AN INTRODUCTION TO OPTIMIZATION TECHNIQUES

Suchin Arunsawatwong

An Introduction to Optimization Techniques

Suchin Arunsawatwong

Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University
Email: suchin.a@chula.ac.th

Contents

Preface **vi**

1 Introduction **1**

2 Mathematical Review **4**

2.1 Methods of Proof 4

2.1.1 Conditional Statement 4

2.1.2 Biconditional Statement 5

2.1.3 Mathematical induction 5

2.2 Vectors in $\mathbb{R}^n$ 6

2.2.1 Norms 6

2.2.2 Angle between two vectors 7

2.2.3 Upper bound of $\left|\mathbf{x}^T\mathbf{y}\right|$ 7

2.2.4 Linear independence 7

2.3 Matrix 7

2.3.1 Rank 8

2.3.2 Eigenvalues and eigenvectors 8

2.3.3 Positive and negative definite matrices 8

2.4 System of linear equations 9

2.4.1 Gaussian elimination & LU Factorization 10

2.5 Calculus of Several Variables 12

2.6 Taylor Series . . . 12
2.6.1 One-dimensional case . . . 13
2.6.2 Multi-dimensional case . . . 13

3 Solution of Nonlinear Equations 15
3.1 Scalar Case . . . 16
3.1.1 Bisection method . . . 16
3.1.2 Newton-Raphson Method . . . 18
3.1.3 Secant Method . . . 20
3.1.4 Convergence Properties . . . 21
3.2 Multivariable Case . . . 24
3.3 Exercises . . . 27

4 Optimality Conditions for Unconstrained Optimization 28
4.1 Local and Global minima . . . 29
4.1.1 Convex & Concave Functions . . . 30
4.1.2 Directional derivatives . . . 31
4.2 Optimality Conditions . . . 31
4.3 Exercises . . . 34

5 Numerical Methods for Unconstrained Optimization 35
5.1 One-Dimensional Optimization . . . 36
5.1.1 Quadratic Interpolation Method . . . 37
5.1.2 Cubic Interpolation Method . . . 39
5.1.3 Golden Section Method . . . 40
5.2 Methods for Unconstrained Optimization . . . 41
5.2.1 Steepest Descent Method . . . 43
5.2.2 Newton's Method . . . 45
5.2.3 Conjugate Directions . . . 51

5.2.4 Quasi-Newton (Variable Metric) Methods . . . 53
5.2.5 Conjugate Gradient Methods . . . 61
5.3 Exercises . . . 70

6 Optimality Conditions for Constrained Optimization 74
6.1 Equality-Constrained Problems . . . 76
6.1.1 Tangent plane . . . 77
6.2 Inequality-Constrained Problems . . . 81
6.2.1 First-Order Necessary Condition . . . 83
6.2.2 Second-Order Conditions . . . 85
6.3 Convex Programs . . . 86
6.4 Exercises . . . 87

7 Representation of Linear Constraints 88
7.1 Equality Constraints . . . 90
7.2 Inequality Constraints . . . 91
7.3 Mixed Equality-Inequality Constraints . . . 91
7.4 Exercises . . . 95

8 Linear Programming 96
8.1 Standard Form . . . 98
8.2 Extreme Points & Basic Feasible Solutions . . . 99
8.3 Simplex Method . . . 103
8.4 Artificial Variables . . . 105
8.4.1 Two-phase method . . . 106
8.4.2 Big-M Method . . . 110
8.5 Exercises . . . 113

9 Quadratic Programming 115

9.1 Problems with Equality Constraints 116
9.2 Active Set Method 117
9.3 Exercises 124

References **125**

Index **128**

Preface

This book contains the materials that I have used in teaching the course 2102-505 *Introduction to Optimization Techniques* at the Department of Electrical Engineering, Faculty of Engineeing, Chulalongkorn University since 1998.

The course considers unconstrained and constrained optimization problems that are defined in an n-dimensional Euclidean space where the associated functions are either linear or nonlinear. The main objective of this course is to provide basic knowledge of optimization theory and basic numerical methods so that students will be able to solve optimization problems of medium size by using computers.

In this course, students are assumed to have backgrounds in elementary linear algebra and calculus of several variables, which are usually taught in the second year undergraduate level. Chapter 2 gives a recap of the mathematical backgrounds that will be required in subsequent chapters. Students who are acquainted with the backgrounds may skip this chapter and go straight to Chapter 3, whereas those who are not acquainted with the backgrounds may read Chapter 2 and, if needed, find more details in the references mentioned in each topic.

Chapter 3 considers the solution of nonlinear algebraic equations by iterative algorithms, which begin with an initial guess and then generate a sequence of points (called iterates) until the algorithms hopefully terminate at a soluition. Also, the concept of convergence of an iterative method is explained where attention is focused on linear, superlinear and quadratic rates of convergence, which are usually found in practical algorithms.

Chapter 4 considers the first order and the second order conditions for optimality for unconstrained optimization problems. These conditions provide basic tools for solving the problems analytically and numerically.

Chapter 5 explains numerical methods for solving unconstrained optimization problems. First we consider how to solve one-dimensional problems, where focus is given to quadratic interpolation, cubic interpolation and golden section methods. Then we consider how to solve multi-dimensional problems, where we focus only descent methods (i.e., search methods that

require derivative infofrmation). In this regard, steepest descent, Newton, quasi-Newton and conjugate gradient methods are explained. Moreover, implementation of inexact line search using Armijo's rule together with Wolfe's condition is discussed.

Chapter 6 considers the first order and the second order conditions for optimality for (general) constrained optimization problems, in which Lagrange's and Karush–Kuhn–Tucker's theorems are covered. The conditions provide basic tools for solving optimization problems with linear and nonlinear constraints analytically and numerically.

Chapter 7 examines a simple and useful way of representing linear constraints in which the constraints are expressed in a form that makes it easy for a search algorithm to move from one feasible point to another.

Chapter 8 examines linear programming problems, in which the objective function and the constraint functions are linear. Attention is focused on Simplex method (due to George Dantzig), which is used to solve linear programs considered in this course. Since the Simplex method needs to start from a basic feasible solution, a systematic approach to find a basic feasible solution for linear programs is to use artificial variables. In this connection, a two phase method and big M method are explained at the end of the chapter.

Chapter 9 examines convex quadratic programing problems, in which the objective function is quadratic and the constraints are linear. The active set method is considere and explained in details.

Finally, I would like to give special thanks to my former research student Tadchanon Chuman for being a teaching assistance in last few years and assisting me to prepare the book.

Suchin Arunsawatwong
1st February 2023

Chapter 1

Introduction

This chapter give a brief introduction of optimization and a scope of optimization problems that are treated in this course.

What is optimization? To optimize means to use something in the best possible way. In mathematics, *optimization* is a procedure of determining the best possible value (either minimum or maximum) of a function, which is often called an objective function or a cost function.

Why do we study optimization? Optimization problems arise in various situations, especially when one needs to make a decision. Such problems are found in engineering, economics and finance where people want to have the best solution. Apart from this, nature optimizes. Examples of optimization are as follows.

- Engineers adjust parameters to optimize the performance of their designs.
- Airline companies schedule crews and aircraft to minimize cost.
- Investors seek to create portfolios that avoid excessive risks while achieving a high rate of return.
- Manufacturers aim for maximum efficiency in the design and operation of their production processes.
- Rays of light follow paths that minimize their travel time.
- Physical systems tend to a state of minimum energy.

In making use of optimization in decision making, we have to clearly define an appropriate objective, which is represented by a quantitative measure of the performance of the system under consideration. For example, the objective could be profit, time, potential energy, transmission loss in an electric power system, etc.

The objective depends on certain variables/parameters of the system. Our goal is to find variables/parameters that optimize the objective. Often, variables of the system have constraints/limitations in some way. For example, quantities such as electron density in a molecule and the interest rate on a loan cannot be negative. The process of identifying objective, variables and constraints for a given problem is known as mathematical modelling. Once the model has been formulated, an optimization algorithm can be used to find a solution, usually with the use of computer.

Similarly, engineering design is a decision making problem. It involves the following steps.

- Formulate a mathematical model of the system under study.
- Identify an objective function and choose design variables. (The objective function depends on design variables.)
- Employ optimization algorithms to find a solution to the problem.

The main objective of this course is to provide basic knowledge of optimization theory and numerical methods so that students can solve simple optimization problems found in practice. For examples of optimization problems arising in engineering applications, readers are referred to, for example, [3, 2, 35].

In this course, we consider 2 types of optimization problem in $\mathbb{R}^n$, where $\mathbb{R}^n$ denotes the n-dimensional Euclidean space.

(I) Unconstrained optimization:

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}), \qquad \text{where } f:\mathbb{R}^n\to\mathbb{R}.$$

(II) Constrained optimization:

$$\min_{\mathbf{x}\in S} f(\mathbf{x})$$

where $S\subset\mathbb{R}^n$ is the set of all points $\mathbf{x}\in\mathbb{R}^n$ satisfying

$$\begin{aligned} h_i(\mathbf{x}) &= 0, \quad i=1,2,\ldots,m_1 \qquad \text{where } h_i:\mathbb{R}^n\to\mathbb{R} \\ g_j(\mathbf{x}) &\le 0, \quad j=1,2,\ldots,m_2 \qquad \text{where } g_j:\mathbb{R}^n\to\mathbb{R}. \end{aligned}$$

In practice, there are a great many applications that can be cast as continuous optimization stated above. Some examples are as follows.

Example Rosenbrock's test function in 50 variables:

$$\min_{(x_1,x_2,\ldots,x_{50})} \quad f(\mathbf{x}) \triangleq \sum_{i=1}^{25}\left[100\big(x_{2i}-x_{2i-1}^2\big)^2+(1-x_{2i-1}^2)\right].$$

Example A constrained optimization problem:

$$\min_{(x_1,x_2,\ldots,x_n)} \quad \frac{1}{2}||Ax-b||^2$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $m < n$ and the variables $x_1, x_2, \ldots, x_n$ satisfy the following constraints

$$x_1 + x_2 + \ldots + x_n = 1 \quad \text{and} \quad x_i \geq 0 \; \forall i.$$

Chapter 2

Mathematical Review

This chapter provides mathematical backgrounds for the topics to be explained in subsequent chapters. Many of the materials in this chapter can be found in [8].

2.1 Methods of Proof

This section briefly explains procedures that will be used for proving mathematical statements found in this course. More details on methods of proof can be found in Appendices B and C of reference [27].

2.1.1 Conditional Statement

Let A a statement and let $\sim A$ denote the negation of A. When statement A implies statement B, A is the *hypothesis* and B the *conclusion* of the implication. We write $A \Rightarrow B$, which is read as one of the following.

- A implies B.
- if A, then B.
- A is sufficient for B.
- B is necessary for A.

In proving $A \Rightarrow B$, we may use by one of the following three methods.

- Direct method $[A \Rightarrow B]$.
  Assume that A is true. Then try to deduce consequences to show that B is true.

