha/(1-\alpha)]\|x_k - x_{k-1}\| \tag{14.8.22}$$

holds for $k = 1, 2, \cdots$

Proof. Obviously,

$$\|x_1 - x_0\| = \|V_0^{-1}F(x_0)\| \le \beta\|F(x_0)\| \le r(1-\alpha).$$

So $x_1 \in S$. Suppose now that $x_1, x_2, \cdots, x_k \in S$. Then

$$\begin{aligned}
\|x_{k+1} - x_k\| &= \|V_k^{-1}F(x_k)\| \le \beta\|F(x_k)\| \\
&\le \beta\|F(x_k) - F(x_{k-1}) - F'(x_{k-1}; x_k - x_{k-1})\| \\
&\quad +\beta\|V_{k-1}(x_k - x_{k-1}) - F'(x_{k-1}, x_k - x_{k-1})\| \\
&\le \beta(\delta+\gamma)\|x_k - x_{k-1}\| = \alpha\|x_k - x_{k-1}\| \le \alpha^k\|x_1 - x_0\| \\
&\le r\alpha^k(1-\alpha).
\end{aligned} \tag{14.8.23}$$

Hence

$$\|x_{k+1} - x_0\| \le \sum_{j=0}^{k}\|x_{j+1} - x_j\| \le \sum_{j=0}^{k} r\alpha^j(1-\alpha) \le r. \tag{14.8.24}$$

So $x_{k+1} \in S$, i.e., all the iterates (14.8.18) remain in S.

For any k and n,

$$\|x_{k+n+1} - x_k\| \le \sum_{j=k}^{k+n}\|x_{j+1} - x_j\| \le \sum_{j=k}^{k+n} r\alpha^j(1-\alpha) \le r\alpha^k. \tag{14.8.25}$$

So the iterates (14.8.18) converge to a point x^* in S. Since F is Lipschitzian in S, $\|V_k\|$ is uniformly bounded. Thus

$$\|F(x^*)\| = \lim_{k\to\infty}\|F(x_k)\| \le \lim_{k\to\infty}\|V_k\|\|x_{k+1} - x_k\| = 0,$$

i.e., $F(x^*) = 0$.

Suppose that there are $x^*, y^* \in S$ with $F(x^*) = 0$ and $F(y^*) = 0$. Let $V^* \in \partial F(x^*)$. Then

$$\begin{aligned} \|y^* - x^*\| &\leq \beta\|V^*(y^* - x^*)\| \\ &\leq \beta\|V^*(y^* - x^*) - F'(x^*; y^* - x^*)\| \\ &\quad +\beta\|F(y^*) - F(x^*) - F'(x^*; y^* - x^*)\| \\ &\leq \beta(\delta + \gamma)\|y^* - x^*\| = \alpha\|y^* - x^*\|. \end{aligned} \tag{14.8.26}$$

This implies

$$\|y^* - x^*\| \leq 0,$$

i.e., $x^* = y^*$. This shows that x^* is the unique solution of (14.8.19).

Finally,

$$\begin{aligned} \|x_{k+n+1} - x_k\| &\leq \sum_{j=k}^{k+n} \|x_{j+1} - x_j\| \leq \sum_{j=0}^{n} \alpha^{j+1}\|x_k - x_{k-1}\| \\ &\leq \frac{\alpha}{1-\alpha}\|x_k - x_{k-1}\|. \end{aligned}$$

Setting $n \to \infty$, we obtain the result (14.8.22). □

Exercises

1. Describe directional derivative, Dini directional derivative, Clarke directional derivative of f at x in the direction d respectively, and their properties and relations.

2. Describe the definition and properties of semi-smoothness.

3. Assume that $f(x)$ is continuously differentiable. Prove that

$$\partial f(x) = \nabla f(x).$$

4. Assume that $c_i(x)(i = 1, ..., m)$ are continuously differentiable. Let $f(x) = \max_{1 \leq i \leq m} c_i(x)$ and $\bar{f}(x) = \sum_{i=1}^{m} |c_i(x)|$. Compute $\partial f(x)$ and $\partial \bar{f}(x)$.

5. Prove Theorem 14.3.3.

6. Assume that $f(x)$ is a convex function. Prove

$$f(x) = \sup_y \sup_{g \in \partial f(y)} [f(y) + g^T(x - y)].$$

7. Prove Theorem 14.4.2.

8. Prove the global convergence of the bundle method for uniformly convex functions.

9. Prove Lemma 14.6.2.

10. Apply the trust-region Algorithm 14.7.1 to problem

$$\min f(x) = \max\{1 + x_1 - x_2^2, 1 - x_1 + (1 + \epsilon)x_2^2\}$$

where $\epsilon > 0$ is a small positive number with the starting point (δ, δ^2) and initial trust-region radius $\Delta_1 = 0.5\delta$, $\delta > 0$ being a small positive number. You should observe that the iterates converge only linearly if the trust-region is chosen $\{d| \quad \|d\|_\infty \leq \Delta_k\}$.

11. Prove Theorem 14.7.2.

12. Modify Algorithm 14.7.1 to derive a nonmonotone algorithm.

13. Give a generalized Newton's method for nonsmooth optimization and establish its global and local convergence.

Appendix: Test Functions

§1.Test Functions for Unconstrained Optimization Problems

Problem 1.1 *Rosenbrock function*:

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2, \tag{1.1}$$

$$x_0 = [-1.2, 1]^T,\ x^* = [1, 1]^T,\ f(x^*) = 0.$$

Problem 1.2 *Extended Rosenbrock function*:

$$f(x) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2], \tag{1.2}$$

$$x_0 = [-1.2, 1, \cdots, -1.2, 1]^T,\ x^* = [1, 1, \cdots, 1, 1]^T,\ f(x^*) = 0.$$

Problem 1.3 *Wood function*:

$$\begin{aligned} f(x) \quad = \quad & 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 \\ & +10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1), \end{aligned} \tag{1.3}$$

$$x_0 = [-3, -1, -3, -1]^T,\ x^* = [1, 1, 1, 1]^T, f(x^*) = 0.$$

Problem 1.4 *Powell singular function*:

$$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4, \tag{1.4}$$

$$x_0 = [3, -1, 0, 1]^T,\ x^* = [0, 0, 0, 0]^T,\ f(x^*) = 0.$$

Problem 1.5 *Cube function*

$$f(x) = 100(x_2 - x_1^3)^2 + (1 - x_1)^2, \tag{1.5}$$

$$x_0 = [-1.2, -1]^T,\ x^* = [1, 1]^T,\ f(x^*) = 0.$$

Problem 1.6 *Trigonometric function*

$$f(x) = \sum_{i=1}^{n} \left[n + i(1 - \cos x_i) - \sin x_i - \sum_{j=1}^{n} \cos x_j \right]^2, \tag{1.6}$$

$$x_0 = \left[\frac{1}{5n}, \cdots, \frac{1}{5n} \right]^T, \ x^* = [0, \cdots, 0]^T, \ f(x^*) = 0.$$

Problem 1.7 *Helical valley function*

$$f(x) = 100[(x_3 - 10\theta)^2 + (\sqrt{x_1^2 + x_2^2} - 1)^2] + x_3^2, \tag{1.7}$$

where

$$2\pi\theta = \begin{cases} \arctan(x_1/x_2) & \text{if } x_1 > 0, \\ \pi + \arctan(x_2/x_1) & \text{if } x_1 < 0, \end{cases}$$

$$x_0 = [-1, 0, 0]^T, \ x^* = [1, 0, 0]^T, \ f(x^*) = 0.$$

§2. Test Functions for Constrained Optimization Problems

The test functions for constrained optimization are selected from Hock and Schittkowski [176].

Problem 2.1 (No. 14 in [176])
Number of Variables: $n = 2$
Objective Function:

$$f(x) = (x_1 - 2)^2 + (x_2 - 1)^2$$

Constraints:

$$-0.25x_1^2 - x_2^2 + 1 \geq 0,$$
$$x_1 - 2x_2 + 1 = 0.$$

Start: $x_0 = (2, 2), \ f(x_0) = 1.$
Solution: $x^* = (0.5(\sqrt{7} - 1), \ 0.25(\sqrt{7} + 1)),$
$f(x^*) = 9 - 2.875\sqrt{7}.$

Problem 2.2 (No. 22 in [176])

Number of Variables: $n = 2$
Objective Function:

$$f(x) = (x_1 - 2)^2 + (x_2 - 1)^2$$

Constraints:

$$\begin{aligned} -x_1 - x_2 + 2 &\geq 0 \\ -x_1^2 + x_2 &\geq 0 \end{aligned}$$

Start: $x_0 = (2, 2),\ f(x_0) = 1.$
Solution: $x^* = (1, 1),\ f(x^*) = 1.$

Problem 2.3 (No. 59 in [176])
Number of Variables: $n = 2$
Objective Functions:

$$\begin{aligned} f(x) \quad &= -75.196 + 3.8112x_1 + 0.0020567x_1^3 - 1.0345\text{E-}5x_1^4 \\ &+6.8306x_2 - 0.030234x_1x_2 + 1.28134\text{E-}3x_2x_1^2 \\ &+2.266\text{E-}7x_1^4x_2 - 0.25645x_2^2 + 0.0034604x_2^3 - 1.3514\text{E-}5x_2^4 \\ &+28.106/(x_2 + 1) + 5.2375\text{E-}6x_1^2x_2^2 + 6.3\text{E-}8x_1^3x_2^2 \\ &-7\text{E-}10x_1^3x_2^3 - 3.405\text{E-}4x_1x_2^2 + 1.6638\text{E-}6x_1x_2^3 \\ &+2.8673\exp(0.0005x_1x_2) - 3.5256\text{E-}5x_1^3x_2 \end{aligned}$$

Constraints:

$$\begin{aligned} &x_1x_2 - 700 \geq 0, \\ &x_2 - x_1^2/125 \geq 0, \\ &(x_2 - 50)^2 - 5(x_1 - 55) \geq 0, \\ &0 \leq x_1 \leq 75, \\ &0 \leq x_2 \leq 65. \end{aligned}$$

Start: $x_0 = (90, 10),\ f(x_0) = 86.878639$
Solution: $x^* = (13.55010424,\ 51.66018129),\ f(x^*) = -7.804226324.$

Problem 2.4 (No. 63 in [176])
Number of Variables: $n = 3$
Objective Function:

$$f(x) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3$$

Constraints:

$$\begin{aligned} &8x_1 + 14x_2 + 7x_3 - 56 = 0, \\ &x_1^2 + x_2^2 + x_3^2 - 25 = 0, \\ &0 \le x_i,\ i = 1, 2, 3. \end{aligned}$$

Start: $x_0 = (2, 2, 2),\ f(x_0) = 976$
Solution: $x^* = (3.512118414,\ 0.2169881741,\ 3.552174034),\ f(x^*) = 961.7151721$

Problem 2.5 (No. 25 in [176])
Number of Variables: $n = 3$
Objective Function:

$$f(x) = \sum_{i=1}^{99} (f_i(x))^2$$

where

$$\begin{aligned} f_i(x) &= -0.01i + \exp(-\frac{1}{x_1}(u_i - x_2)^{x_3}) \\ u_i &= 25 + (-50\ln(0.01i))^{2/3},\ i = 1, \cdots, 99. \end{aligned}$$