- Proof by contraposition [$A \Rightarrow B$ is equivalent to $\sim B \Rightarrow \sim A$].
  Assume that $\sim B$ is true. Then try to deduce consequences to show that $\sim A$ is true.
- Proof by contradiction [$A \Rightarrow B$ is equivalent to $\sim(A$ and $\sim B)$].
  Assume that A holds. Assume by contradiction that $\sim B$ is true. Then try to deduce consequences to show a contradiction.

Example Prove that if $n+1$ pigeons are placed in n holes, then some hole contains at least 2 pigeons.

Solution: Let A be the statement "$n+1$ pigeons are placed in n holes". And let B denote the statement "some hole contains at least 2 pigeons".

Assume, by contradiction, that $\sim B$ holds, that is, every hole contains at most one pigeon. Since there are n holes (from statement A), there must be at most n pigeons (from statement $\sim B$). Hence, this shows a contradiction. ■

Exercise By using contradiction, prove that if $r = \sqrt{2}$, then r is an irrational number.

2.1.2 Biconditional Statement

When statements A and B are *logically equivalent*, we write $A \Leftrightarrow B$, which is read as one of the following.

- A if and only if B.
- A is necessary and sufficient for B.

By using the fact that $A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A)$, it readily follows that establishing the statement $A \Leftrightarrow B$ comprises 2 steps:

- Step 1: show $A \Rightarrow B$,
- Step 2: show $B \Rightarrow A$.

2.1.3 Mathematical induction

Apart from conditional and biconditional statements, we occasionally have to prove a statement of the form

$$A(n) \text{ is true for } n \geq N_0 \tag{2.1}$$

where $A(n)$ denote a statement depending on an integer n.

According to the principle of mathematical induction, proving statement (2.1) comprises 2 steps.

- Establish that $A(n)$ is true for $n = N_0$.
- Establish that if $A(k)$ is true, then $A(k+1)$ is true.

Exercise Use the mathematical induction to prove that

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} \quad \text{and} \quad \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}.$$

2.2 Vectors in $\mathbb{R}^n$

2.2.1 Norms

For a vector $\mathbf{x} \in \mathbb{R}^n$, a *norm* of $\mathbf{x}$ (denoted by $\| \mathbf{x} \|$) is a real-valued function on $\mathbb{R}^n$ which has the following properties.

$$\| \mathbf{x} \| \geq \mathbf{0} \qquad \text{and} \qquad \| \mathbf{x} \| = \mathbf{0} \Leftrightarrow \mathbf{x} = \mathbf{0}.$$

$$\| \alpha \mathbf{x} \| = |\alpha| \| \mathbf{x} \| \qquad \text{for any scalar } \alpha.$$

$$\| \mathbf{x} + \mathbf{y} \| \leq \| \mathbf{x} \| + \| \mathbf{y} \| \qquad \text{for all } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n. \tag{2.2}$$

Property (2.2) is called *triangle inequality*.

For $p \in \mathbb{R}$, the p-norm of a vector $\mathbf{x}$ (denoted by $\| \mathbf{x} \|_p$) is defined as

$$\| \mathbf{x} \|_p = \left\{ \sum_{i=1}^{n} | x_i |^p \right\}^{1/p}.$$

The three most common vector norms are as follows:

$$\textbf{1-norm} \qquad \| \mathbf{x} \|_1 = \sum_{i=1}^{n} |x_i| \tag{2.3}$$

$$\textbf{Euclidean or 2-norm} \qquad \| \mathbf{x} \|_2 = \left\{ \sum_{i=1}^{n} | x_i |^2 \right\}^{1/2} \tag{2.4}$$

$$\infty\textbf{-norm} \qquad \| \mathbf{x} \|_\infty = \max_{1 \leq i \leq n} |x_i| \tag{2.5}$$

2.2.2 Angle between two vectors

The angle θ between two nonzero vectors $\mathbf{x}$, $\mathbf{y}$ is given by

$$\cos\theta = \frac{\mathbf{x}^T\mathbf{y}}{\|\,\mathbf{x}\,\|_2\|\,\mathbf{y}\,\|_2}.$$

The vectors $\mathbf{x}$ and $\mathbf{y}$ are said to be *orthogonal* if

$$\mathbf{x}^T\mathbf{y} = \sum_{i=1}^{n} x_i y_i = 0.$$

2.2.3 Upper bound of $\left|\,\mathbf{x}^T\mathbf{y}\,\right|$

For vectors $\mathbf{x}$, $\mathbf{y} \in \mathbb{R}^n$, an upper bound of $\left|\mathbf{x}^T\mathbf{y}\right|$ can be given as follows:

$$\left|\,\mathbf{x}^T\mathbf{y}\,\right| \le \|\,\mathbf{x}\,\|_n\,\|\,\mathbf{y}\,\|_m \quad \text{where } \frac{1}{n}+\frac{1}{m}=1. \tag{2.6}$$

Inequality (2.6) is known as the Hölder inequality. For $n = m = 2$, inequality (2.6) is known as the *Cauchy-Schwarz inequality*, which is

$$\left|\,\mathbf{x}^T\mathbf{y}\,\right| \le \|\,\mathbf{x}\,\|_2\,\|\,\mathbf{y}\,\|_2. \tag{2.7}$$

2.2.4 Linear independence

For a set of vectors, the definitions of linear independence and dependence are given as follows.

Definition 2.2.1. *Let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$ be vectors in $\mathbb{R}^n$ where $k \le n$. A set of vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$ is said to be* linearly independent *if*

$$\alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2 + \ldots + \alpha_k\mathbf{x}_k = \mathbf{0} \implies \alpha_1 = \cdots = \alpha_k = 0.$$

Otherwise, it is said to be linearly dependent. ■

It should be noted that the equation $\alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2 + \ldots + \alpha_k\mathbf{x}_k = \mathbf{0}$ leads to a system of homogeneous equations with the unknowns $\alpha_1, \ldots, \alpha_k$.

2.3 Matrix

For the details of the materials on matrices , students are referred to well-known books on matrices such as [19, 20, 28, 38].

2.3.1 Rank

The rank of a matrix is defined as follows.

Definition 2.3.1. *For a matrix $A \in \mathbb{R}^{m\times n}$, the rank of A – denoted by* $\text{rank}(A)$ *– is the maximum number of linearly independent columns (or rows) in A.* ■

Given $A \in \mathbb{R}^{m\times n}$, the rank of A is obtained by the following procedure.

- Perform elementary row operations to carry A to an echelon matrix A_1.
- The rank of A is equal to the number of the non-zero rows of A_1.

2.3.2 Eigenvalues and eigenvectors

The definitions of the eigenvalues and the eigenvectors of a square matrix are given below.

Definition 2.3.2. *For a matrix $A \in \mathbb{R}^{n\times n}$, if the scalar λ and a vector $\mathbf{x}$ satisfy*

$$A\mathbf{x} = \lambda\mathbf{x}$$

then λ is an eigenvalue *and $\mathbf{x}$ is an* eigenvector *of A. Hence, the eigenvalues of A are the solutions of*

$$\det[\lambda I - A] = 0,$$

where I is the identity matrix. ■

For a matrix $A \in \mathbb{R}^{n\times n}$, it can be verified that

$$\text{trace}(A) = \sum_{i=1}^{n} a_{ii} = \sum_{i=1}^{n} \lambda_i(A),$$

$$\det(A) = \prod_{i=1}^{n} \lambda_i(A),$$

where $\lambda_i(A)$ is an eigenvalue of A.

2.3.3 Positive and negative definite matrices

In studying optmization, we often have to deal with positive (or negative) definite matrices.

Definition 2.3.3. *A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is said to be* positive definite *(or pdf) if*

$$\mathbf{x}^T A\mathbf{x} > 0 \quad \text{for all } \mathbf{x} \neq \mathbf{0}$$

A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is said to be positive semi-definite *(or psdf) if*

$$\mathbf{x}^T A\mathbf{x} \geq 0 \quad \text{for all } \mathbf{x}.$$ ■

Definition 2.3.4. *A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is said to be* negative definite *(or ndf) if*

$$\mathbf{x}^T A\mathbf{x} < 0 \quad \text{for all } \mathbf{x} \neq \mathbf{0}.$$

A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is said to be negative semi-definite *(or nsdf) if*

$$\mathbf{x}^T A\mathbf{x} \leq 0 \quad \text{for all } \mathbf{x}.$$ ■

Definition 2.3.5. *A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is said to be* indefinite *if $\mathbf{x}^T A\mathbf{x}$ can take positive and negative values.* ■

Proposition 2.3.6. *If $A \in \mathbb{R}^{n\times n}$ is symmetric, then all the eigenvalues of A are real.* ■

Sometimes, it is not convenient to determine whether a square matrix is pdf (or psdf) by using the above definition. The following propositions provide a necessary and sufficient condition for a matrix to be pdf (or psdf), and hence a practical method for determining whether the matrix is pdf (or psdf).

Proposition 2.3.7. *A matrix A is positive definite if, and only if, all its eigenvalues of A are real and strictly positive.* ■

Proposition 2.3.8. *A matrix A is positive semidefinite if, and only if, all its eigenvalues of A are real and nonnegative.* ■

2.4 System of linear equations

For useful references on the numerical solution of a system of linear equations, students are referred to well-known books such as [19, 7].

Let us consider a system of linear equations

$$A\mathbf{x} = \mathbf{b} \tag{2.8}$$

where $\mathbf{x}$ are the unknown vector, A is a square matrix and $\mathbf{b}$ is a known vector.

Theorem 2.4.1. *For equation (2.8), only one of the following three possibilities must hold:*

1. $\text{rank}[A \mid \mathbf{b}] > \text{rank}(A)$, *and no solution exists to* $A\mathbf{x} = \mathbf{b}$.
2. $\text{rank}[A \mid \mathbf{b}] = \text{rank}(A) < \dim(\mathbf{x})$, *and the system* $A\mathbf{x} = \mathbf{b}$ *has infinitely many solutions.*
3. $\text{rank}[A \mid \mathbf{b}] = \text{rank}(A) = \dim(\mathbf{x})$, *and the system* $A\mathbf{x} = \mathbf{b}$ *has exactly one solution.* ■

It may noted that Theorem 2.4.1 also holds when A is a non-square matrix.

2.4.1 Gaussian elimination & LU Factorization

A solution of the linear system (2.8) is often obtained by Gaussian elimination or LU factorization method. In many cases, both methods are equivalent in terms of the number of flops. However, when we have to solve the system (2.8) consecutively for a fixed A and different $\mathbf{b}$'s, LU factorization method can be more efficient.

Consider the system

$$\begin{aligned} 2.0x_1 - 4.0x_2 - 1.0x_3 &= 2.0 \\ 0.4x_1 + 2.2x_2 + 1.8x_3 &= 2.4 \\ 0.8x_1 - 0.1x_2 - 1.4x_3 &= 5.8 \end{aligned} \tag{2.9}$$

Performing Gaussian elimination of (2.9) yields an equivalent triangular system

$$\begin{aligned} 2.0x_1 - 4.0x_2 - 1.0x_3 &= 2.0 \\ 3.0x_2 + 2.0x_3 &= 2.0 \\ 2.0x_3 &= 4.0 \end{aligned} \tag{2.10}$$

Thus, $\mathbf{x}$ can readily be determined by performing *backward substitutions* on the system (2.11).

In connection with the above, consider a matrix $A \in \mathbb{R}^{n\times n}$ and let A be factorized such that

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & & & \mathbf{O} \\ l_{21} & 1 & & \\ \vdots & & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & & u_{2n} \\ & & \ddots & \vdots \\ \mathbf{O} & & & u_{nn} \end{pmatrix}$$

Then equation (2.8) becomes

$$L\,(U\mathbf{x}) = \mathbf{b} \tag{2.11}$$

By letting $U\mathbf{x} = \mathbf{z}$, it follows that

$$\begin{array}{llll} \text{solve} & L\mathbf{z} = \mathbf{b} & \text{for } \mathbf{z} & \text{(forward substitution)} \\ \text{solve} & U\mathbf{x} = \mathbf{z} & \text{for } \mathbf{x} & \text{(backward substitution)} \end{array}$$

The cost of solving $A\mathbf{x} = \mathbf{b}$ by LU factorization is as follows. (See, *e.g.*, [28] for the details.)

cost (as $n \to \infty$)	multiplications	additions
computing the factors L, U	$n^3/3$	$n^3/3$
solving $A\mathbf{x} = \mathbf{b}$ by substitution	n^2	n^2

Now consider another system of linear equations

$$\begin{aligned} 0.0x_1 - 4.0x_2 - 1.0x_3 &= 2.0 \\ 0.4x_1 + 2.2x_2 + 1.8x_3 &= 2.4 \\ 0.8x_1 - 0.1x_2 - 1.4x_3 &= 5.8 \end{aligned}$$

Can we still solve the above system using Gaussian elimination? The answer is *YES*. Interchange the first equation with the third one. Then proceed as before and we can obtain the solution. The process of permuting the orders of the rows of the matrix is known as **pivoting**.

Usually Gaussian elimination method is implemented with **partial pivoting** scheme, in which the method selects the pivot row to be the one with the maximum pivot entry in absolute value from those in the leading column of the reduced submatrix.

It is known (see, for example, [19, 7]) that Gaussian elimination process with partial pivoting is equivalent to

$$PA = LU \tag{2.12}$$

where P is a permutation matrix (and also orthogonal) that records the interchange of equations used during the elimination. In fact, the columns of P are a permutation of the columns of the identity matrix.

With the factorization in (2.12), the solution of 2.8 is obtained by the following three-step procedure.

- Form $\tilde{\mathbf{b}} = P\mathbf{b}$ by permuting the elements of $\mathbf{b}$.
- Solve $L\mathbf{z} = \tilde{\mathbf{b}}$ for $\mathbf{z}$ by performing forward substitution.
- Solve $U\mathbf{x} = \mathbf{z}$ for $\mathbf{x}$ by performing backward substitution.

2.5 Calculus of Several Variables

An important tool for deriving optimality conditions (see Chapter 4) is calculus of several variables. A standard reference on this topic is, for example, Kaplan's book [22].

The definition of a continuously differentiable function is given as follows.

Definition 2.5.1. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be* continuously differentiable at the point $\mathbf{x} \in \mathbb{R}^n$ *if the partial derivatives $\frac{\partial f(\mathbf{x})}{\partial x_i}$ exists and be continuous for all $1 \leq i \leq n$.* ■

In connection with the above, we say $f \in C^1$ if f is continuously differentiable at every $\mathbf{x} \in \mathbb{R}^n$. For $f \in C^1$, we define the *gradient* of f as follows.

$$\nabla f(\mathbf{x}) \triangleq \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \cdots, \frac{\partial f(\mathbf{x})}{\partial x_n}\right]^T.$$

In a similar fashion to the above definition, a function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be *twice continuously differentiable at* $\mathbf{x} \in \mathbb{R}^n$ if $\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j}$ exists and continuous for all $1 \leq i, j \leq n$. And we say $f \in C^2$ if f is twice continuously differentiable at every $\mathbf{x} \in \mathbb{R}^n$.

For $f \in C^2$, the *Hessian (matrix)* of f is defined as

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

Recall that for functions with continuous second derivatives,

$$\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} = \frac{\partial^2 f(\mathbf{x})}{\partial x_j \partial x_i}$$

Thus, the Hessian is always symmetric as long as $f \in C^2$.

2.6 Taylor Series

Taylor series and the mean value theorem are useful tools in mathematical analysis. As will be seen later, they will be used to establish many important results.

2.6.1 One-dimensional case

For $f : \mathbb{R} \to \mathbb{R}$, the mean value theorem is stated as follows.

Theorem 2.6.1. *If f is continuous on $[a, b]$ and differentiable on (a, b), then there is a number $\lambda \in (a, b)$ such that*

$$f(b) = f(a) + f'(\lambda)\{b - a\}.$$

■

By using Theorem 2.6.1, the Taylor's series of f is given by the following theorem.

Theorem 2.6.2. [Taylor] *Assume that f has $n + 1$ continuous derivatives. It follows that*

$$f(x_0 + s) = f(x_0) + sf'(x_0) + \cdots + \frac{s^n}{n!} f^{(n)}(x_0) + \frac{s^{n+1}}{(n+1)!} f^{(n+1)}(\xi)$$

for some $\xi \in (x_0, x_0 + s)$. ■

Equivalently, the variable $\xi \in (x_0, x_0 + s)$ in the above thoerem can also be expressed as

$$\xi = x_0 + \lambda s, \quad \exists \lambda \in (0, 1). \tag{2.13}$$

Note that the expression (2.13) can be easily extended to the multivariable case and will be used next.

2.6.2 Multi-dimensional case

For $f : \mathbb{R}^n \to \mathbb{R}$, the mean value thoerem is stated below.

Theorem 2.6.3. *Let f be continuously differentiable in $\mathbb{R}^n$. Then it follows that*

$$f(\mathbf{x}_0 + \mathbf{s}) = f(\mathbf{x}_0) + \mathbf{s}^T \nabla f(\mathbf{x}_0 + \lambda \mathbf{s}), \quad \lambda \in (0, 1).$$

Alternatively, the above can be expressed as

$$f(\mathbf{x}_0 + \mathbf{s}) = f(\mathbf{x}_0) + \int_0^1 \mathbf{s}^T \nabla f(\mathbf{x}_0 + t\mathbf{s})\, dt.$$

■

For $f \in C^2$, it follows from theorem 2.6.3 that

$$f(\mathbf{x}_0 + \mathbf{s}) = f(\mathbf{x}_0) \ + \mathbf{s}^T \nabla f(\mathbf{x}_0) + \frac{1}{2} \mathbf{s}^T \nabla^2 f(\mathbf{x}_0 + \lambda \mathbf{s})\, \mathbf{s}, \quad \lambda \in (0, 1).$$

Notice that $\mathbf{x}_0 + \lambda\mathbf{s}$ for $\lambda \in (0, 1)$ is a point lying on the straight line connection between the points $\mathbf{x}_0$ and $\mathbf{x}_0 + \mathbf{s}$.

A Taylor's series for f is given by

$$f(\mathbf{x}_0 + \mathbf{s}) = f(\mathbf{x}_0) + \mathbf{s}^T \nabla f(\mathbf{x}_0) + \frac{1}{2}\,\mathbf{s}^T \nabla^2 f(\mathbf{x}_0)\mathbf{s} + \cdots . \tag{2.14}$$

Using summations rather than vector-matrix notation, the expression in (2.14) can be rewritten as

$$f(\mathbf{x}_0 + \mathbf{s}) = f(\mathbf{x}_0) + \sum_{i=1}^{n} s_i \frac{\partial f(\mathbf{x}_0)}{\partial x_i} + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} s_i s_j \frac{\partial^2 f(\mathbf{x}_0)}{\partial x_i \partial x_j} + \cdots .$$

The higher-order terms of the Taylor series can also be written down, but the notation will be more complex and they will not be required in this course.

Chapter 3

Solution of Nonlinear Equations

This chapter describes numerical methods for solving nonlinear algebraic equations. As will become clear in Chapter 4, the problem of solving nonlinear equations is closely related to the optimization problem.

3.1 Scalar Case

Consider a nonlinear algebraic equation

$$f(x) = 0, \tag{3.1}$$

where $f : \mathbb{R} \to \mathbb{R}$ is continuous. For example,

$$f(x) = x^3 + 5x - 1 \qquad \text{or} \quad f(x) = \mathrm{e}^x - \mathrm{e}^{-x}.$$

In practice, an iterative method is often employed to compute a numerical solution of such an equation. In this section, attention will be focused mainly on bisection and Newton-Raphson methods.

3.1.1 Bisection method

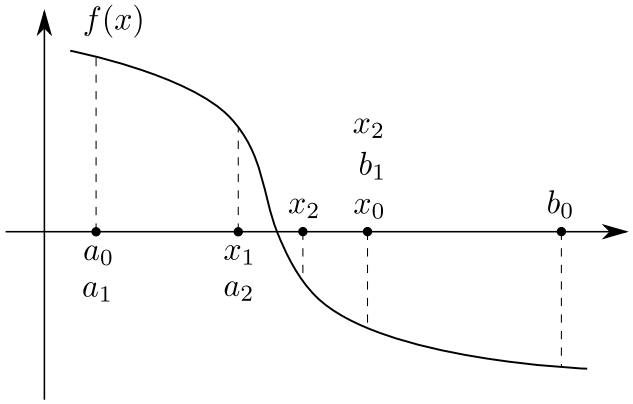

Figure 3.1: Illustration of the bisection method

The bisection method exploits the fact that a continuous function cannot change the sign of its value immediately.

Proposition 3.1.1. *Let f be a continuous function. For a given interval $[a, b]$, if $f(a)$ and $f(b)$ have different signs, then there exists a root of $f(x) = 0$ in $[a, b]$.* ■

Consider the above figure. With the initial interval $[a_0, b_0]$, compute the mid-point x_0 and $f(x_0)$. Because $f(x_0)$ and $f(b_0)$ are the same, the solution is in $[a_1, b_1]$ where $a_1 = a_0$ and $b_1 = x_0$.

Next compute the midpoint $x_1 = (a_1 + b_1)/2$ and $f(x_1)$. Because $f(a_1)$ and $f(x_1)$ are the same, the solution is in $[a_2, b_2]$ where $a_2 = x_1$ and $b_2 = b_1$.

In this example, if we keep doing like this, then $[a_k, b_k]$ will eventually converge to the solution as $k \to \infty$. And the obtained midpoint x_k may be considered as an approximate soliution.

From the above, we can establish the following algorithm.

Algorithm Given a_0, b_0 such that $f(a_0)f(b_0) < 0$,
    $i = 0$
    repeat
        $x_i = (a_i + b_i)/2$
        if $f(x_i)f(a_i) < 0$ then
            $a_{i+1} = a_i$; $b_{i+1} = x_i$;
        else
            $a_{i+1} = x_i$; $b_{i+1} = b_i$;
        endif
        $i = i + 1$;
    until $[a, b]$ is small enough.