Constraints:

$$\begin{aligned} 0.1 \le\ & x_1 \ \le 100 \\ 0 \le\ & x_2 \ \le 25.6 \\ 0 \le\ & x_3 \ \le 5 \end{aligned}$$

Start: $x_0 = (100, 12.5, 3),\ f(x_0) = 32.835$
Solution: $x^* = (50, 25, 1.5),\ f(x^*) = 0$

Problem 2.6 (No. 35 in [176])
Number of Variables: $n = 3$
Objective Function:

$$\begin{aligned} f(x) &= 9 - 8x_1 - 6x_2 - 4x_3 + 2x_1^2 + 2x_2^2 + x_3^2 \\ &\quad +2x_1x_2 + 2x_1x_3 \end{aligned}$$

Constraints:

$$\begin{aligned} &3 - x_1 - x_2 - 2x_3 \ge 0 \\ &0 \le x_i,\ i = 1, 2, 3. \end{aligned}$$

Start: $x_0 = (0.5, 0.5, 0.5),\ f(x_0) = 2.25$
Solution: $x^* = (4/3, 7/9, 4/9),\ f(x^*) = 1/9.$

Problem 2.7 (No. 38 in [176])
Number of Variables: $n = 4$
Objective Function:

$$\begin{aligned} f(x) &= 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 \\ &\quad +10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1) \end{aligned}$$

Constraints:

$$-10 \le x_i \le 10,\ i = 1, \cdots, 4$$

Start: $x_0 = (-3, -1, -3, -1),\ f(x_0) = 19192$
Solution: $x^* = (1, 1, 1, 1),\ f(x^*) = 0.$

Problem 2.8 (No. 43 in [176])
Number of Variables: $n = 4$
Objective Function:

$$f(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$$

Constraints:

$$\begin{aligned} &8 - x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 \ge 0 \\ &10 - x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 \ge 0 \\ &5 - 2x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 \ge 0 \end{aligned}$$

Start: $x_0 = (0, 0, 0, 0),\ f(x_0) = 0.$
Solution: $x^* = (0, 1, 2, -1),\ f(x^*) = -44$

Problem 2.9 (No. 73 in [176])
Number of Variables: $n = 4$
Objective Function:

$$f(x) = 24.55x_1 + 26.75x_2 + 39x_3 + 40.50x_4$$

Constraints:

$$\begin{aligned} &2.3x_1 + 5.6x_2 + 11.1x_3 + 1.3x_4 - 5 \ge 0 \\ &12x_1 + 11.9x_2 + 41.8x_3 + 52.1x_4 - 21 \\ &-1.645(0.28x_1^2 + 0.19x_2^2 + 20.5x_3^2 + 0.62x_4^2)^{\frac{1}{2}} \ge 0 \\ &x_1 + x_2 + x_3 + x_4 - 1 = 0 \\ &0 \le x_i,\ i = 1, \cdots, 4. \end{aligned}$$

Start: $x_0 = (1, 1, 1, 1),\ f(x_0) = 130.8$
Solution:

$$x^* = (0.6355216, -0.12\text{E-}11, 0.3127019, 0.05177655),$$
$$f(x^*) = 29.894378$$

Problem 2.10 (No. 83 in [176])
Number of Variables: $n = 5$
Objective Function:

$$f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$$

Constraints:

$$92 \geq a_1 + a_2x_2x_5 + a_3x_1x_4 - a_4x_3x_5 \geq 0$$
$$20 \geq a_5 + a_6x_2x_5 + a_7x_1x_2 + a_8x_3^2 - 90 \geq 0$$
$$5 \geq a_9 + a_{10}x_3x_5 + a_{11}x_1x_3 + a_{12}x_3x_4 - 20 \geq 0$$
$$78 \leq x_1 \leq 102$$
$$33 \leq x_2 \leq 45$$
$$27 \leq x_i \leq 45,\ i = 3, 4, 5,$$

where

$$a_1 = 85.334407,\ a_2 = 0.0056858,\ a_3 = 0.0006262,$$
$$a_4 = 0.0022053,\ a_5 = 80.51249,\ a_6 = 0.0071317,$$
$$a_7 = 0.0029955,\ a_8 = 0.0021813,\ a_9 = 9.300961,$$
$$a_{10} = 0.0047026,\ a_{11} = 0.0012547,\ a_{12} = 0.0019085$$

Start: $x_0 = (78, 33, 27, 27, 27),\ f(x_0) = -32217$
Solution: $x^* = (78, 33, 29.99526, 45, 36.77581),\ f(x^*) = -30665.53867$

Problem 2.11 (No. 86 in [176])
Number of Variables: $n = 5$
Objective Function:

$$f(x) = \sum_{j=1}^{5} e_jx_j + \sum_{i=1}^{5}\sum_{j=1}^{5} c_{ij}x_ix_j + \sum_{j=1}^{5} d_jx_j^3$$

Constraints:

$$\sum_{j=1}^{5} a_{ij}x_j - b_i \geq 0,\ i = 1, \cdots, 10,$$
$$0 \leq x_i,\ i = 1, \cdots, 5,$$

where

j	1	2	3	4	5
e_j	-15	-27	-36	-18	-12
c_{1j}	30	-20	-10	32	-10
c_{2j}	-20	39	-6	-31	32
c_{3j}	-10	-6	10	-6	-10
c_{4j}	32	-31	-6	39	-20
c_{5j}	-10	32	-10	-20	30
d_j	4	8	10	6	2
a_{1j}	-16	2	0	1	0
a_{2j}	0	-2	0	4	2
a_{3j}	-3.5	0	2	0	0
a_{4j}	0	-2	0	-4	-1
a_{5j}	0	-9	-2	1	-2.8
b_j	-40	-2	-0.25	-4	-4

Start: $x_0 = (0, 0, 0, 0, 1),\ f(x_0) = 20$
Solution: $x^* = (0.3, 0.33346761, 0.4, 0.42831010, 0.22396487),\ f(x^*) = -32.34867897$

Problem 2.12 (No. 93 in [176])
Number of Variables: $n = 6$
Objective Function:

$$\begin{aligned} f(x) \quad = \quad & 0.0204x_1x_4(x_1 + x_2 + x_3) + 0.0187x_2x_3(x_1 + 1.57x_2 + x_4) \\ & +0.0607x_1x_4x_5^2(x_1 + x_2 + x_3) \\ & +0.0437x_2x_3x_6^2(x_1 + 1.57x_2 + x_4) \end{aligned}$$

Constraints:

$$\begin{aligned} & 0.001x_1x_2x_3x_4x_5x_6 - 2.07 \geq 0, \\ & 1 - 0.00062x_1x_4x_5^2(x_1 + x_2 + x_3), \\ & -0.00058x_2x_3x_6^2(x_1 + 1.57x_2 + x_4) \geq 0, \\ & 0 \leq x_i,\ i = 1, \cdots 6. \end{aligned}$$

Start: $x_0 = (5.54, 4.4, 12.02, 11.82, 0.702, 0.852),\ f(x_0) = 137.066$

Solution:

$$x^* = (5.332666, 4.656744, 10.43299,$$
$$12.08230, 0.7526074, 0.87865084),$$
$$f(x^*) = 135.075961$$

Problem 2.13 (No. 108 in [176])
Number of Variables: $n = 9$
Objective Function:

$$f(x) = -0.5(x_1x_4 - x_2x_3 + x_3x_9 - x_5x_9 + x_5x_8 - x_6x_7)$$

Constraints:

$$1 - x_3^2 - x_4^2 \geq 0,$$
$$1 - x_5^2 - x_6^2 \geq 0,$$
$$1 - x_9^2 \geq 0,$$
$$1 - x_1^2 - (x_2 - x_9)^2 \geq 0,$$
$$1 - (x_1 - x_5)^2 - (x_2 - x_6)^2 \geq 0,$$
$$1 - (x_1 - x_7)^2 - (x_2 - x_8)^2 \geq 0,$$
$$1 - (x_3 - x_5)^2 - (x_4 - x_6)^2 \geq 0,$$
$$1 - (x_3 - x_7)^2 - (x_4 - x_8)^2 \geq 0,$$
$$1 - x_7^2 - (x_8 - x_9)^2 \geq 0,$$
$$x_1x_4 - x_2x_3 \geq 0,$$
$$x_3x_9 \geq 0,$$
$$-x_5x_9 \geq 0,$$
$$x_5x_8 - x_6x_7 \geq 0,$$
$$0 \leq x_9.$$

Start:

$$x_0 = (1, 1, 1, 1, 1, 1, 1, 1, 1),$$
$$f(x_0) = 0$$

Solution:

$$x^* = (0.8841292, 0.4672425, 0.03742076, 0.9992996,$$
$$0.8841292, 0.4672424, 0.03742076, 0.9992996,$$
$$0.26\text{E-}19),$$
$$f(x^*) = -0.8660254038$$

Problem 2.14 (No. 110 in [176])
Number of Variables: $n = 10$
Objective Function:

$$f(x) = \sum_{i=1}^{10}[(\ln(x_i - 2))^2 + (\ln(10 - x_i))^2 - (\prod_{i=1}^{10} x_i)^2]$$

Constraints:

$$2.001 \le x_i \le 9.999,\ i = 1, \cdots, 10.$$

Start: $x_0 = (9, \cdots, 9),\ f(x_0) = -43.134337$
Solution: $x^* = (9.35025655, \cdots, 9.35025655),\ f(x^*) = -45.77846971$

Problem 2.15 (No. 111 in [176])
Number of Variables: $n = 10$
Objective Function:

$$f(x) = \sum_{j=1}^{10} \exp(x_j)(c_j + x_j - \ln(\sum_{k=1}^{10} \exp(x_k)))$$

where

$$c_1 = -6.089,\ c_2 = -17.164,\ c_3 = -34.054,$$
$$c_4 = -5.914,\ c_5 = -24.721,\ c_6 = -14.986,$$
$$c_7 = -24.100,\ c_8 = -10.708,\ c_9 = -26.662,\ c_{10} = -22.179$$

Constraints:

$$\exp(x_1) + 2\exp(x_2) + 2\exp(x_3) + \exp(x_6) + \exp(x_{10}) - 2 = 0,$$
$$\exp(x_4) + 2\exp(x_5) + \exp(x_6) + \exp(x_7) - 1 = 0,$$
$$\exp(x_3) + \exp(x_7) + \exp(x_8) + 2\exp(x_9) + \exp(x_{10}) - 1 = 0,$$
$$-100 \le x_i \le 100,\ i = 1, \cdots, 10.$$

Start: $x_0 = (-2.3, \cdots, -2.3),\ f(x_0) = -21.015$
Solution:

$$\begin{aligned} x^* &= (-3.201212, -1.912060, -0.2444413, -6.537489, \\ &\quad -0.7231524, -7.267738, -3.596711, -4.017769, \\ &\quad -3.287462, -2.335582), \\ f(x^*) &= -47.76109026 \end{aligned}$$

Problem 2.16 (No. 112 in [176])
Number of Variables: $n = 10$
Objective Function:

$$f(x) = \sum_{j=1}^{10} x_j (c_j + \ln \frac{x_j}{x_1 + \cdots + x_{10}})$$

where c_j are defined in Problem 2.15.
Constraints:

$$\begin{aligned} & x_1 + 2x_2 + 2x_3 + x_6 + x_{10} - 2 = 0, \\ & x_4 + 2x_5 + x_6 + x_7 - 1 = 0, \\ & x_3 + x_7 + x_8 + 2x_9 + x_{10} = 0, \\ & \text{1.E-6} \le x_i,\ i = 1, \cdots, 10. \end{aligned}$$

Start: $x_0 = (0.1, \cdots, 0.1),\ f(x_0) = -20.961$
Solution:

$$\begin{aligned} x^* &= (0.01773548, 0.08200180, 0.8825646, 0.7233256\text{E-}3, \\ &\quad 0.4907851, 0.4335469\text{E-}3, 0.01727298, \\ &\quad 0.007765639, 0.01984929, 0.05269826), \\ f(x^*) &= -47.707579 \end{aligned}$$

Problem 2.17 (No. 117 in [176])
Number of Variables: $n = 15$
Objective Function:

$$f(x) = -\sum_{j=1}^{10} b_j x_j + \sum_{j=1}^{5} \sum_{k=1}^{5} c_{kj} x_{10+k} x_{10+j} + 2 \sum_{j=1}^{5} d_j x_{10+j}^3$$

Constraints:

$$2\sum_{k=1}^{5} c_{kj}x_{10+k} + 3d_j x_{10+j}^2 + e_j - \sum_{k=1}^{10} a_{kj}x_k \geq 0,\ j = 1, \cdots, 5,$$
$$0 \leq x_i,\ i = 1, \cdots, 15,$$

where

j	1	2	3	4	5
a_{6j}	2	0	-4	0	0
a_{7j}	-1	-1	-1	-1	-1
a_{8j}	-1	-2	-3	-2	-1
a_{9j}	1	2	3	4	5
a_{10j}	1	1	1	1	1
b_{5+j}	-1	-40	-60	5	1

and other parameters are defined as in Problem 2.11.

Start:

$$x_0 = 0.001(1, 1, 1, 1, 1, 1, 60000, 1, 1, 1, 1, 1, 1, 1, 1),$$
$$f(x_0) = 2400.1053$$

Solution:

$$\begin{aligned} x^* &= (0, 0, 5.174136, 0, 3.061093, 11.83968, 0, 0, \\ &\quad 0.1039071, 0, 0.2999929, 0.3334709, 0.3999910, \\ &\quad 0.4283145, 0.2239607) \\ f(x^*) &= 32.348679 \end{aligned}$$

Bibliography

[1] N. Abachi, On variable metric algorithms, *J. Optimization Theory and Methods* 7 (1971) 391-410.

[2] M. Al-Baali, Descent property and global convergence of the Fletcher-Reeves method with inexact line search, *IMA J. Numerical Analysis* 5 (1985) 121-124.

[3] K.A. Ariyawansa, Deriving collinear scaling algorithms as extensions of quasi-Newton methods and the local convergence of DFP- and BFGS-related collinear scaling algorithms, *Mathematical Programming* 49 (1990) 23-48.

[4] L. Armijo, Minimization of functions having Lipschitz continuous partial derivatives, *Pacific J. Mathematics* 16 (1966) 1-3.

[5] M. Avriel, *Nonlinear Programming: Analysis and Methods*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, (1976).

[6] P. Baptist and J. Stoer, On the relation between quadratic termination and convergence properties of minimization algorithms, Part II: Applications, *Numerische Mathematik* 28 (1977) 367-392.

[7] R. Bartels and A. Conn, An approach to nonlinear l_1 data fitting, in: J.P. Hennart ed., *Lecture Notes in Mathematics 909: Numerical Analysis, Cocoyoc 1981* (Springer-Verlag, Berlin, 1982), 48-58.

[8] J. Barzilai and J.M. Borwein, Two-point step size gradient methods, *IMA Journal of Numerical Analysis* 8 (1988) 141-148.

[9] M.S. Bazara and C.M. Shetty, *Nonlinear Programming, Theory and Algorithms*, John Wiley and Sons, New York, (1979).

[10] E.M.L. Beale, A derivative of conjugate gradients, in F.A. Lootsma eds., *Numerical Methods for Nonlinear Optimization*, London, Academic Press, (1972), 39-43.

[11] C.S. Beighter, D.T. Phillips and D.J. Wilde, *Foundations of Optimization*, Prentice-Hall, Englewood Cliffs, N.J., (1979).

[12] A. Ben-Israel and T.N.E. Greville, *Generalized Inverses: Theory and Applications*, John Wiley & Sons, New York, (1974).

[13] D.P. Bertsekas, *Constrained Optimization and Lagrange Multipler Methods*, Academic Press, New York, (1982).

[14] D.P. Bertsekas, Nonlinear Programming, Athena Scientific, Belmont, Mass., (1995).

[15] M.C. Bartholomew-Biggs, The estimation of the Hessian matrix in nonlinear least squares problems with non-zero residuals, Mathematical Programming 12 (1977) 67-80.

[16] M.C. Bartholomew-Biggs, Recursive quadratic programming methods based on the augmented Lagrangian, *Mathematical Programming Study* 31 (1987) 21-24.

[17] M.C. Biggs, Minimization algorithms making use of non-quadratic properties of the objective function, *Institute of Mathematics and Its Applications* 8 (1971) 315-327.

[18] A. Björck, *Numerical Methods for Least Squares Problems*, SIAM Publications, Philadelphia, Penn, (1996).

[19] P.T. Boggs and J.W. Tolle, Merit function for nonlinear programming problems, *Operations Research and System Analysis Report* No 81-2, University of North Carolina at Capel Hill, (1981).

[20] P.T. Boggs and J.W. Tolle, Convergence properties of a class of rank-two updates, *SIAM Journal on Optimization* 4 (1994) 262-287.

[21] P.T. Boggs and J.W. Tolle, Sequential quadratic programming, *Acta Numerica* 4 (1996) 1-51.

[22] P.T. Boggs, J.W. Tolle, and P. Wang, On the local convergence methods for constrained optimization, SIAM J. Control and Optimization 20 (1982) 161-171.

[23] I. Bongartz, A.R. Conn, N.I.M. Gould, and Ph.L. Toint, CUTE: Constrained and unconstrained testing environment, *ACM Transactions on Mathematical Software* 21 (1995) 123-160.

[24] C.A. Botsaris and D.H. Jacobson, A Newton-type curvilinear search method for optimization, *J. Mathematical Analysis and Applications* 54 (1976) 217–229.

[25] A. Bouaricha and R.B. Schnabel, Tensor methods for large sparse nonlinear least-squares problems, *SIAM J. Scientific Computing* 21 (1999) 1199-1221.

[26] C.G. Broyden, A class of methods for solving nonlinear simultaneous equations, *Mathematics of Computation* 19 (1965) 577-593.

[27] C.G. Broyden, The convergence of a class double-rank minimization algorithms, *Journal of the Institute of Mathematics and its Applications* 6 (1970) 76-90.

[28] C.G. Broyden, J.E. Dennis and J.J. Moré, On the local and superlinear convergence of quasi-Newton algorithm, *J. Inst. Math. Appl.* 12 (1973) 222-236.

[29] C.G. Broyden, J.E. Dennis, Jr., and J.J. Moré, On the local superlinear convergence of quasi-Newton methods, *J. Institute of Mathematics and Applications* 12 (1973) 223-246.

[30] A. Buckley, A combined conjugate gradient quasi-Newton minimization algorithm, *Mathematical Programming* 15 (1978) 200-210.

[31] J.P. Buleau and J.Ph. Vial, Curvilinear path and trust region in unconstrained optimization: a convergence analysis, *Mathematical Programming Study* 30 (1987) 82-101.

[32] J.R. Bunch and L. Kaufman, Some stable methods for calculating inertia and solving symmetric linear systems, *Mathematics of Computation* 31 (1977) 163-179.

[33] J.R. Bunch and B.N. Parlett, Direct methods for solving symmetric indefinite systems of linear equations, *SIAM Journal on Numerical Analysis* 8 (1971) 639-655.

[34] J.V. Burke, Descent methods for composite nondifferential optimization problems, *Mathematical Programming* 33 (1985) 260-279.

[35] J.V. Burke, Second order necessary and sufficient conditions for convex composite NDO, *Mathematical Programming* 38 (1987) 287-302.

[36] J.V. Burke, A robust trust region method for constrained nonlinear programming problems, *SIAM J. Optimization* 2 (1992) 325-347.

[37] J.V. Burke and J.J. Moré, On the identification of active constraints, *SIAM J. Numerical Analysis* 25 (1988) 1197-1211.

[38] J.V. Burke, J.J. Moré, and G. Toraldo, Convergence properties of trust region methods for linear and convex constraints, *Mathematical Programming* 47 (1990) 305-336.

[39] W. Burmeister, Die konvergenzordnung des Fletcher-Powell algorithmus, *Z. Angew. Math. Mech.* 53 (1973) 693-699.

[40] R.H. Byrd, An example of irregular convergence in some constrained optimization methods that use projected Hessian, *Mathematical Programming* 32 (1985) 232-237.

[41] R.H. Byrd, M.E. Hribar, and J. Nocedal, An interior-point algorithm for large-scale nonlinear programming, Technical Report 97/05, Optimization Technology Center, Argonne National Laboratory and Northwestern University, July (1997).

[42] R.H. Byrd, H.F. Khalfan, and R.B. Schnabel, Analysis of symmetric rank-one trust region method, *SIAM Journal on Optimization* 6 (1996) 1025-1039.

[43] R.H. Byrd, D.C. Liu, and J. Nocedal, On the behavior of Broydens class of quasi-Newton methods, *SIAM Journal on Optimization* (1992).

[44] R.H. Byrd and J. Nocedal, A tool for the analysis of quasi-Newton methods with application to unconstrained minimization, *SIAM Journal on Numerical Analysis* 26 (1989) 727-739.

[45] R.H. Byrd and J. Nocedal, An analysis of reduced Hessian methods for constrained optimization, *Mathematical Programming* 49 (1991) 285-323.

[46] R.H. Byrd, J. Nocedal, and R.B. Schnabel, Representations of quasi-Newton matrices and their use in limit-memory methods, *Mathematical Programming* 63 (1994) 129-156.

[47] R.H. Byrd, J. Nocedal and Y. Yuan, Global convergence of a class of variable metric algorithms, *SIAM J. Numerical Analysis* 24 (1987) 1171-1190.

[48] R.H. Byrd, R.B. Schnabel, and G.A. Schultz, A trust region algorithm for nonlinearly constrained optimization, *SIAM J. Numerical Analysis* 24 (1987) 1152-1170.