Example Compute $\sqrt{2}$ by the bisection method.
Solution Let

$$f(x) = x^2 - 2 = 0.$$

Then it follows that the positive root x^* of $f(x) = 0$ is $\sqrt{2}$.
Since $\sqrt{1} = 1$ and $\sqrt{4} = 2$, we have

$$1 < \sqrt{2} < 2.$$

So we choose $a_0 = 1$ and $b_0 = 2$. Then we have

$$f(a_0) = -1, \quad f(b_0) = 2 \qquad (f(a)f(b) < 0).$$

For $k = 0, 1, 2, 3$, we have the following computational results.

$k=0$	$x_0 = \dfrac{a+b}{2} = 1.5$ $a_1 = 1$ $f(a_1) = -1,$	$f(x_0) = 0.25$ $b_1 = 1.5$ $f(b_1) = 0.25$
$k=1$	$x_1 = 1.25$ $a_2 = 1.25$ $f(a_2) = -0.4375$	$f(x_1) = -0.4375$ $b_2 = 1.5$ $f(b_2) = 0.25$
$k=2$	$x_2 = 1.375$ $a_3 = 1.375$ $f(a_3) = -0.109375$	$f(x_2) = -0.109375$ $b_3 = 1.5$ $f(b_3) = 0.25$
$k=3$	$x_3 = 1.4375$ $a_4 = 1.375$ $f(a_4) = -0.109375$	$f(x_3) = 0.06640625$ $b_4 = 1.4375$ $f(b_4) = 0.06640625$

We do the iteration further until $|a_k - b_k|$ is small enough. Ans.

3.1.2 Newton-Raphson Method

Suppose that x^* is a root of $f(x) = 0$. Consider the Taylor series of f at the point x_k. Then we have

$$f(x_k + s) \approx f(x_k) + sf'(x_k).$$

If $x_k + s$ is an good estimate of x^* so that

$$f(x^*) \approx f(x_k) + sf'(x_k) = 0$$

and if $f'(x_k) \neq 0$, then one can obtain

$$s = -f(x_k)/f'(x_k).$$

Hence, the new estimate of the solution is

$$x_{k+1} = x_k - f(x_k)/f'(x_k). \tag{3.2}$$

Formula (3.2) is known as **Newton-Raphson** method.

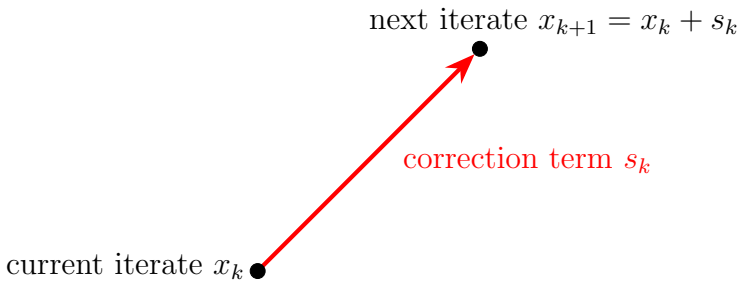

Figure 3.2: Newton–Raphson iteration moves from iterate x_k to x_{k+1}.

Example Compute $\sqrt{2}$ by Newton-Raphson method:

Solution Let

$$f(x) \triangleq x^2 - 2 = 0.$$

Then $x^* = \sqrt{2}$ is the positive root of $f(x) = 0$. Since

$$f'(x) = 2x,$$

Newton–Raphson formula becomes

$$x_{k+1} = x_k - \frac{x_k^2 - 2}{2\,x_k}. \tag{3.3}$$

Rearranging (3.3) yields

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{2}{x_k}\right). \tag{3.4}$$

Although the formulae (3.3) and (3.4) are equivalent, the latter one is preferred from the computational point of view. (Why?)

With the initial point $x_0 = 1$, we have the computational results as follows:

$k = 0$	$x_1 = 1.50000000$	$f(x_1) = 0.25$
$k = 1$	$x_2 = 1.41666667$	$f(x_2) = 0.00694$
$k = 2$	$x_3 = 1.41421569$	$f(x_3) = 6.007 \times 10^{-6}$
$k = 3$	$x_4 = 1.41421356$	$f(x_4) = 4.51 \times 10^{-12}$
$k = 4$	$x_5 = 1.41421356$	

Since $|x_5 - x_4| < 10^{-8}$, we terminate the iteration and obtain $x^* \approx x_5 = 1.41421356$ Ans.

Geometric interpretation of Newton-Raphson method To get a good understanding of how Newton-Raphson works, let us look at the graphical interpretation of the formula (3.2). Figure 3.3 illustrates how Newton-Raphson method converges.

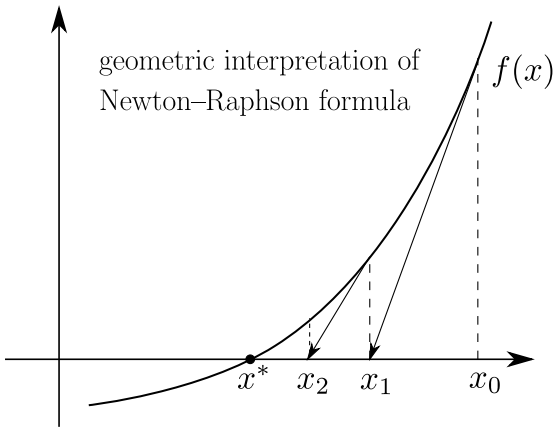

Figure 3.3: Geometric interpretation of Newton–Raphson formula

If one draws a tangent line to $f(x)$ at the given point x_k, then the tangent line intersects the x axis at the point x_{k+1}, which is expected to get closer to the root x^*.

Exercise Consider $f(x) = 0$ where $f(x) = \tanh(x)$. In this case, does the Newton-Raphson iteration converges for any initial point x_0? (You may try considering the graph of $f(x)$.) ■

Exercise For the equation $f(x) = \tanh(x) = 0$, determine the maximum value of $a > 0$ such that the Newton–Raphson iteration always converges to the root whenever $x_0 \in (0, a)$. You may use the identities: $1 - \tanh(x)^2 = 1/\cosh(x)^2$ and $\sinh(2x) = 2\sinh(x)\cosh(x)$. ■

3.1.3 Secant Method[1]

Here we consider a variation of Newton-Raphson method, in which the calculation of the derivative is avoided. The new method approximates the first derivative in the formula by using two previous points.

Recall that Newton-Raphson formula is

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

[1] A secant means a line that intersects a curve.

Then we approximate $f'(x_k)$ with $f(x_k)$ and $f(x_{k-1})$. That is,

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}.$$

Hence, we obtain the formula of the secant method as follows:

$$x_{k+1} = x_k - \frac{f(x_k)[x_k - x_{k-1}]}{f(x_k) - f(x_{k-1})} \tag{3.5}$$

The difference between Newton-Raphson and the secant methods are shown in Figure 3.4.

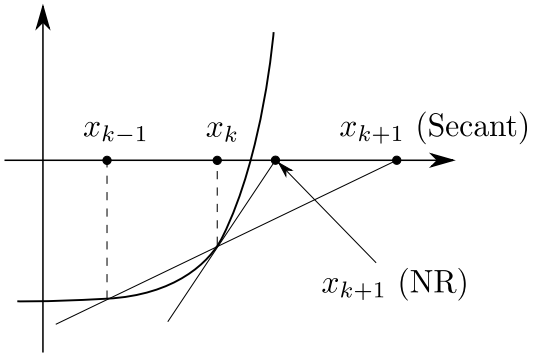

Figure 3.4: Newton-Raphson method versus secant method

3.1.4 Convergence Properties

In this subsection, we analyze the convergence properties of iterative methods given by

$$x_{k+1} = \phi(x_k), \quad k = 0, 1, 2, \ldots.$$

Usually we hope that eventually the sequence $\{x_0, x_1, \ldots, x_k, \ldots\}$ generated will converge to a solution x^* of the problem.

When discussing an iterative method, two questions are often asked:

- Does it converge?
- If it does, how fast does it converge?

Definition 3.1.2. [rate of convergence] *Assume that a sequence $\{x_k\}$ converges to a point x^*. The sequence $\{x_k\}$ is said to converge to x^* with rate r and rate constant C if*

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

If $r = 1$ with $0 < C < 1$, then the convergence is called linear.
If $r = 2$, then the convergence is called quadratic. ■

The convergence is said to be *superlinear* if $r = 1$ and $C = 0$. Superlinear convergence includes all cases where $r > 1$. This is because if

$$\lim_{k\to\infty} \frac{\| x_{k+1} - x^* \|}{\| x_k - x^* \|^r} = C < \infty \text{ for } r > 1$$

then

$$\lim_{k\to\infty} \frac{\| x_{k+1} - x^* \|}{\| x_k - x^* \|} = \lim_{k\to\infty} \frac{\| x_{k+1} - x^* \|}{\| x_k - x^* \|^r} \| x_{k+1} - x^* \|^{r-1} = 0.$$

Problem 1: The equation $f(x) = 0$ can always be written as

$$x = g(x).$$

In this connection, we call the iterative method given by

$$x_{k+1} = g(x_k)$$

the *fixed-point iteration*. For Newton-Raphson method, we readily have

$$g(x) = x - \frac{f(x)}{f'(x)};$$

$$g'(x) = 1 - \frac{f'(x)f'(x) - f(x)f''(x)}{[f'(x)]^2} = \frac{f(x)f''(x)}{[f'(x)]^2}. \tag{3.6}$$

Now consider the equation $x = g(x)$, a root of which (say x^*) can be obtained from the iteration:

$$x_{k+1} = g(x_k), \quad k = 0, 1, 2, \ldots. \tag{3.7}$$

From (3.7), it follows that

$$x_{k+1} - x^* = g(x_k) - g(x^*) = \frac{g(x_k) - g(x^*)}{x_k - x^*}(x_k - x^*). \tag{3.8}$$

If $g(x)$ and $g'(x)$ are continuous on the interval from x^* to x_k, then by the mean-value theorem we have

$$g(x_k) = g(x^*) + g'(\xi)\,(x_k - x^*) \tag{3.9}$$

where ξ lies between x^* and x_k.

From (3.8) and (3.9) it follows that

$$x_{k+1} - x^* = g'(\xi)\,(x_k - x^*)\,. \tag{3.10}$$

By defining $e_k \triangleq x_k - x^*$, (3.10) becomes

$$e_{k+1} = g'(\xi)\, e_k.$$

Thus, one can see that if $|g'(x)| \leq K < 1$ for all $x \in [x^* - h, x^* + h]$, then the iteration (3.7) always converges to the root x^* provided that $x_0 \in [x^* - h, x^* + h]$.

It is important to note that the above is only a sufficient condition, since for some equations convergence is still secured although the condition is not satisfied. Hence, from the above and from (3.6), one can conclude that if

$$\left| \frac{f(x)f''(x)}{[f'(x)]^2} \right| < 1 \quad \text{and} \quad f'(x) \neq 0 \tag{3.11}$$

on an interval about the root x^*, then Newton-Raphson will converge for any x_0 in the interval.

Problem 2: Now suppose that Newton-Raphson converges. Then we will show that the method converges with a quadratic rate.

First expand $f(x^*)$ in a Taylor series about the point x_k.

$$f(x^*) = f(x_k) + f'(x_k)(x^* - x_k) + \frac{1}{2} f''(\xi)(x^* - x_k)^2 = 0, \tag{3.12}$$

where ξ is an unknown point between x^* and x_k. If $f'(x_k) \neq 0$, then dividing (3.12) by $f'(x_k)$ and rearranging it yields

$$\left(x_k - \frac{f(x_k)}{f'(x_k)} \right) - x^* = \frac{f''(\xi)}{2f'(x_k)} (x^* - x_k)^2. \tag{3.13}$$

From Newton-Raphson iteration, (3.13) can be written as

$$x_{k+1} - x^* = \frac{f''(\xi)}{2f'(x_k)} (x^* - x_k)^2. \tag{3.14}$$

That is to say,

$$e_{k+1} = \frac{f''(\xi)}{2f'(x_k)}\, e_k^2. \tag{3.15}$$

If the iteration converges, then

$$\lim_{k \to \infty} \frac{|e_{k+1}|}{|e_k|^2} = C = \left| \frac{f''(x^*)}{2f'(x^*)} \right|. \tag{3.16}$$

Hence, one can easily see that Newton-Raphson method has a quadratic rate of convergence provided that $f'(x^*) \neq 0$.

Note in passing that it can be shown that the secant method has a superlinear rate of convergence with $r = (1 + \sqrt{5})/2 \approx 1.618$.

3.2 Multivariable Case

In this section, we study Newton-Raphson method for solving a system of nonlinear equations. This will be the basis for developing Newton method for optimization.

Consider a system of nonlinear equations:

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \qquad \text{or} \qquad \left\{ \begin{array}{rcl} f_1(\mathbf{x}) & = & 0 \\ & \vdots & \\ f_n(\mathbf{x}) & = & 0 \end{array} \right\}, \tag{3.17}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ $(i = 1, 2, \ldots, n)$ is continuous.

Let $\mathbf{x}^{(k)}$ be the current point. Define the next point $\mathbf{x}^{(k+1)} \triangleq \mathbf{x}^{(k)} + \mathbf{s}$. As before, write the Taylor series of $\mathbf{f}$ in (3.17) about the point $\mathbf{x}^{(k)} \in \mathbb{R}^n$.

$$\left. \begin{array}{rcl} f_1(\mathbf{x}^{(k)} + \mathbf{s}) & \approx & f_1(\mathbf{x}^{(k)}) + \frac{\partial f_1(\mathbf{x}^{(k)})}{\partial \mathbf{x}}^T \mathbf{s} \\ f_2(\mathbf{x}^{(k)} + \mathbf{s}) & \approx & f_2(\mathbf{x}^{(k)}) + \frac{\partial f_2(\mathbf{x}^{(k)})}{\partial \mathbf{x}}^T \mathbf{s} \\ \vdots & & \vdots \\ f_n(\mathbf{x}^{(k)} + \mathbf{s}) & \approx & f_n(\mathbf{x}^{(k)}) + \frac{\partial f_n(\mathbf{x}^{(k)})}{\partial \mathbf{x}}^T \mathbf{s} \end{array} \right\}. \tag{3.18}$$

Using the vector-matrix notation, one can rewrite (3.18) as follows:

$$\mathbf{f}(\mathbf{x}^{(k)} + \mathbf{s}) \approx f(\mathbf{x}^{(k)}) + J(\mathbf{x}^{(k)})\, \mathbf{s} \tag{3.19}$$

where the *Jacobian matrix* $J(\mathbf{x})$ of $\mathbf{f}$ is defined by

$$J(\mathbf{x}) = \left[\begin{array}{cccc} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \frac{\partial f_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial f_2(\mathbf{x})}{\partial x_1} & \frac{\partial f_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n(\mathbf{x})}{\partial x_1} & \frac{\partial f_n(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_n(\mathbf{x})}{\partial x_n} \end{array} \right] = \left[\begin{array}{c} \nabla f_1(\mathbf{x})^T \\ \nabla f_2(\mathbf{x})^T \\ \vdots \\ \nabla f_n(\mathbf{x})^T \end{array} \right]. \tag{3.20}$$

Next assume that $\mathbf{x}^{(k+1)}$ is sufficiently close to the root $\mathbf{x}^*$ so that $\mathbf{f}(\mathbf{x}^{(k+1)}) \approx \mathbf{0}$. Consequently, by using (3.19), one obtains Newton-Raphson formula.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}, \quad \text{where} \;\; J(\mathbf{x}^{(k)})\, \mathbf{s} = -\mathbf{f}(\mathbf{x}^{(k)}). \tag{3.21}$$

It should be noted that in computing $\mathbf{s}$, one should try to avoid inverting the matrix $J(\mathbf{x}^{(k)})$.

In practice, when it is not convenient to use analytical expressions of the partial derivatives, $\frac{\partial f_i(\mathbf{x})}{\partial x_j}$ can usually be approximated by finite differences. For example, the first-order forward difference approximation is given by

$$\frac{\partial f_i(\mathbf{x})}{\partial x_j} \approx \frac{f_i(\mathbf{x}+h\mathbf{e}_j) - f_i(\mathbf{x})}{h} \qquad \text{(forward difference)} \tag{3.22}$$

whereas the second-order contral difference approximation is given by

$$\frac{\partial f_i(\mathbf{x})}{\partial x_j} \approx \frac{f_i(x+h\mathbf{e}_j) - f_i(\mathbf{x}-h\mathbf{e}_j)}{2h} \qquad \text{(central difference)} \tag{3.23}$$

where $\mathbf{e}_j$ is the jth column of the identity matrix $I \in \mathbb{R}^{n\times n}$. Further details on numerical differentiation can be found in, for example, [7].

Example By using Newton-Raphson method, solve the equations:

$$\begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} x_1x_2 - x_2^3 - 1 \\ x_1^2x_2 + x_2 - 5 \end{bmatrix} = \mathbf{0}.$$

Solution: First determine the Jacobian matrix $J(\mathbf{x})$.

$$J(\mathbf{x}) = \begin{bmatrix} x_2 & x_1 - 3x_2^2 \\ 2x_1x_2 & x_1^2 + 1 \end{bmatrix}.$$

Choose $\mathbf{x}^{(0)} = [2,\, 3]^T$. Then we have

$$f(\mathbf{x}^{(0)}) = [-22, 10]^T \quad \text{and} \quad J(\mathbf{x}^{(0)}) = \begin{bmatrix} 3 & -25 \\ 12 & 5 \end{bmatrix}.$$

Using (3.21), one obtains the following results.

$k = 0$	$\mathbf{x}^{(1)} =$	$\begin{bmatrix} 1.5555\,5555\,6 \\ 2.0666\,6666\,7 \end{bmatrix}$	$\left\| f(\mathbf{x}^{(1)}) \right\|_2 = 6.92784$
$k = 1$	$\mathbf{x}^{(2)} =$	$\begin{bmatrix} 1.5472\,0541\,3 \\ 1.4777\,9333\,5 \end{bmatrix}$	$\left\| f(\mathbf{x}^{(2)}) \right\|_2 = 1.94092$
$k = 2$	$\mathbf{x}^{(3)} =$	$\begin{bmatrix} 1.7805\,3502\,9 \\ 1.1588\,6481\,1 \end{bmatrix}$	$\left\| f(\mathbf{x}^{(3)}) \right\|_2 = 0.520498$
$k = 3$	$\mathbf{x}^{(4)} =$	$\begin{bmatrix} 1.9528\,4300\,1 \\ 1.0284\,4268\,8 \end{bmatrix}$	$\left\| f(\mathbf{x}^{(4)}) \right\|_2 = 0.093554$
$k = 4$	$\mathbf{x}^{(5)} =$	$\begin{bmatrix} 1.9977\,6297\,3 \\ 1.0012\,4040\,9 \end{bmatrix}$	$\left\| f(\mathbf{x}^{(5)}) \right\|_2 = 4.4405 \times 10^{-3}$
$k = 5$	$\mathbf{x}^{(6)} =$	$\begin{bmatrix} 1.9999\,9523\,6 \\ 1.0000\,0259\,9 \end{bmatrix}$	$\left\| f(\mathbf{x}^{(6)}) \right\|_2 = 9.5346 \times 10^{-6}$
$k = 6$	$\mathbf{x}^{(7)} =$	$\begin{bmatrix} 2.0000\,0000\,0 \\ 1.0000\,0000\,0 \end{bmatrix}$	$\left\| f(\mathbf{x}^{(7)}) \right\|_2 = 4.2274 \times 10^{-11}$

It can seen that with the chosen starting point $\mathbf{x}^{(0)}$, the Newton-Raphson iteration converges rapidly to a solution $\mathbf{x}^* = [2,\ 1]^T$. ■

Exercise Try the above problem with different initial points. ■

3.3 Exercises

3.1 Apply Newton's method to solve $f(x) = x^2 - a = 0$ where $a > 0$. Prove that your iteration converges to a root if the initial point $x_0 \neq 0$. When does the iteration converge to the positive root and when does it converge to the negative root? Provide explanation to justify your answer.

3.2 Let a be some positive constant. It is possible to use Newton's method to calculate $1/a$ to any desired accuracy without doing division. Determine a function f such that $f(1/a) = 0$, and for which the formula for Newton's method only uses the arithmetic operations of addition, subtraction, and multiplication. Determine all values of the initial point x_0 for which the method converges. (Hint. You may consider the graph of the function you choose.)

3.3 A function f has a root of multiplicity $m > 1$ at the point x^* if

$$f(x^*) = f'(x^*) = \cdots = f^{(m-1)}(x^*) = 0.$$

Assume that Newton's method with initial guess x_0 converges to such a root. Prove that Newton's method converges linearly but not quadratically. Assume that the iteration

$$x_{k+1} = x_k - mf(x_k)/f'(x_k)$$

converges to x^*. If $f^{(m)}(x^*) \neq 0$, prove that this sequence converges quadratically.

Chapter 4

Optimality Conditions for Unconstrained Optimization

In this chapter, we consider an unconstrained optimization described by

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \quad \text{or} \quad \min\{f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\} \tag{4.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth nonlinear function. Specifically, we will derive several optimality conditions for the problem (4.1).