[49] R.H. Byrd, R.B. Schnabel, and G.A. Schultz, Approximate solution of the trust region problem by minimization over two-dimensional subspaces, *Mathematical Programming* 40 (1988) 247-263.

[50] P.H. Calamai and J.J. Moré, Projected gradient methods for linearly constrained problems, *Mathematical Programming* 39 (1987) 93-116.

[51] M.R. Celis, A trust region strategy for nonlinear equality constrained optimization, Ph.D. thesis, Dept of Math. Sci., Rice University, Houston, (1985).

[52] M.R. Celis, J.E. Dennis and R.A. Tapia, A trust region algorithm for nonlinear equality constrained optimization, in P.T. Boggs, R.H. Byrd and R.B. Schnabel, eds., *Numerical Optimization* (SIAM Philadelphia, 1985), 71-82.

[53] R.M. Chamberlain, Some examples of cycling in variable metric methods for constrained minimization, *Mathematical Programming* 16 (1979) 378-383.

[54] R.M. Chamberlain, C. Lemarechal, H.C. Pedersen, and M.J.D. Powell, The watchdog techniques for forcing convergence in algorithms for constrained optimization, *Mathematical Programming Study* 16 (1982) 1-17.

[55] C. Charelambous, Unconstrained optimization based on homogeneous models, *Mathematical Programming* 5 (1973) 189-198.

[56] C. Charelambous and A.R. Conn, An efficient method to solve the minimax problem directly, *SIAM J. Numerical Analysis* 15 (1978) 162-187.

[57] X. Chen, Superlinear convergence of smoothing quasi-Newton methods for nonsmooth equations, *J. of Computational and Applied Mathematics* 80 (1997) 105-126.

[58] E.W. Cheney and A.A. Goldstein, Newton's method for convex programming and Chebyshev approximation, *Numerische Mathematik* 1 (1959) 253-268.

[59] V. Chvatál, *Linear Programming*, W.M. Freeman and Company, New York, (1983).

[60] F.H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley and Sons, New York, (1983).

[61] A. Cohen, Rate of convergence of several conjugate gradient algorithms, *SIAM J. Numer Anal.* 9 (1972) 248-259.

[62] T.F. Coleman and A.R. Conn, Nonlinear programming via an exact penalty function: asymptotic analysis, *Mathematical Programming* 24 (1982) 123-136.

[63] T.F. Coleman and A.R. Conn, On the local convergence of a quasi-Newton method for the nonlinear programming problem, *SIAM J. Numerical Analysis* 21 (1984) 755-769.

[64] A.R. Conn, N.I.M. Gould, D. Orban, and Ph.L. Toint, A primal-dual trust region algorithm for nonconvex nonlinear programming, *Mathematical Programming* 87 (2000) 215-249.

[65] A.R. Conn, N.I.M. Gould, and Ph.L. Toint, Global convergence of a class of trust region algorithms for optimization with simple bounds, *SIAM J. on Numerical Analysis* 25 (1988) 433-460.

[66] A.R. Conn, N.I.M. Gould, and Ph.L. Toint, Testing a class of algorithms for solving minimization problems with simple bounds on the variables, *Mathematics of Computation* 50 (1988) 399-430.

[67] A.R. Conn, N.I.M. Gould, and Ph.L. Toint, Convergence of quasi-Newton matrices generated by symmetric rank one update, *Mathematical Programming* 50 (1991) 177-195.

[68] A.R. Conn, N.I.M. Gould, and Ph.L. Toint, LANCELOT: a FORTRAN package for large-scale nonlinear optimization (Release A), No. 17 in *Springer Series in Computational Mathematics*, Springer-Verlag, New York, (1992).

[69] A.R. Conn, N.I.M. Gould, and Ph.L. Toint, Convergence properties of minimization algorithms for convex constraints using a structured trust region, *SIAM Journal on Numerical Analysis* 25 (1996) 1059-1086.

[70] A.R. Conn, N.I.M. Gould and Ph.L. Toint, *Trust-Region Methods*, SIAM, (2000).

[71] G. Corradi, Quasi-Newton methods for nonlinear equations and unconstrained optimization methods, *International Journal of Computer Mathematics* 38 (1991) 71-89.

[72] C.W. Cryer, *Numerical Functional Analysis*, Clarendon Press, Oxford, (1982).

[73] Y.H. Dai, New properties of a nonlinear conjugate gradient method, *Numerische Mathematik* 89 (2001) 83-98.

[74] Y.H. Dai and Y. Yuan, Convergence properties of Fletcher–Reeves method, *IMA J. Numerical Analysis* 16 (1996) 155-164.

[75] Y.H. Dai and Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, *SIAM J. Optimization* 10 (1999) 177-182.

[76] Y.H. Dai and Y. Yuan, An efficient hybrid conjugate gradient method for unconstrained optimization, *Annals of Operations Research* 103 (2001) 33-47.

[77] Y.H. Dai and Y. Yuan, A three-parameter family of nonlinear conjugate gradient methods, *Mathematics of Computation* 70 (2001) 1155-1167.

[78] G.B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, New Jersey, (1963).

[79] W.C. Davidon, Variable metric methods for minimization, Argonne National Labs Report, ANL-5990, (1959).

[80] W.C. Davidon, Optimally conditioned optimization algorithms without line searches, *Mathematical Programming* 9 (1975) 1-30.

[81] W.C. Davidon, Optimization by nonlinear scaling, in: D. Jacobs ed., *Proceedings of the conference on Applications of Numerical Software — Needs and Availability*, Academic Press, New York, (1978), 377-383.

[82] W.C. Davidon, Conic approximation and Collinear scaling for optimizers, *SIAM Numer. Anal.* 17 (1980) 268-281.

[83] R.S.Dembo, S.C.Eisenstat, and T.Steihaug, Inexact Newton methods, *SIAM Journal on Numerical Analysis* 19 (1982) 400-408.

[84] V.F. Demyanov and L.V. Vaselev, *Nondifferentiable Optimization*, Optimization Software, Inc., New York, (1985).

[85] N.Y. Deng, *Computational Methods for Unconstrained Optimization*, Science Press, Beijing, (1982).

[86] N.Y. Deng, Y. Xiao and F. Zhou, A nonmonotonic trust region algorithm, *Journal of Optimization Theory and Applications* 76 (1993) 259-285.

[87] J.E. Dennis Jr., M. El-Alem, and M.C. Maciel, A global convergence theory for general trust region based algorithms for equality constrained optimization, *SIAM J. Optimization* 7 (1997) 177-207.

[88] J.E. Dennis Jr., D.M. Gay and R.E. Welsch, An adaptive nonlinear least-squares algorithm, ACM Transactions on Math. Software 7 (1981) 348-368.

[89] J.E. Dennis Jr., S.B. Li, and R.A. Tapia, A unified approach to global convergence of trust region methods for nonsmooth optimization, *Mathematical Programming* 68 (1995) 319-346.

[90] J.E. Dennis and H.H.W. Mei, Two new unconstrained optimization algorithms with use function and gradient values, *Journal of Optimization Theory and Applications* 28 (1979) 453-482.

[91] J.E. Dennis Jr., and J.J. Moré, A charaterization of superlinear convergence and its application to quasi-Newton methods, *Math. Comp.* 28 (1974) 549-560.

[92] J.E. Dennis Jr., and J.J. Moré, Quasi-Newton Methods, motivation and theory, *SIAM Review* 19 (1977) 46-89.

[93] J.E. Dennis Jr., and R.B. Schnabel, Least change secant updates for quasi-Newton methods, *SIAM Review* 19 (1979) 443-459.

[94] J.E. Dennis Jr., and R.B. Schnabel, A new derivation of symmetric positive definite secant updates, in: O.L. Mangasarian, R.R. Meyer and S.M. Robinson eds., *Nonlinear Programming* vol. 4, Academic Press, New York, (1980) 167-199.

[95] J.E. Dennis and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, (1983).

[96] J.E. Dennis and R.B. Schnabel, A view of unconstrained optimization, in: *Optimization Vol. 1 of Handbooks in Operations Research and Management*, Elsevier Science Publishers, Amsterdam, (1989) 1–72.

[97] J.E. Dennis Jr. and K. Turner, Generalized conjugate directions, Report 85-11, Dept of Mathmatics, Rice University, Houston, (1985).

[98] J.E.Dennis Jr. and H.F. Walker, Convergence theorems for least change secant update methods, *SIAM J. Numer. Anal.* 18 (1981) 949-987; 19 (1982) 443-443.

[99] J.E.Dennis Jr. and H.F. Walker, Least-change sparse secant update methods with inaccurate secant conditions, *SIAM J. Numer. Anal.* 22 (1985) 760-778.

[100] J.E. Dennis, Jr. and H. Wolkowicz, Sizing and least change secant methods, Research Report 90-02, Faculty of Mathematics, University of Waterloo, Canada, (1990).

[101] S. Di and W. Sun, Trust region method for conic model to solve unconstrained optimization problems, *Optimization Methods and Software* 6 (1996) 237–263.

[102] G. Di Pillo and L. Grippo, A new class of augmented Lagrangians in nonlinear programming, *SIAM J. Control and Optimization* 17 (1979) 618-828.

[103] G. Di Pillo and L. Grippo, An exact penalty function method with global convergence properties for nonlinear programming problem, *Math. Prog.* 36 (1986) 1-18.

[104] G. Di Pillo, L. Grippo and F. Lampariello, A class of algorithms for the solution of optimization problems with inequalities, CNR Inst. di Anal. dei Sistemi ed Inf. Report R18, (1981).

[105] L.C.W. Dixon, The choice of step length, a crucial factor in the performance of variable metric method, in: F.A. Lootsma, ed., *Numerical Methods for Nonlinear Optimization*, (Academic Press, London, 1972) 149–170.

[106] L.C.W. Dixon, Variable metric algorithms: necessary and sufficient conditions for identical behavior of nonquadratical functions, *J. Optimization Theory and Appl.* 10 (1972) 34–40.

[107] L.C.W.Dixon, Quasi-Newton family generates identical points, Part I and Part II, *Math. Prog.* 2 (1972) 383–387, 3 (1972) 345–358.

[108] L.C.W. Dixon, E. Spedicato and G.P. Szego, eds. *Nonlinear Optimization* Birkhauser, Boston, (1980).

[109] L.C.W. Dixon and G.P. Szegö, *Towards Global Optimization*, Vol. 1, Vol. 2, North-Holland, Amsterdam, (1975), (1978).

[110] I.S. Duff, J. Nocedal, and J.K. Reid, The use of linear programming for the solution of sparse sets of nonlinear equations, *SIAM J. Scientific and Statistical Computing* 8 (1987) 99–108.

[111] M. El-Alem, *A Global Convergence Theory for a Class of Trust Region Algorithms for Constrained Optimization*, Ph. D. Thesis, Dept of Mathematical Sciences, Rice University, Houston, (1988).

[112] M. El-Alem, A global convergence theory for the Celis-Dennis-Tapia trust region algorithm for constrained optimization, *SIAM J. Numerical Analysis* 28 (1991) 266–290.