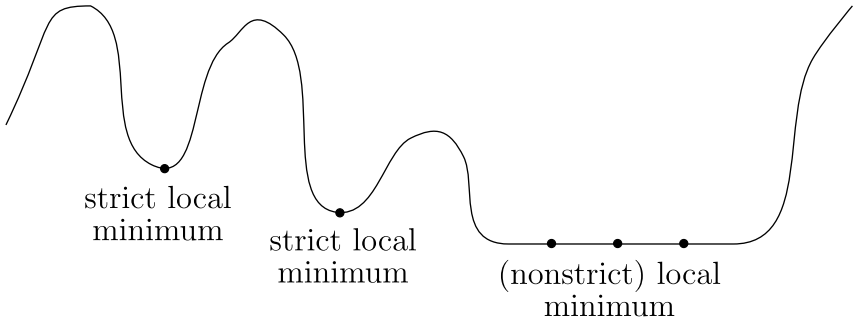

Figure 4.1: Strict and nonstrict local minima

4.1 Local and Global minima

Definition 4.1.1. [local minimizer] *Consider a function $f : \mathbb{R}^n \to \mathbb{R}$. Then a point $\mathbf{x}^* \in \mathbb{R}^n$ is said to be a* local minimizer *of f if*

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \text{for all } \mathbf{x} \text{ such that } \| \mathbf{x} - \mathbf{x}^* \| < \epsilon \tag{4.2}$$

where $\epsilon > 0$ is a (typically small) real number whose value may depend on $\mathbf{x}^*$. Similarly, a point $\mathbf{x}^* \in \mathbb{R}^n$ is said to be a strict local minimizer *of f if*

$$f(\mathbf{x}^*) < f(\mathbf{x}) \quad \text{for all } \mathbf{x} \text{ such that } \| \mathbf{x} - \mathbf{x}^* \| < \epsilon \tag{4.3}$$

■

Definition 4.1.2. [global minimizer] *A point $\mathbf{x}^* \in \mathbb{R}^n$ is said to be a* global minimizer *of f if*

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathbb{R}^n \tag{4.4}$$

A point $\mathbf{x}^* \in \mathbb{R}^n$ is said to be a strict global minimizer *of f if*

$$f(\mathbf{x}^*) < f(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathbb{R}^n \text{ such that } \mathbf{x} \neq \mathbf{x}^* \tag{4.5}$$

■

4.1.1 Convex & Concave Functions

When the objective function f has a convex property, the associated optimization problem is called a convex optimization problem, which is easier to solve both in theory and in practice.

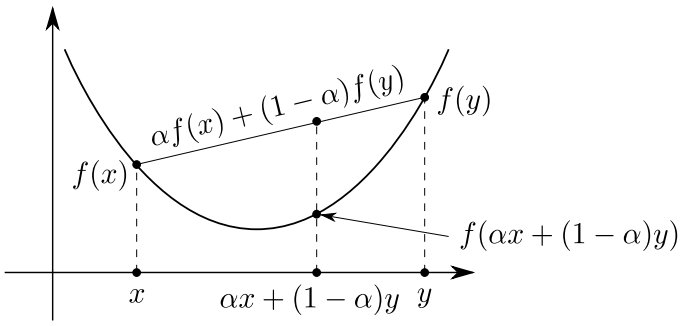

Figure 4.2: A convex function

Definition 4.1.3. [convex set] *A set $S \subset \mathbb{R}^n$ is said to be* convex *if, for any $\mathbf{x}, \mathbf{y} \in S$,*

$$\alpha\mathbf{x} + (1-\alpha)\mathbf{y} \in S \quad \text{for all } 0 \le \alpha \le 1. \tag{4.6}$$

In other words, the set S is convex if the line segment connecting any two points $\mathbf{x}, \mathbf{y} \in S$ is also in S. Notice that ■

$$\alpha\mathbf{x} + (1-\alpha)\mathbf{y} = \mathbf{y} + \alpha(\mathbf{x} - \mathbf{y}).$$

Definition 4.1.4. [convex function] *Let $S \subset \mathbb{R}^n$ be a convex set. Then a function $f : S \to \mathbb{R}$ is said to be* convex *(or* strictly convex*) if, for any $\mathbf{x}, \mathbf{y} \in S$,*

$$f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \ \le \ (\text{or } <) \ \alpha f(\mathbf{x}) + (1-\alpha) f(\mathbf{y}) \tag{4.7}$$

for all $0 \le \alpha \le 1$ (or $0 < \alpha < 1$). ■

In a similar fashion, we define a concave function as follows.

Definition 4.1.5. [concave function] *Let $S \subset \mathbb{R}^n$ be a convex set. Then a function $f : S \to \mathbb{R}$ is said to be* concave *(or* strictly concave*) if, for any $\mathbf{x}, \mathbf{y} \in S$,*

$$f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \ \ge \ (\text{or } >) \ \alpha f(\mathbf{x}) + (1-\alpha) f(\mathbf{y})$$

for all $0 \le \alpha \le 1$ (or $0 < \alpha < 1$). ■

4.1.2 Directional derivatives

Consider a function $f : \mathbb{R}^n \to \mathbb{R}$. Let $\mathbf{x} = \mathbf{x}_0 + \lambda \mathbf{s}$, where $\mathbf{x}_0, \mathbf{s} \in \mathbb{R}^n$ are given and $\lambda \in \mathbb{R}$ is a parameter. The *directional derivative* of f in the direction $\mathbf{s}$ is defined by

$$D_{\mathbf{s}} f(\mathbf{x}_0) \triangleq \lim_{\lambda \to 0} \frac{f(\mathbf{x}_0 + \lambda \mathbf{s}) - f(\mathbf{x}_0)}{\lambda}.$$

Using Taylor series, one can verify that

$$f(\mathbf{x}_0 + \lambda \mathbf{s}) = f(\mathbf{x}_0) + \lambda \mathbf{s}^T \nabla f(\mathbf{x}_0) + \frac{1}{2} \lambda^2 \mathbf{s}^T \nabla^2 f(\boldsymbol{\xi}) \mathbf{s},$$

where $\boldsymbol{\xi} \in \mathbb{R}^n$ is a point lying between $\mathbf{x}_0$ and $\mathbf{x}_0 + \lambda \mathbf{s}$. From the above, it readily follows that

$$\lim_{\lambda \to 0} \frac{f(\mathbf{x}_0 + \lambda \mathbf{s}) - f(\mathbf{x}_0)}{\lambda} = \mathbf{s}^T \nabla f(\mathbf{x}_0).$$

Hence, we have

$$D_{\mathbf{s}} f(\mathbf{x}_0) = \mathbf{s}^T \nabla f(\mathbf{x}_0).$$

When $\mathbf{x}$ is written as $\mathbf{x} = \mathbf{x}_0 + \lambda \mathbf{s}$ where $\mathbf{x}_0$ and $\mathbf{s}$ are given, one can see that $f(\mathbf{x}_0 + \lambda \mathbf{s})$ becomes a function of a single variable λ. In this connection, define

$$\phi(\lambda) \triangleq f(\mathbf{x}_0 + \lambda \mathbf{s}). \tag{4.8}$$

By applying the chain rule, one can easily see that

$$\frac{d\phi}{d\lambda} = \left(\frac{\partial f}{\partial \mathbf{x}} \right)^T \frac{d\mathbf{x}}{d\lambda} = \mathbf{s}^T \nabla f(\mathbf{x}_0 + \lambda \mathbf{s}). \tag{4.9}$$

4.2 Optimality Conditions

Now we are ready to establish three important theorems.

Theorem 4.2.1. [1st order necessary condition] *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. If $\mathbf{x}^*$ is a local minimizer of f, then $\nabla f(\mathbf{x}^*) = \mathbf{0}$.* ■

Proof Let $\mathbf{x}^*$ be a local minimizer of f, and let $\mathbf{x}$ be given by

$$\mathbf{x} = \mathbf{x}^* + \lambda \mathbf{s},$$

where $\lambda \in \mathbb{R}$ and $\mathbf{s} \in \mathbb{R}^n$. By the definition of a local minimizer and by the continuity of f, one can see that

$$\left. \frac{df}{d\lambda} \right|_{\mathbf{x} = \mathbf{x}^*} = 0, \quad \forall \mathbf{s} \in \mathbb{R}^n.$$

$$\left.\frac{df}{d\lambda}\right|_{\lambda=0} = \mathbf{s}^T \nabla f(\mathbf{x}^*) = 0, \quad \forall \mathbf{s} \in \mathbb{R}^n.$$

Hence, $\nabla f(\mathbf{x}^*) = \mathbf{0}$ and the theorem is proved. Q.E.D.

Theorem 4.2.2. [2nd order necessary condition] *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable. If $\mathbf{x}^*$ is a local minimizer of f, then $\nabla^2 f(\mathbf{x}^*)$ is positive semi-definite.* ■

Proof Suppose f is twice continuously differentiable. Consequently, $\nabla^2 f$ is continuous.

Let $\mathbf{x}^*$ be a local minimizer of f. Assume, by contradiction, that $\nabla^2 f(\mathbf{x}^*)$ is not *positive semi-definite*. Then, we can choose a vector $\mathbf{s}$ such that

$$\mathbf{s}^T \nabla^2 f(\mathbf{x}^*) \mathbf{s} < 0.$$

Because $\nabla^2 f$ is continuous near $\mathbf{x}^*$, there is a scalar $T > 0$ such that

$$\mathbf{s}^T \nabla^2 f(\mathbf{x}^* + t\mathbf{s}) \mathbf{s} < 0 \quad \forall t \in [0, T].$$

By doing a Taylor series expansion of f around $\mathbf{x}^*$, we have for all $\bar{t} \in (0, T]$ and for some $t \in (0, \bar{t})$ that

$$\begin{aligned} f(\mathbf{x}^* + \bar{t}\mathbf{s}) &= f(\mathbf{x}^*) + \bar{t}\mathbf{s}^T \nabla f(\mathbf{x}^*) + \frac{1}{2}\bar{t}^2 \mathbf{s}^T \nabla^2 f(\mathbf{x}^* + t\mathbf{s})\mathbf{s} \\ &< f(\mathbf{x}^*). \end{aligned}$$

We have found direction from $\mathbf{x}^*$ along which f is decreasing; therefore, $\mathbf{x}^*$ is not a local minimizer and we have a contradiction. So we can conclude that $\nabla^2 f(\mathbf{x}^*)$ is positive semi-definite. Q.E.D.

Theorem 4.2.3. [2nd order sufficient condition] *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable. If $\nabla f(\mathbf{x}^*) = \mathbf{0}$ and if $\nabla^2 f(\mathbf{x}^*)$ is positive definite, then $\mathbf{x}^*$ is a strict (and isolated) local minimizer of f.* ■

Proof Because $\nabla^2 f(\mathbf{x})$ is continuous and positive definite at $\mathbf{x} = \mathbf{x}^*$, we can choose a radius $r > 0$ so that $\nabla^2 f(\mathbf{x})$ remains positive definite for all $\mathbf{x}$ in the open ball

$$\mathcal{D} \triangleq \{\mathbf{x} : \|\mathbf{x} - \mathbf{x}^*\| < r\}.$$

Taking any nonzero vector $\mathbf{s}$ with $\|\mathbf{s}\| < r$, we have $\mathbf{x}^* + \mathbf{s} \in \mathcal{D}$ and therefore

$$\begin{aligned} f(\mathbf{x}^* + \mathbf{s}) &= f(\mathbf{x}^*) + \mathbf{s}^T \nabla f(\mathbf{x}^*) + \frac{1}{2}\, \mathbf{s}^T \nabla^2 f(\mathbf{z}) \mathbf{s} \\ &= f(\mathbf{x}^*) + \frac{1}{2}\, \mathbf{s}^T \nabla^2 f(\mathbf{z}) \mathbf{s} \end{aligned}$$

where $\mathbf{z} = \mathbf{x}^* + t\mathbf{s}$ for some $t \in (0, 1)$. Since $\mathbf{z} \in \mathcal{D}$, we have $\mathbf{s}^T \nabla^2 f(\mathbf{z})\, \mathbf{s} > 0$. Hence, it is easy to see that

$$f(\mathbf{x}^* + \mathbf{s}) > f(\mathbf{x}^*) \qquad \forall \mathbf{s} \in \mathcal{D}$$

and the theorem is proved. Q.E.D.

4.3 Exercises

4.1 Let f be a real-valued function of n variables.

(a) Assume that f is continuously differentiable. Then prove that f is convex on the convex set S if and only if

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \quad \text{for all } \mathbf{x}, \mathbf{y} \in S.$$

(b) Assume that f is twice continuously differentiable. Then prove that f is convex on the convex set S if $\nabla^2 f(\mathbf{x})$ is positive semi-definite for all $\mathbf{x} \in S$, by using the result obtained from part (a).

4.2 Prove that if f is convex, then any stationary point is also a global minimizer.

4.3 Let $f : \mathbb{R}^2 \to \mathbb{R}$ be defined by

$$f(\mathbf{x}) = (x_1 - x_2)^4 + x_1^2 - x_2^2 - 2x_1 + 2x_2 + 1$$

where $\mathbf{x} = [x_1, x_2]^T$. Suppose that we wish to minimize f over $\mathbb{R}^2$. Find all points satisfying the first-order necessary condition. Do these points satisfy the second-order necessary condition?

4.4 Let

$$f(\mathbf{x}) = 2x_1^2 + x_2^2 - 2x_1x_2 + 2x_1^3 + x_1^4$$

where $\mathbf{x} = [x_1, x_2]^T$. Determine the minimizers/maximizers of f and indicate what kind of minima or maxima (local, global, strict, etc.) they are.

4.5 Consider "Rosenbrock's function": $f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ where $\mathbf{x} = [x_1, x_2]^T$ (know to be a "nasty" function–often used as a benchmark for testing algorithms). Determine all the local minimizers by using the second-order conditions. What is the global minimum of f?.

Chapter 5

Numerical Methods for Unconstrained Optimization

Many optimization algorithms, especially those that we are interested in for this course, takes a general form:

Specify an initial guess $\mathbf{x}^{(0)} \in \mathbb{R}^n$ and $k_{\max}$.
  For $k = 0, 1, \ldots, k_{\max}$ {
    If $\mathbf{x}^{(k)}$ is optimal, then stop.
    Determine a new point: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}$.}

where $\mathbf{s}^{(k)} \in \mathbb{R}^n$ is called a *search direction* and the scalar λ_k is called a *step length*.

Obviously, the above algorithm model suggests that we need to solve the two problems: *viz.*, testing the optimality and determining $\mathbf{x}^{(k+1)}$.

The optimality test are carried out using the conditions introduced in Chapter 4. Furthermore, the information obtained from the optimal test is often the basis for computing $\mathbf{x}^{(k+1)}$.

For the minimization algorithms to be described subsequently, the search direction $\mathbf{s}^{(k)}$ is computed so that $f(\mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)})$ decreases for sufficiently small $\lambda_k > 0$. Once $\mathbf{s}^{(k)}$ is obtained, the step length λ_k needs to be calculated by solving a one-dimensional search problem. Ideally, it is required that α_k minimizes $f(\mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)})$. However, with some good minimization algorithms, α_k is often chosen so that $f(\mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}) < f(\mathbf{x}^{(k)})$ with a sufficient degree.

5.1 One-Dimensional Optimization

Definition 5.1.1. [unimodal function] *A function $f : \mathbb{R} \to \mathbb{R}$ is said to be* unimodal *in the interval $a \le x \le b$ if it has only one extremum* [1] *within the interval.* ■

Note that a convex (or concave) function is always unimodal. But the converse is not necessary true.

One important aspect of the single variable minimization problem is that several of multivariable algorithms implement a sequence of single variable minimizations along carefully chosen lines (or directions) in the higher dimensional space $\mathbb{R}^n$.

During the process of minimizing a function $f : \mathbb{R}^n \to \mathbb{R}$, once a point $\mathbf{x}^{(k)}$ and a search direction $\mathbf{s}^{(k)}$ are chosen such that $\mathbf{x} = \mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)}$, then the problem of minimizing f in $\mathbb{R}^n$ becomes that of $f(x_k + \lambda s_k)$ in $\mathbb{R}$.

[1] An extremum is either a minimum or a maximum

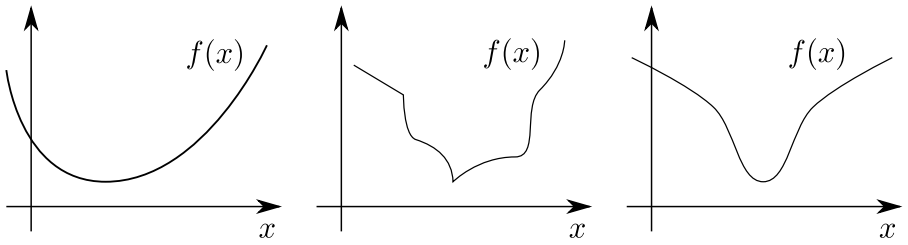

Figure 5.1: Unimodal functions

For convenience, define $f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)}) \triangleq \phi(\lambda)$. Then

$$\phi'(\lambda) = \mathbf{s}^{(k)T} \nabla f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)}).$$

The following proposition is useful for determining an optimal step length λ^*, for which $\phi(\lambda^*)$ is minimal.

Proposition 5.1.2. *Suppose that a function $f : \mathbb{R} \to \mathbb{R}$ is unimodal. Then f has a minimum in the interval $[a, c]$ if one of the following holds:*

- $f'(a) < 0 < f'(c)$;
- *there exists $b \in (a, c)$ such that $f(b) < f(a)$ and $f(b) < f(c)$.* ■

5.1.1 Quadratic Interpolation Method

This method was first suggested by Powell [31] for function minimization without using derivatives.

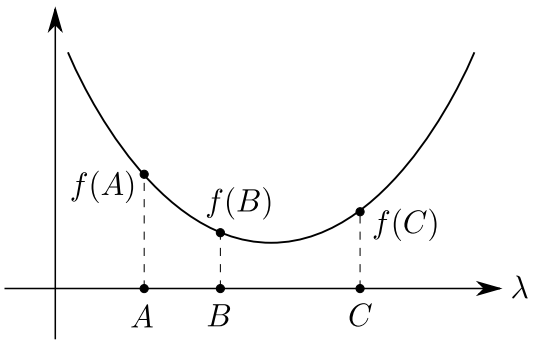

Figure 5.2: Quadratic interpolation method

Suppose that three points $\lambda = A$, $\lambda = B$, $\lambda = C$ are given such that $A < B < C$, $f(B) < f(A)$ and $f(B) < f(C)$. If f is approximated by a quadratic function

$$h(\lambda) = a + b\lambda + c\lambda^2$$

within $[A, C]$, then an estimate of the minimum of f (denoted by $\tilde{\lambda}^*$) is given by the minimum of h. Hence,

$$\tilde{\lambda}^* = \frac{f(A)\{B^2 - C^2\} + f(B)\{C^2 - A^2\} + f(C)\{A^2 - B^2\}}{2\left[f(A)\{B - C\} + f(B)\{C - A\} + f(C)\{A - B\}\right]}. \tag{5.1}$$

In addition, the coefficients a, b, c are given by

$$a = \frac{f(A)BC\{C - B\} + f(B)CA\{A - C\} + f(C)AB\{B - A\}}{(A - B)(B - C)(C - A)}.$$

$$b = \frac{f(A)\{B^2 - C^2\} + f(B)\{C^2 - A^2\} + f(C)\{A^2 - B^2\}}{(A - B)(B - C)(C - A)}.$$

$$c = -\frac{f(A)\{B - C\} + f(B)\{C - A\} + f(C)\{A - B\}}{(A - B)(B - C)(C - A)}.$$

See, for example, [36] for further details.

After the estimate $\tilde{\lambda}^*$ is obtained, there are now four points bracketing the minimum of f; namely, $\lambda = A$, $\lambda = B$, $\lambda = C$ and $\lambda = \tilde{\lambda}^*$. Out of these four points, one can choose the best three points that bracket the minimum. In doing so, there are four possibilities.

	IF	THEN
Case 1	$A < \tilde{\lambda}^* < B, f(\tilde{\lambda}^*) < f(B)$	$A_{\text{new}} \leftarrow A, B_{\text{new}} \leftarrow \tilde{\lambda}^*, C_{\text{new}} \leftarrow B$
Case 2	$A < \tilde{\lambda}^* < B, f(\tilde{\lambda}^*) \geq f(B)$	$A_{\text{new}} \leftarrow \tilde{\lambda}^*, B_{\text{new}} \leftarrow B, C_{\text{new}} \leftarrow C$
Case 3	$B < \tilde{\lambda}^* < C, f(\tilde{\lambda}^*) < f(B)$	$A_{\text{new}} \leftarrow B, B_{\text{new}} \leftarrow \tilde{\lambda}^*, C_{\text{new}} \leftarrow C$
Case 4	$B < \tilde{\lambda}^* < C, f(\tilde{\lambda}^*) \geq f(B)$	$A_{\text{new}} \leftarrow A, B_{\text{new}} \leftarrow B, C_{\text{new}} \leftarrow \tilde{\lambda}^*$

With such new points, refit the quadratic again to get a better approximation.

A criterion for stopping the quadratic refitting is, for example,

$$\left|\frac{h(\tilde{\lambda}^*) - f(\tilde{\lambda}^*)}{f(\tilde{\lambda}^*)}\right| \leq \varepsilon \quad \text{when } f(\tilde{\lambda}^*) \neq 0. \tag{5.2}$$

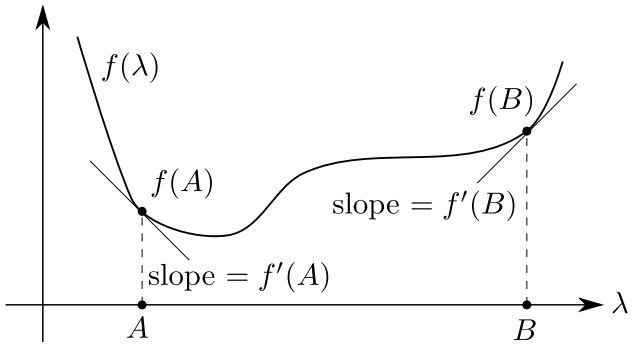

Figure 5.3: Cubic interpolation method

5.1.2 Cubic Interpolation Method

This method was introduced by Davidon [10, 11] in conjunction with minimization by using variable metric methods.

Suppose that two points $\lambda = A$, $\lambda = B$ are given such that $A < B$, $f'(A) < 0$ and $f'(B) > 0$. If f is approximated by a cubic function

$$h(\lambda) = a + b\lambda + c\lambda^2 + d\lambda^3$$

within $[A,\ B]$, then an estimate of the minimum of f (denoted by $\tilde{\lambda}^*$) is given by the minimum of h. Hence,

$$\tilde{\lambda}^* = A + \frac{f'(A) + Z + Q}{f'(A) + f'(B) + 2Z}(B - A) \tag{5.3}$$

where

$$Z = \frac{3\{f(A) - f(B)\}}{B - A} + f'(A) + f'(B),$$
$$Q = \left\{Z^2 - f'(A)f'(B)\right\}^{1/2}.$$

For the details on cubic interpolation, see [17] and [36].