[113] M. El-Alem, A robust trust region algorithm with nonmonotone penalty parameter scheme for constrained optimization, *SIAM J. Optimization* 5 (1995) 348–378.

[114] M. El-Hallabi, A global convergence theory for a class of trust region methods for nonsmooth optimization, Report MASC TR 90-16, Rice University, USA.

[115] M. El-Hallabi and Tapia, A global convergence theory for arbitrary norm trust-region methods for nonlinear equations, Report MASC TR 93-43, Rice University, Houston, USA.

[116] M. El-Hallabi and R.A. Tapia, An inexact trust regionfeasible-point algorithm for nonlinear systems and inequalities, Report MASC TR 95-09, Rice University, Houston, USA.

[117] I.I. Eremin, A generalization of the Motzkin-Agmon relaxation method, *Soviet Math. Doklady* 6 (1965) 219-221.

[118] Yu.M. Ermoliev, Method of solution of nonlinear extremal problems, (in Russian), *Kibernetika* 2 (1966) 1-17.

[119] D.J. Evans, W. Sun, R.J.B. Sampaio, and J. Yuan, Restricted generalized inverse corresponding to constrained quadratic system, *International Journal of Computer Mathematics* 62 (1996) 285-296.

[120] F. Facchinei and S. Lucidi, Nonmonotone bundle-type scheme for convex nonsmooth minimization, *J. Optimization Theory and Applications* 76 (1993) 241-257.

[121] Shu-Cheng Fang and S. Puthenpura, *Linear Programming and Extensions, Theory and Algorithms*, Prentice Hall, Inc., (1993).

[122] A.V. Fiacco and G.P. McCormick, *Nonlinear Programming: Sequential Unconstrained Mininization Techniques*, (John Wiley, New York 1968).

[123] J. Flachs, On the convergence, invariance, and related aspects of a modification of Huang's algorithm, *J. Optimization Theory and Methods* 37 (1982) 315-341.

[124] J. Flachs, On the generalization of updates for quasi-Newton method, *J. Optimization Theory and Applications* 48 (1986) 379-418.

[125] R. Fletcher, A new approach to variable metric algorithms, *Computer J.* 13 (1970) 317-322.

[126] R. Fletcher, An exact penalty function for nonlinear programming with inequalities, *Math. Prog.* 5 (1973) 129-150.

[127] R. Fletcher, An ideal penalty function for constrained optimization, *J. Inst. Math. Applications* 15 (1975) 319-342.

[128] R. Fletcher, *Practical Methods of Optimization, Vol. 1, Unconstrained Optimization*, (John Wiley and Sons, Chichester, 1980).

[129] R. Fletcher, *Practical Methods of Optimization, Vol. 2, Constrained Optimization*, (John Wiley and Sons, Chichester, 1981).

[130] R. Fletcher, A model algorithm for composite NDO problem, *Math. Prog. Study* 17 (1982) 67-76. (1982a)

[131] R. Fletcher, Second order correction for nondifferentiable optimization, in: G.A. Watson, ed., *Numerical Analysis*, (Springer-Verlag, Berlin, 1982), 85-115. (1982b).

[132] R. Fletcher, Penalty functions, in: A. Bachem, M. Grötschel and B. Korte, eds., *Mathematical Programming: The State of the Art*, (Springer-Verlag, Berlin, 1983), 87-114.

[133] R. Fletcher, *Practical Methods of Optimization* (second edition), (John Wiley and Sons, Chichester, 1987).

[134] R. Fletcher, An optimal positive definite update for sparse hessian matrices, *SIAM Journal on Optimization* 5 (1995) 192-218.

[135] R. Fletcher and T.L. Freeman, A modified Newton method for minimization, *J. Optimization Theory and Methods* 23 (1977) 357-372.

[136] R. Fletcher, S. Leyffer, and Ph.L. Toint, On the global convergence of a filter-SQP algorithm, *SIAM J. Optimization* No.1 (2002) 44-59.

[137] R. Fletcher and M.J.D. Powell, A rapid convergent descent method for minimization, *Computer Journal* 6 (1963) 163-168.

[138] R. Fletcher and C.M. Reeves, Function minimization by conjugate gradients, *Computer Journal* 7 (1964) 149-154.

[139] R. Fletcher and C. Xu, Hybrid methods of nonlinear least squares, IMA J. of Numerical Analysis 7 (1987).

[140] J.A. Ford and R.A. Ghundhari, On the use of curvature estimates in quasi-Newton methods, *J. Comput. Appl. Math.* 35 (1991) 185-196.

[141] M. Fukushima, A descent algorithm for non-smooth convex programming, *Mathematical Programming* 30 (1984) 163-175.

[142] M. Fukushima, A succesive quadratic programming algorithm with global and superlinear convergence properties, *Mathematical Programming* 35 (1986) 253-264.

[143] M. Fukushima and L. Qi, A globally and superlinearly convergent algorithm for nonsmooth convex minimization, *SIAM J. Optimization* 6 (1996) 1106-1120.

[144] D.M. Gay, Computing optimal local constrained step, *SIAM J. Sci. Stat. Comp.* 2 (1981) 186-197.

[145] D.M. Gay, A trust region approach to linearly constrained optimization, in: D.F. Griffiths, ed., *Lecture Notes in Mathematics 1066: Numerical Analysis*, Springer-Verlag, Berlin, (1984) 72-105.

[146] J.C. Gilbert and J. Nocedal, Global convergence properties of conjugate gradient methods for optimization, *SIAM J. Optimization* 2 (1992) 21-42.

[147] P.E. Gill and W. Murray, Quasi-Newton methods for unconstrained optimization, *J. Inst. Maths. Appli.* 9 (1972) 91-108.

[148] P.E. Gill, G.H. Golub, W. Murray, and M.A. Sauders, Methods for modifying matrix factorizations, *Mathematics of Computation* 28 (1974) 505-535.

[149] P.E. Gill and W. Murray, Newton-type methods for unconstrained and linearly constrained optimization, *Mathematical Programming* 28 (1974) 311-350.

[150] P.E. Gill and W. Murray, Numerically stable methods for quadratic programming, *Math. Prog.* 14 (1978) 348-372.

[151] P.E. Gill and W. Murray, Conjugate gradient methods for large-scale nonlinear optimization, Technical Report SOL 79-15, Department of Operations Research, Stanford University, Stanford, California, (1979).

[152] P.E. Gill and W. Murray and M.H. Wright, *Practical Optimization*, Academic Press, London, (1981).

[153] D. Goldfarb, A family of variable metric methods derived by variation mean, *Mathematics of Computation* 23 (1970) 23-26.

[154] D.Goldfarb, Curvilinear path steplength algorithms for minimization which use directions of negative curvature, *Mathematical Programming* 18 (1980) 31-40.

[155] D. Goldfarb and A. Idinani, A numerical stable dual method for solving strictly convex quadratic programs, *Math. Prog.* 27 (1983) 1-33.

[156] S.M. Goldfeld, R.E. Quandt, and H.F. Trotter, Maximisation by quadratic hill-climbing, *Econometrica* 34 (1966) 541-551.

[157] A.A. Goldstein, On steepest descent, *SIAM J. Control* 3 (1965) 147-151.

[158] A.A. Goldstein, *Constructive Real Analysis*, Harper & Row, New York, (1967).

[159] A.A. Goldstein and J.F. Price, An effective algorithm for minimization, *Numer. Math.* 10 (1967) 184-189.

[160] G.H. Golub and C.F.Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 3rd ed., (1996).

[161] N.I.M. Gould, D. Orban, Ph.L. Toint, CUTEr and SifDec: A constrained and unconstrained testing environment revisited, *ACM Transactions on Mathematical Software* 29 (2003) 373-394.

[162] L. Grandinetti, Some investigation in a new algorithm for nonlinear optimization based on conic model of objective function, *J. Optimization Theory and Applications* 43 (1984) 1-21.

[163] J. Greenstadt, Variations on variable metric methods, *Mathematics of Computation* 24 (1970) 1-22.

[164] L. Grippo, F. Lampariello and S. Lucidi, A nonmonotone line search technique for Newton's methods, *SIAM J. Numer. Anal.* 23 (1986) 707-716.

[165] L. Grippo and S. Lucidi, A globally convergence version of the Polak–Ribiere conjugate gradient method, *Mathematical Programming* 78 (1997) 375-391.

[166] J. Hald and K. Madsen, Combined LP and quasi-Newton methods for minmax, *Math. Prog.* 20 (1981) 49-62.

[167] D. Han and W. Sun, New decomposition methods for solving variable inequality problems, *Mathematics and Computer Modeling* 37 (2003) 408–418.

[168] Q. Han, W. Sun, J. Han and R.J.B. Sampaio, An adaptive conic trust-region method for unconstrained optimization, *Optimization Methods and Software* 20 (2005) 645-663.

[169] S.P. Han, A global convergent method for nonlinear programming, *J. Optimization Theory and Applications* 22 (1977) 297-309.

[170] S.P. Han, J.S. Pang and N. Rangaraj, Globally convergent Newton methods for nonsmooth equations, *Mathematics of Operations Research*, 17 (1992) 586-607.

[171] X. He and W. Sun, Analysis on Greville's method, *Journal of Nanjing University, Mathematical Biquarterly* 5 (1988) 1-10.

[172] X. He and W. Sun, *Introduction to Generalized Inverses of Matrices*, Jiangsu Sci. & Tech. Publishing House, Nanjing, (1991). (in Chinese).

[173] M.R. Hestenes and E. Stiefel, Method of conjugate gradient for solving linear system, *J. Res. Nat. Bur. Stand.* 49 (1952) 409-436.

[174] D.M. Himmeblau, *Applied Nonlinear Programming*, McGraw-Hill, (1972).

[175] J.B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, New York, (1993).

[176] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economical and Mathematical Systems 187, Springer-Verlag, Berlin, (1981).

[177] E. Höpfinger, On the solution of the unidimensional local minimization problem, *J. Optimization Theory and Appl.* 18 (1976) 425-428.

[178] R. Hooke and T.A. Jeeves, Direct search solution of numerical and statistical problems, *J. ACM* 8 (1961) 212-229.

[179] Yuda Hu, *Nonlinear Programming*, Higher Education Press, Beijing, (1990). (in Chinese).

[180] H.Y. Huang, Unified approach to quadratically convergent algorithms for function minimization, *J. Optimization Theory and Appl.* 5 (1970) 405-423.

[181] D.H. Jacobson and W. Oxman, An algorithm that minimizes homogeneous functions of n variables in $n+2$ iterations and rapidly minimizes general functions, *J. Math. Anal. Appl.* 38 (1972) 533-552.

[182] D.H. Jacobson and L.M. Pels, A modified homogeneous algorithm for function minimization, *J. Math. Anal. Appl.* 46 (1974) 533-541.

[183] F. John, Extremum problem with inequalities as subsidiary conditions, in: F.D. Friedrichs, et al. (eds.) *Studies and Essays, Courant Anniversary Volume* (Interscience Publishers, New York, 1948)

[184] N. Karmarkar, A new polynomial-time algorithm for linear programming, *Combinatorica*, 4 (1984) 374-395.

[185] W. Karush, *Minima of functions of several variables with inequalities as side conditions*, Master's thesis, University of Chicago, Chicago, Illinois, (1939).

[186] J.E. Kelley, The cutting plane method for solving convex programs, *J. of SIAM* 8 (1960) 703-712.

[187] J.E. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM Publications, Philadelphia, Penn., (1995).

[188] L.G. Khachiyan, A polynomial algorithm in linear programming, *Soviet Mathematics Doklady* 20 (1979) 191-194.

[189] H.F.H. Khalfan, *Topics in quasi-Newton methods for unconstrained optimization*, Ph D thesis, University of Colorado, (1989).

[190] K.C. Kiwiel, *Methods of Descent for Nondifferentiable Optimization*, Lecture Notes in Mathematics 1133, Springer-Verlag, Berlin, (1985).

[191] M. Kojima and S. Shindo, Extensions of Newton and quasi-Newton methods to systems of PC^1 equations, *J. Oper. Res. Soc. Japan* 29 (1986) 352-374.

[192] J. Kowalik and K. Ramakrishnan, A numerically stable optimization method based on homogeneous function, *Math. Prog.* 11 (1976) 50-66.

[193] H.W. Kuhn and A.W. Tucker, Nonlinear programming, in: J. Neyman, ed., *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability* (University of California Press, Berkeley, California, 1951) 481-492.

[194] C.J.L. Lagrange, Essai dune nouvelle methods pour deteminer les maxima et les minima, *Miscellanea Taurinensia* 2 (1760-61) *Oeuvres*, 1, pp. 356-357, 360.

[195] C. Lemaréchal, Bundle methods in nonsmooth optimization, in: C. Lemaréchal and R. Mifflin, eds., *Nonsmooth Optimization* (Pergamon, Oxford, 1978) 79-102.

[196] C. Lemaréchal, Nondifferentiable optimization, in: L.C.W. Dixon, E. Spedicato and G.P. Szego, eds., *Nonlinear Optimization* (Birkhauser, Boston, 1980) 149-199.

[197] C.Lemarechal and C.Sagastizabal, Variational metric bundle methods: From conceptual to implementable forms, *Mathematical Programming* B, 76 (1997) 393-410.

[198] C. Lemarechal and C. Sagastizabal, Practical aspects of the Moreau Yosida relarization: Theoretical preliminaries, *SIAM J. Optimization* 7 (1997).

[199] K. Levenberg, A method for the solution of certain nonlinear problems in least squares, *Qart. Appl. Math.* 2 (1944) 164-166.

[200] D.C. Liu and J. Nocedál, On the limited memory BFGS method for large scale optimization, *Mathematical Programming* 45 (1989) 503-528.

[201] D.G. Luenberger, *Linear and Nonlinear Programming* (2nd Edition), (Addison-Wesley, Massachusetts, 1984).

[202] Z.Q. Luo, J.S. Pang and D. Ralph, *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, Cambridge, (1996).

[203] G. McCormick, A modification of Armijio's step-size rule for negative curvature, *Mathematical Programming* 13 (1977) 111-115.

[204] G. McCormick, *Nonlinear Programming: Theory, Algorithms, and Applications*, (John Wiley and Sons, New York, 1983)

[205] G. McCormick and K. Ritter, Alternative proofs of the convergence properties of the conjugate gradient method, *J. Optimization Theory and Applications* 13 (1974) 497-518.

[206] K. Madsen, An algorithm for the minimax solution of overdetermined systems of nonlinear equations, *J. Inst. Math. Appl.* 16 (1975) 1-20.

[207] O.L Mangasarian, *Nonlinear Programming*, (McGraw-Hill, New York, 1969).

[208] O.L. Mangasarian and S. Fromowitz, The Fritz John necessary optimality conditions in the presence of equality and inequality constraints, *J. Math. Anal. Appl.* 17 (1967) 37-47.

[209] N. Maratos, *Exact Penalty Function Algorithms for Finite Dimensional and Control Optimization Problems*, Ph. D. thesis, Imperial College Sci. Tech., University of London, (1978).

[210] D.W. Marquardt, An algorithm for least-squares estimation of nonlinear inequalities, *SIAM J. Appl. Math.* 11 (1963) 431-441.

[211] J.M. Martinez, Quasi-Newton methods with factorization scaling for solving sparse nonlinear systems of equations, *Computing* 38 (1987) 133-144.

[212] J.M. Martinez and A.C. Moretti, A trust region method for minimization of nonsmooth functions with linear constraints, *Mathematical Programming* 76 (1997) 431-449.

[213] J.M. Martinez and S.A. Santos, A trust-region strategy for minimization on arbitrary domains, *Mathematical Programming* 68 (1995) 267-301.

[214] D.Q. Mayne and E. Polak, A superlinearly convergent algorithm for constrained optimization problems, *Math. Prog. Study* 16 (1982) 45-61.

[215] R.R. Meyer, Theoretical and computational aspects of nonlinear regression, in: J. Rosen, O. Mangasarian and K. Ritter eds., *Nonlinear Programming*, (Academic Press, London, 1970), 465-486.

[216] R. Mifflin, An algorithm for constrained optimization with semismooth functions, *Math. Oper.Research* 2 (1977) 197-207.

[217] R. Mifflin, Semismooth and semiconvex function in constrained optimization, *SIAM J. Control and Optimization* 15 (1977) 957-972.

[218] J.J. Moré, The Levenberg-Marquardt algorithm: implementation and theory, in: G.A. Watson, ed., *Lecture Notes in Mathematics 630: Numerical Analysis* (Springer-Verlag, Berlin, 1978) 105-116.

[219] J.J. Moré, Recent developments in algorithms and software for trust region methods, in: A. Bachem, M. Grötschel and B. Korte, eds., *Mathematical Programming: The State of the Art* (Springer-Verlag, Berlin, 1983) 258-287.

[220] J.J. Moré, B.S. Garbow and K.E. Hilstrom, Testing unconstrained optimization software, *ACM Transactions on Mathematical Software* 7 (1983) 17-41; 9 (1983) 503-524.

[221] J.J. Moré and D.C.Sorensen, On the use of directions of negative curvature in adified Newton method, *Mathematical Programming* 16 (1979) 1-20.

[222] J.J. Moré and D.C. Sorensen, Computing a trust region step, *SIAM J. Sci. Stat. Comp.* 4 (1983) 553-572.

[223] T. Motzkin and I.J. Schoenberg, The relaxation method for linear inequalities, *Candian J. Math.* 6 (1954) 393-404.

[224] W. Murray and M.L. Overton, A projected Lagrangian algorithm for nonlinear minimax optimization, *SIAM J. Sci. Stat. Comp.* 1 (1980) 345-370.

[225] W. Murray and M.L. Overton, A projected Lagrangian algorithm for nonlinear L_1 optimization, *SIAM J. Sci. Stat. Comp.* 2 (1981) 207-224.

[226] B.A. Murtagh, and R.H.W. Sargent, A constrained minimization method with quadratic convergence, in: R. Fletcher, ed., *Optimization* (Academic Press, London, 1969) 215-346.

[227] S.G. Nash, Preconditioning of truncated-Newton methods, *SIAM J.Scientific Statistics and Computing* 6 (1985) 599-616.

[228] S.G. Nash and J. Nocedal, A numerical study of the limited memory BFGS method and truncated-Newton method for large-scale optimization, *SIAM J. Optimization* 1 (1991) 358-372.

[229] S.G. Nash, A survey of truncated-Newton methods, *Journal of Computational and Applied Mathematics* 124 (2000) 45-59.

[230] L. Nazareth, A relationship between the BFGS and conjugate gradient algorithms and its implications for new algorithms, *SIAM J. Numer. Anal.* 16 (1979) 794-800.

[231] L. Nazareth, Some recent approaches to solving large residual nonlinear least squares problems, *SIAM Review* 22 (1980) 1-11.

[232] J. Nocedal and M.L. Overton, Projected Hessian update algorithms for nonlinear constrained optimization, *SIAM J. Numer. Anal.* 22 (1985) 821-850.

[233] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer, New York, (1999).

[234] J. Nocedal and Y. Yuan, Analysis of a self-scaling quasi-Newton method, *Math. Prog.* 61 (1993) 19-37.

[235] E.A. Nurminski, (ed.) *Progress on Nondifferentiable Optimization* (IIASA, Laxenburg, 1982)

[236] .O. Omojokun, *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*, Ph. D. Thesis, University of Colorado at Boulder, (1989).

[237] S.S. Oren, *Self-scaling variable metric algorithm for unconstrained minimization*, Ph.D. Dissertation, Computer Science Department, Stanford University, USA, (1972).

[238] S.S. Oren, Self-scaling variable metric algorithm without line-search for unconstrained minimization, *Mathematics of Computation* 27 (1973) 873-885.

[239] S.S. Oren, On the selection of parameters in self-scaling variable metric algorithms, *Mathematical Programming* 7 (1974) 351-367.

[240] S.S. Oren, Self-scaling variable metric algorithm II: Implementation and experiments, *Management Science* 20 (1974) 863-874.

[241] S.S. Oren, Perspectives on self-scaling variable metric algorithms, *J. Optimization Theory and Methods* 37 (1982) 137-147.

[242] S.S. Oren, Planar quasi-Newton algorithm for unconstrained saddle point problems, *J. Optimization Theory and Methods* 43 (1984) 167-204.

[243] S.S. Oren and D.G. Luenberber, Self-scaling variable metric (SSVM) algorithm I: Criteria and sufficient conditions for scaling a class of algorithms, *Management Science* 20 (1974) 845-862.

[244] S.S. Oren and E. Spedicato, Optimal conditioning of self-scaling variable metric algorithm, *Math. Prog.* 10 (1976) 70-90.

[245] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, (Academic Press, New York, 1970).

[246] M.L. Overton, Algorithms for nonlinear l_1 and l_∞ fitting, in: M.J.D. Powell, ed., *Nonlinear Optimization 1981* (Academic Press, London, 1982).

[247] J.S. Pang, Newton's method for B-differentiable equations, *Mathematics of Operations Research* 15 (1990) 311-341.

[248] J.S. Pang, S.H. Han and N. Rangaraj, Minimization of locally Lipschitzian functions, *SIAM J. Optimization* 1 (1991) 57-82.

[249] J.S, Pang, A B-differentiable equation based, globally and locally quadratic convergent algorithm for nonlinear programs, complementarity and variable inequality problems, *Mathematical Programming* 51 (1991) 101-131.

[250] J.S. Pang and L. Qi, Nonsmooth equations: Motivation and Algorithms, *SIAM J. Optimization*

[251] J.S. Pang and L. Qi, A globally convergent Newton method for convex SC^1 minimization problems, *J. Optimization Theory and Applications* 85 (1995) 633-648.

[252] J.D. Pearson, Variable metric methods of minimization, *The Computer J.* 12 (1969) 171-178.

[253] E. Polak and G. Ribière, Note sur la convergence de directions conjugées, *Rev. Francaise Informat. Recherche Operationelle*, 3e année 16 (1969) 35-43.

[254] B.T. Polyak, A general method of solving extremal problems, *Soviet Math. Doklady* 8 (1967) 14-29.

[255] B.T. Polyak, The conjugate gradient method in extremum problems, *USSR Comp. Math. and Math. Phys.* 9 (1969) 94-112.

[256] B.T. Polyak, Subgradient methods: A survey of Soviet research, in: C. Lemarechal and R. Mifflin, eds., *Nonsmooth Optimization* (Pergamon, Oxford, 1978) 5-30.

[257] M.J.D. Powell, An efficient method for finding the minum of a function of several variables without calculating derivatives, *The Computer J.* 7 (1964) 155-162.

[258] M.J.D. Powell, On the calculation of orthogonal vectors, *Computer J.* 11 (1968) 302-304.

[259] M.J.D. Powell, A theory on rank one modifications to a matrix and its inverse, *The Computer J.* 12 (1969) 288-290.

[260] M.J.D. Powell, A new algorithm for unconstrained optimization, in: J.B. Rosen, O.L. Mangasarian and K. Ritter, eds., *Nonlinear Programming* (Academic Press, New York, 1970) 31-66.

[261] M.J.D. Powell, A hybird method for nonlinear equations, in: P. Robinowitz, ed., *Numerical Methods for Nonlinear Algebraic Equations* (Gordon and Breach Science, London, 1970) 87-144.

[262] M.J.D. Powell, On the convergence of the variable metric algorithm, *J. Inst. Maths. Appl.* 7 (1971) 21-36.

[263] M.J.D. Powell, Quadratic termination properties of minimization algorithms, Part I and Part II, *J. Inst. Maths. Appl.* 10 (1972) 332-357.

[264] M.J.D. Powell, Convergence properties of a class of minimization algorithms, in: O.L. Mangasarian, R.R. Meyer and S.M. Robinson, eds., *Nonlinear Programming 2* (Academic Press, New York, 1975) 1-27.

[265] M.J.D. Powell, Some global convergence properties of a variable metric algorithm for minimization without exact line searches, in: R.W. Cottle and C.E. Lemke, eds., *Noninear Programming, SIAM-AMS Proceedings vol. IX* (SIAM publications, Philadelphia, 1976) 53-72. (1976a).

[266] M.J.D. Powell, Some convergence properties of the conjugate gradient method, *Math. Prog.* 11 (1976) 42-49. (1976b)

[267] M.J.D. Powell, Restart procedure for the conjugate gradient method *Mathematical Programming* 12 (1977) 241-254.

[268] M.J.D. Powell, A fast algorithm for nonlinearly constrained optimization calculations, in: G.A. Watson, ed., *Numerical Analysis* (Springer-Verlag, Berlin, 1978) 144-157.

[269] M.J.D. Powell, VMCWD: A FORTRAN subroutine for constrained optimization, DAMTP Report 1982/NA4, University of Cambridge, England (1982).

[270] M.J.D. Powell, Nonconvex minimization calculations and the conjugate gradient method, in: D.F. Griffiths, ed., *Numerical Analysis* Lecture Notes in Mathematics 1066 (Springer-Verlag, Berlin, 1984) pp. 122-141.

[271] M.J.D. Powell, On the rate of convergence of variable metric algorithms for unconstrained optimization, in: Z. Ciesielki and C. Olech, eds., *Proceeding of the International Congress of Mathematicians* (Elsevier, New York, 1984) 1525-1539.

[272] M.J.D. Powell, General algorithms for discrete nonlinear approximation calculations, in: L.L. Schumacher, ed., *Approximation Theory IV* (Academic Press, New York, 1984) 187-218.

[273] M.J.D. Powell, On the global convergence of trust region algorithms for unconstrained optimization, *Math. Prog.* 29 (1984) 297-303.

[274] M.J.D. Powell, On the quadratic programming algorithm of Goldfarb and Idnani, *Math. Prog. Study* 25 (1985) 46-61.

[275] M.J.D. Powell, Updating conjugate directions by the BFGS formular, *Mathematical Programming* 38 (1987) 29-46.

[276] M.J.D. Powell and Ph.L. Toint, On the estimation of sparse Hessian matrices, *SIAM J. Numer. Anal.* 16 (1979) 1060-1074.

[277] M.J.D. Powell and Y. Yuan, A recursive quadratic programming algorithm that uses differentiable exact penalty function, *Mathematical Programming* 35 (1986) 265-278.

[278] M.J.D. Powell and Y. Yuan, A trust region algorithm for equality constrained optimization, *Mathematical Programming* 49 (1991) 189-211.

[279] L. Qi, Convergence analysis of some algorithms for solving nonsmoth equations, *Mathematics of Operations Research* 18 (1993) 227-244.

[280] L. Qi, Trust region algorithms for solving nonsmooth equations, *SIAM J. Optimization* 5 (1995) 219-230.

[281] L. Qi and X. Chen, A globally convergent successive approximation method for severely nonsmooth equations, *SIAM J. Control and Optimization* 33 (1995) 402-418.

[282] L. Qi and D. Sun, A survey of some nonsmooth equations and smoothing Newton methods, in: A. Eberhard, R. Hill, D. Ralph and B.M. Glover eds., *Progress in Optimization*, Kluwer Academic, Dordrecht, (1999), 121-146.

[283] L. Qi and J. Sun, A nonsmooth version of Newton's method, *Mathematical Programming* 58 (1993) 353-367.

[284] L. Qi and W. Sun, An iterative method for the minimax problem, in D.Z. Du and P.M. Pardalos eds., *Minimax and Applications*, Kluwer Academic Publisher, Boston, (1995), 55-67.

[285] D. Ralph, Global convergence of damped Newton's method for nonsmooth equations, via the path search, *Mathematics of Operations Research*.

[286] F. Rendle and H. Wolkowicz, A semidefinite framework for trust region subproblems with applications to large scale minimization, *Mathematical Programming* 77 (1997) 273-299.

[287] K. Ritter, On the rate of superlinear convergence of a class of variable metric methods, *Numerische Mathematik* 35 (1980) 293-313.

[288] R.T. Rockafellar, *Convex Analysis* (Princeton University Press, Princeton, 1970).

[289] R.T. Rockafellar, Augmented Lagrangians and applications of the proximal point algorithm in convex programming, *Mathematics of Operations Research* 1 (1976) 97-116. (1976a).

[290] R.T. Rockafellar, Monotone operators and the proximal point algorithm. *SIAM J. Control and Optimization* 14 (1976) 877-898.

[291] R.T. Rockafellar, *The Theory of Subgradient and Its Application to Problems of Optimization: Convex and Not Convex Functions* (Heldermann-Verlag, West Berlin, 1981).

[292] R.T. Rockafellar, Computational scheme for solving large-scale problems in extended linear-quadratic programming, *Mathematical Programming* 48 (1990) 447-474.

[293] R.T. Rockafellar and R.J.B. Wets, *Variational Analysis*, Springer-Verlag, Berlin, (1998).

[294] J.B. Rosen, The gradient projection method for nonlinear programming, Part 1: Linear constraints, *J. SIAM* 8 (1960) 181-217.

[295] J.B. Rosen, The gradient projection method for nonlinear programming, Part 2: Nonlinear constraints, *J. SIAM* 9 (1961) 514-532.

[296] R.J.B. Sampaio, W. Sun, and J. Yuan, On the trust region algorithm for nonsmooth optimization, *Applied Mathematics and Computation* 85 (1997) 109-116.

[297] K. Schittkowski, The nonlinear programming method of Wilson, Han and Powell with an augmented Lagragian type line search function, Part 1: convergence analysis, *Numerische Mathematik* 38 (1981) 83-114.

[298] K. Schittkowski, More test examples for nonlinear programming codes, *Lecture Notes in Economics and Mathematical System* 282, Springer-Verlag, Berlin, (1987).

[299] R.B. Schnabel, *Analysing and improving quasi-Newton methods for unconstrained optimization*, PhD thesis, Department of Computer Science, Cornell University, Ithaca, NY, (1977).

[300] R.B. Schnabel, Conic methods for unconstrained optimization and tensor methods for nonlinear equations, in: A. Bachem, M. Grotschel and B. Korte eds., *Mathematical Programming, The State of the Art* (Springer-Verlag, Berlin, 1983) 417-438.

[301] R.B. Schnabel and Ta-Tung Chow, Tensor methods for unconstrained optimization using second derivatives, *SIAM J. Optimization* 1 (1991) 293-315.

[302] R.B. Schnabel and P.D. Frank, Tensor methods for nonlinear equations, *SIAM J.Numer.Anal.* 21 (1984) 815-843.

[303] L.K. Schubert, Modification of a quasi-Newton method for nonlinear equations with sparse Jacobian, *Mathematics of Computation* 24 (1970) 27-30.

[304] D.F. Shanno, Conditioning of quasi-Newton methods for function minimization, *Math. Comput.* 24 (1970) 647-656.

[305] D.F. Shanno, Conjugate gradient methods with inexact searches, *Math. Oper. Res.* 3 (1978) 244-256.

[306] D.F. Shanno and K.H. Phua, Matrix conditioning and nonlinear optimization, *Mathematical Programming* 14 (1978) 149-160.

[307] D.F. Shanno and K.H. Phua, Remark on Algorithm 500: Minimization of unconstrained multivariate functions, *ACM Transactions on Mathematical Software* 6 (1980) 618-622.

[308] S. Sheng, A class of collinear scaling algorithm for unconstrained optimization, *Numerical Mathematics, A Journal of Chinese Universities* 6 (1997) 219-230.

[309] N.Z. Shor, An application of the method of gradient descent to the solution of the network transpotation problems (in Russian), in: Notes Scientific Seminar on Theory and Application of Cybernetics and Operations Research (Acdemy of Science, Ukrain, SSSR, 1 (1962) 9-17.

[310] N.Z. Shor, On the speed of convergence of the generalized gradient, *Kibernetika* 3 (1968) 98-99.

[311] N.Z. Shor, An application of the operation of space dilation to the problems of minimizing convex functions, *Kibernetika* 1 (1970) 6-12.

[312] N.Z. Shor, Generalized gradient methods of nondifferentiable optimization emplying space dilation operations, in: A. Bachem, M. Grotchel, and B. Korte, eds., *Mathematical Programming, The State of the Art*, Springer-Verlag, Berlin, (1983), 501-529.

[313] N.Z. Shor, *Minimization Methods for Non-differentiable Functions*, Springer-Verlag, Berlin, (1985).

[314] G.A. Shultz, R.B. Schnable and R.H. Byrd, A family of trust region based algorithms for unconstrained minimization with strong global convergence properties, *SIAM J. Numerical Analysis* 22 (1985) 47-67.

[315] D.C. Sorensen, The q-superlinear convergence of a collinear scaling algorithm for unconstrained optimization, *SIAM J.Numer.Anal.* 17 (1980) 84-114.

[316] D.C. Sorensen, Newton's method with a model trust region modification, *SIAM J. Numer. Anal.* 20 (1982) 409-426.

[317] D.C. Sorensen, Trust region methods for unconstrained optimization, in: M.J.D. Powell, ed., *Nonlinear Optimization 1981* (Academic Press, London, 1982) 29-38. (1982b).

[318] E. Spedicato, A variable metric method for function minimization derived from invariancy to nonlinear scaling, *J. Optimization Theory and its Applications* (1976).

[319] E. Spedicato, A note on the determination of the scaling parameters in a class of quasi-Newton methods for unconstrained optimization, *J. Inst. Maths. Applics.* 21 (1978) 285-291.

[320] E. Spedicato and Zunquan Xia, Finding general solutions of the quasi-Newton equation via the ABS approach, *Optimization Methods and Software* 1 (1992) 243-252.

[321] T. Steihaug, *Quasi-Newton methods for large scale optimization*, Ph. D. Dissertation, SOM Technical Report No.49, Yale University. (1980).

[322] T. Steihaug, The conjugate gradient and trust regions in large scale optimization, *SIAM Journal on Numerical Analysis* 20 (1983) 626-637.

[323] G.W. Stewart, A modification of Davidon's method to accept difference approximation of derivatives, *J. ACM* 14 (1967) 72-83.

[324] J. Stoer, On the convergence rate of imperfect minimization algorithms in broyden's β class", *Math. Prog.*, 9 (1975) 313-335.

[325] J. Stoer, On the relation between quadratic termination and convergence properties of minimization algorithms, Part I: Theory, *Numerische Mathematik* 28 (1977) 343-366.

[326] J. Stoer, Foundations of recusive quadratic programming methods for solving nonlinear programs, in: K. Schittkowski, ed. *Computational Mathematical Programming* (Springer-Verlag, Berlin, 1985) 165-207.

[327] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, (Springer-Verlag, New York, 1993).

[328] J. Stoer, High order long-step methods for solving linear complementarity problems, *Annals of Operations Research* 103 (2001) 149-159.

[329] J. Stoer, M. Wechs and S. Mizuno, High order infeasible-interior-point methods for solving sufficient linear complementarity problems, *Mathematics of Operations Research* 23 (1998) 832-862.

[330] W. Sun and X. Chang, An unconstrained minimization method based on homogeneous functions, *Journal of Applied Mathematics & Computational Mathematics* 3 (1989) 81-88.

[331] W. Sun and Z. Wu, Numerical research on self-scaling variable metric algorithm, *Numerical Mathematics, A Journal of Chinese Universities* 11 (1989) 145-158.

[332] W. Sun, Generalized Newton method for LC^1 unconstrained optimization, *Journal of Computational Mathematics* 15 (1995) 502-508.

[333] W. Sun, On nonquadratic model optimization methods, *Asia and Pacific Journal of Operations Research* 13 (1996) 43-63.

[334] W. Sun, On convergence of an iterative method for minimax problem, *Journal of Australian Mathematics Society*, Series B, 39 (1997) 280-292.

[335] W. Sun, Newton's method and quasi-Newton-SQP method for general LC^1 constrained optimization, *Applied Mathematics and Computation*, 92 (1998) 69-84.

[336] W. Sun and Y. Wei, Inverse order rule for weighted generalized inverse, *SIAM Matrix Analysis and Applications* 19 (1998) 772-775.

[337] W. Sun and Y. Yuan, A conic trust-region method for nonlinearly constrained optimization, *Annals of Operations Research* 103 (2001) 175-191.

[338] W. Sun, J. Yuan and Y. Yuan, Conic trust-region method for linearly constrained optimization, *Journal of Computational Mathematics* 21 (2003) 295-304.

[339] W. Sun, C. Xu and D. Zhu, *Optimization Methods*, Higher Education Press, Beijing, (2004). (in Chinese).

[340] R.A. Tapia, Diagonalized multiplier methods and quasi-Newton methods for constrained optimization, *J. Optimization Theory and Applications* 22 (1977) 135-194.

[341] Ph.L. Toint, Towards an efficient sparsity exploiting Newton method for minimization, in: I.S. Duff, ed., *Sparse Matrices and Their Uses* (Academic Press, London, 1981) 57-88.

[342] Ph.L. Toint, Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space, *IMA J. Numer. Anal.* 8 (1988) 231-252.

[343] Ph.L. Toint, An assessment of nonmonotone linesearch technique for unconstrained optimization, *SIAM J. Scientific Computing* 17 (1996) 725-739.

[344] L. Vandenberghe and S. Boyd, Semidefinite programming, *SIAM Review* 38 (1996) 49-95.

[345] A. Vardi, A trust region algorithm for equality constrained minimization: convergence properties and implementation, *SIAM J. Numer. Anal.* 22 (1985) 575-591.

[346] H. Wang and Y. Yuan, An second order convergent method for one-dimensional optimization, *Chinese Journal of Operations Research* 11 (1992) 1-10.

[347] G.R. Walsch, *An Introduction to Linear Programming*, (John Wiley and Sons, New York, 1985).

[348] G.A. Watson, Methods for best approximation and regression, in: A. Iserles and M.J.D. Powell eds., *The State of the Art in Numerical Analysis*, (Clarendon Press, Oxford, 1987) 139-164.

[349] R.B. Wilson, A simplicial algorithm for concave programming, Ph.D. thesis, Graduate School of Business administration, Harvard University, (1963).

[350] P. Wolfe, Methods of recent advances in mathematical programming, in: R.L. Graves and P. Wolfe, eds., *Recent Advances in Mathematical Programming* (McGraw-Hill, New York, 1963) 67-86.

[351] P. Wolfe, Another variable metric method, working paper, (1968).

[352] P. Wolfe, Convergence conditions for ascent methods, *SIAM Review* 11 (1969) 226-235.

[353] P. Wolfe, Convergence conditions for ascent methods, (II): some corrections, *SIAM Review* 13 (1971) 185-188.

[354] P. Wolfe, A method of conjugate subgradients for minimizing nondifferentiable functions, *Math. Prog. Study* 3 (1975) 145-173.

[355] H. Wolkowitz, R. Saigal, and L. Vandenberghe, edc., *Handbook of Semidefinite Programming: Theory, Algorithms and Applications*, Kluwer International Series in Operational Research and Management Science, Kluwer, Boston, (2000).

[356] R. S. Womersley, Local properties of algorithms for minimizing nonsmooth composite functions, *Mathematical Programming* 32 (1985) 69-89.

[357] S.J. Wright, Local properties of inexact methods for minimizing nonsmooth composite functions, *Mathematical Programming* 37 (1987) 232-252.

[358] S.J. Wright, *Primal-Dual Interior-Point Methods*, SIAM Publications, Philadelphia, PA, (1997).

[359] S. Xi, *Nonlinear Optimization Methods*, Higher Education Press, Beijing, (1992). (in Chinese).

[360] S. Xi and F. Zhao, *Computational Methods for Optimization*, Shanghai Sci & Tech Press, Shanghai, (1983). (in Chinese).

[361] Y. Xiao and F.J. Zhou, Nonmonotone trust region methods with curvilinear path in unconstrained optimization, *Computing* 48 (1992) 303-317.

[362] C.X. Xu and J. Zhang, A survey of quasi-Newton equations and quasi-Newton methods for optimization, *Annals of Operations Research* 103 (2001) 213-234.

[363] Y. Ye and M.J. Todd, Containing and shrinking ellipsoids in the path-following algorithm, *Math. Porg.* 47 (1990) 1-9.

[364] Y. Ye and E. Tse, An extension of Karmarkar's algorithm to convex quadratic programming, *Math. Prog.* 44 (1989) 157-179.

[365] T.J. Ypma, Local convergence of inexact Newton methods, *SIAM J. Numer. Anal.* 21 (1984) 583-590.

[366] Y. Yuan, On the least Q-order of convergence of variable metric algorithms, *IMA J. Numerical Analysis* 4 (1984) 233-239. (1984a).

[367] Y. Yuan, An example of only linearly convergence of trust region algorithms for nonsmooth optimization, *IMA J. Numerical Analysis* 4(1984) 327-335. (1984b).

[368] Y. Yuan, Conditions for convergence of trust region algorithms for nonsmooth optimization, *Mathematical Programming* 31 (1985) 220-228. (1985a).

[369] Y. Yuan, On the superlinear convergence of a trust region algorithm for nonsmooth optimization, *Mathematical Programming* 31 (1985) 269-285. (1985b).

[370] Y. Yuan, An only 2-step Q-superlinear convergence example for some algorithms that use reduced Hessian approximation, *Mathematical Programming* 32 (1985) 224-231. (1985c).

[371] Y. Yuan, On a subproblem of trust region algorithms for constrained optimization, *Mathematical Programming* 47 (1990) 53-63.

[372] Y. Yuan, A modified BFGS algorithm for unconstrained optimization, *IMA Journal of Numerical Analysis* 11 (1991) 325-332.

[373] Y. Yuan, A dual algorithm for minimizing a quadratic function with two quadratic constraints, *Journal of Computational Mathematics* 9 (1991) 348-359.

[374] Y.Yuan, On self-dual update formulae in the Broyden family, *Optimization Methods and Software* 1 (1992) 117-127.

[375] Y. Yuan, *NumericaL Methods for Nonlinear Programming*, Shanghai Sci. & Tech Press, Shanghai, (1993).

[376] Y. Yuan and R. Byrd, Non-quasi-Newton updates for unconstrained optimization, *J. Comp. Math.* 13 (1995) 95-107.

[377] Y. Yuan, On the truncated conjugate gradient method, *Mathematical Programming* 87 (2000) 561-573.

[378] W.I. Zangwill. Non-linear programming via penalty functions, *Management Sci.* 13 (1967) 344-358.

[379] J.Z. Zhang, N.Y. Deng, and L.H. Chen, A new quasi-Newton equation and related methods for unconstrained optimization, *Journal of Optimization Theory and applications* 102 (1999) 147-167.

[380] J.Z. Zhang and C.X. Xu, A class of indefinite dogleg path methods for unconstrained minimization, *SIAM J. on Optimization* 9 (1994) 646-667.

[381] Y. Zhang, Computing a Celis-Dennis-Tapia trust region step for equality constrained optimization, *Mathematical Programming* 55 (1992) 109-124.

[382] H.C. Zhou and W. Sun, Optimality and duality without a constraint qualification for minimax programming, *Bulletin of the Australian Mathematical Society* 67 (2003) 121-130.

[383] H.C. Zhou and W. Sun, Nonmonotone descent algorithm for nonsmooth unconstrained optimization problems, *International Journal of Pure and Applied Mathematics* 9 (2003) 153-163.

[384] M. Zhu, Y. Xue, and F. Zhang, A quasi-Newton type trust region method based on the conic model, *Numerical Mathematics, A Journal of Chinese Universities* No.1 (1995) 36-47.

[385] G. Zoutendijk, Nonlinear programming, computational methods, in: J. Abadie ed. *Integer and Nonlinear Programming*, North-Holland, Amsterdam, (1970), 37-86.

[386] J. Zowe, Nondifferentiable optimization — a motivation and a short introduction into the subgradient and the bundle concept, in: K. Schittkowski, ed., *Computational Mathematical Programming* (Springer-Verlag, Berlin, 1985) 321-356.

Index