After the estimate $\tilde{\lambda}^*$ is obtained, there are now three points bracketing the minimum of f; namely, $\lambda = A$, $\lambda = \tilde{\lambda}^*$ and $\lambda = B$. Out of these three points, one can choose two points that bracket the minimum by looking at the slope of f.

A criterion one may use to terminate the iteration is, for example,

$$\left|f'(\tilde{\lambda}^*)\right| \le \varepsilon \quad \text{or} \quad \left|\tilde{\lambda}^*_{k+1} - \tilde{\lambda}^*_k\right| \le \varepsilon$$

where $\varepsilon > 0$ is specified. If the criterion is not satisfied, refit the cubic function again.

5.1.3 Golden Section Method

The golden section method arranges the size of the interval containing the minimum such that it is decreased by a constant factor τ at each step. In each iteration, only one function evaluation is required.

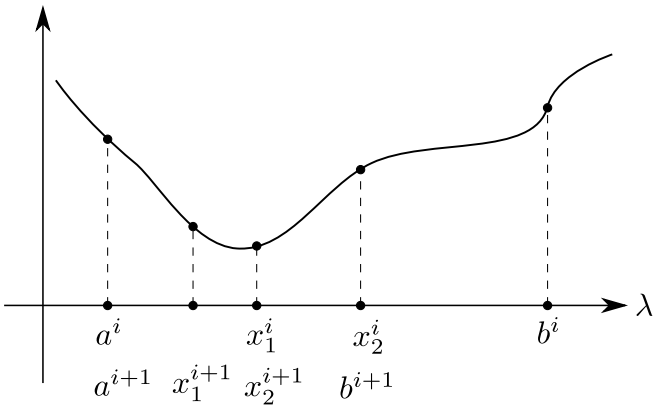

Figure 5.4: Golden section method

Let the current interval be (a^i, b^i) and let x_1^i and x_2^i be the points at which the function is evaluated where $a^i < x_1^i < x_2^i < b^i$. Then these points must satisfy

$$\frac{x_2^i - a^i}{b^i - a^i} = \frac{b^i - x_1^i}{b^i - a^i} = \tau, \tag{5.4}$$

from which it follows that

$$x_1^i - a = b^i - x_2^i. \tag{5.5}$$

Assume that $f(x_2^i) > f(x_1^i)$, so that we have

$$b^{i+1} = x_2^i \quad \text{and} \quad a^{i+1} = a^i \quad \text{and} \quad x_2^{i+1} = x_1^i.$$

This is sufficient to specify the procedure precisely and therefore

$$\frac{x_2^{i+1} - a^i}{x_2^i - a^i} = \frac{x_1^i - a^i}{x_2^i - a^i} = \frac{x_2^i - a^i}{b^i - a^i} = \tau \tag{5.6}$$

and by (5.5)

$$x_1^i - a^i = b^i - a^i - \left(x_2^i - a^i\right) \tag{5.7}$$

so that

$$\frac{x_1^i - a^i}{x_2^i - a^i} = -1 + \frac{1}{\tau} = \tau. \tag{5.8}$$

Hence, one obtains

$$\tau^2 + \tau - 1 = 0 \quad \Longrightarrow \quad \tau = \frac{\sqrt{5}-1}{2} \approx 0.618 \quad \text{and} \quad 1 - \tau \approx 0.382.$$

The implementation of this algorithm is as follows:

If $f(x_2^i) > f(x_1^i)$ then
  $b^{i+1} = x_2^i; \qquad a^{i+1} = a^i;$
  $x_2^{i+1} = x_1^i;$
  $x_1^{i+1} = a^i + (1-\tau)(b^{i+1} - a^i);$
else
  $a^{i+1} = x_1^i; \qquad b^{i+1} = b^i;$
  $x_1^{i+1} = x_2^i;$
  $x_2^{i+1} = b^i - (1-\tau)(b^i - a^{i+1});$
end

5.2 Methods for Unconstrained Optimization

There are two types of methods for optimization:

- methods that do not require derivative (or direct search methods)
- methods that require derivatives (or descent methods): steepest descent method, Newton methods, conjugate gradient methods, quasi Newton (or variable metric) methods.

In our course, we focus attention only to descent methods.

Algorithm model: Suppose that a function $f : \mathbb{R}^n \to \mathbb{R}$ and a starting point $\mathbf{x}^{(0)} \in \mathbb{R}^n$ are given. Let $k = 0$.

1. Determine a search direction $\mathbf{s}^{(k)}$.
2. Find λ_k to minimize $f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)})$ with respect to λ (or at least to reduce f sufficiently).
3. Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}$.
4. If the termination criterion is not satisfied, then set $k = k+1$ and goto step 1; otherwise terminate with $\mathbf{x}^* = \mathbf{x}^{(k)}$.

In step 1, different methods correspond to different ways of choosing the search direction $\mathbf{s}^{(k)}$, depending on information available in the problem.

Step 2 is called *the line search subprogram* and is carried out by repeatedly sampling $f(\mathbf{x})$ and possibly its derivatives for different points $\mathbf{x} = \mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)}$ along the direction $\mathbf{s}^{(k)}$.

Ideally, the exact minimizing value of λ (*an exact line search*) would be required and this cannot be implemented in practice with a finite number of operations. Nevertheless, the idea is conceptually useful and occurs in some idealized proofs of convergence.

One can easily deduce that for an exact line search, the optimal step length λ_k must satisfy

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\phi(\lambda_k) = \mathbf{s}^{(k)T}\nabla f(\mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}) = 0.$$

To terminate the iterations, one may use the following criteria.

$$\begin{aligned}
& \left\| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\| && \leq \quad \varepsilon_1, \\
\text{or} \qquad & \left\| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\| && \leq \quad \varepsilon_2 \left\| \mathbf{x}^{(k)} \right\|, \\
\text{or} \qquad & \left| f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)}) \right| && \leq \quad \varepsilon_3, \\
\text{or} \qquad & \left| f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)}) \right| && \leq \quad \varepsilon_4 \left| f(\mathbf{x}^{(k)}) \right|.
\end{aligned}$$

Consider a function $f \in C^1$ and let $\mathbf{x} = \mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)}$. Then a search direction $\mathbf{s}^{(k)}$ is said to satisfy the *descent property* if

$$\mathbf{s}^{(k)T}\nabla f(\mathbf{x}^{(k)}) < 0.$$

This guarantees that the function f can be reduced in the line search for some $\lambda_k > 0$. In certain cases, by a suitable choice of line search conditions, it is possible to incorporate the descent property into a convergence proof.

Quadratic functions A *quadratic function* is written as

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{p}^T \mathbf{x} + r$$

where $Q \in \mathbb{R}^{n \times n}$ is symmetric, $\mathbf{p} \in \mathbb{R}^n$ and $r \in \mathbb{R}$. It can be easily verified that

$$\nabla f(\mathbf{x}) = Q\mathbf{x} + \mathbf{p} \quad \text{and} \quad \nabla^2 f(\mathbf{x}) = Q.$$

Accordingly, one can deduce that the minimizer of f is given by

$$\mathbf{x}^* = -Q^{-1}\mathbf{p}.$$

Note also that the quadratic function, though simple, arises in many applications. Therefore, it is often used as a model in the analysis and design of optimization methods.

5.2.1 Steepest Descent Method

The steepest descent is the simplest descent method and was invented by Cauchy in the 19th century. The method has theoretical uses in proving the convergence of other methods and in providing lower bounds on the performance of better algorithms. Therefore, it is well known and has been widely used although, for most practical problems, the method is hopelessly inefficient.

The steepest descent method computes the search direction from

$$\mathbf{s}^{(k)} = -\nabla f(\mathbf{x}^{(k)}) \tag{5.9}$$

and then uses a line search to determine $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}$.

Proposition 5.2.1. *In minimizing a function $f : \mathbb{R}^n \to \mathbb{R}$, every search direction generated by the steepest descent method has a descent property.* ■

Proof It is easily verified that

$$\frac{\mathrm{d}\phi}{\mathrm{d}\lambda}(0) = \frac{\mathrm{d}f}{\mathrm{d}\lambda}(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)})\bigg|_{\lambda=0} = -\nabla f(\mathbf{x}^{(k)})^T \nabla f(\mathbf{x}^{(k)}) < 0.$$

Hence, one can conclude that there always exist some $\lambda > 0$ such that $f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)}) < f(\mathbf{x}^{(k)})$. Q.E.D.

Furthermore, it should be noted that at the point $\mathbf{x}^{(k)}$, the search direction $\mathbf{s}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$ gives the maximal rate of decrease of the function f (assuming $||\mathbf{s}^{(k)}|| = 1$).

To prove this statement, recall that the directional derivative of f along a line $\mathbf{s}^{(k)}$ is given by

$$D_{\mathbf{s}^{(k)}} f(\mathbf{x}^{(k)}) = \phi'(0) = \mathbf{s}^{(k)T} \nabla f(\mathbf{x}^{(k)}).$$

Since $\mathbf{s}^{(k)T} \nabla f(\mathbf{x}^{(k)}) = \| \mathbf{s}^{(k)} \| \| \nabla f(\mathbf{x}^{(k)}) \| \cos\theta$, one can see that $|\phi'(0)|$ has the maximal value when $\mathbf{s}^{(k)}$ and $\nabla f(\mathbf{x}^{(k)})$ have either the same or opposite direction.

Example Consider a quadratic function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{p}^T \mathbf{x}$, where Q is positive definite. If the steepest descent method is used with an exact line search, determine the optimal step length λ_k^* at each step k.

Solution Let $\mathbf{x} = \mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)}$ and define $\phi(\lambda) \triangleq f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)})$. Then it readily follows that

$$\phi(\lambda) = \frac{1}{2}\left[\mathbf{x}^{(k)T} Q \mathbf{x}^{(k)} + 2\lambda \mathbf{s}^{(k)T} Q \mathbf{x}^{(k)} + \lambda^2 \mathbf{s}^{(k)T} Q \mathbf{x}^{(k)}\right] + \mathbf{p}^T \mathbf{x}^{(k)} + \lambda \mathbf{p}^T \mathbf{s}^{(k)},$$

and

$$\phi'(\lambda) = \mathbf{s}^{(k)T} Q \mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)T} Q \mathbf{s}^{(k)} + \mathbf{p}^T \mathbf{x}^{(k)}.$$

Determine λ_k^* using the fact that $\phi'(\lambda_k^*) = 0$. Consequently,

$$\lambda_k^* = \frac{-\mathbf{s}^{(k)T}(Q\mathbf{x}^{(k)} + \mathbf{p})}{\mathbf{s}^{(k)T} Q \mathbf{s}^{(k)}} = \frac{-\mathbf{s}^{(k)T} \nabla f(\mathbf{x}^{(k)})}{\mathbf{s}^{(k)T} Q \mathbf{s}^{(k)}}. \tag{5.10}$$

It should be noted here that (5.10) provides a formula for computing the optimal step length λ_k^* for general methods minimizing the quadratic function. For the steepest descent method, $\mathbf{s}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$ and hence

$$\lambda_k^* = \frac{\nabla f(\mathbf{x}^{(k)})^T \nabla f(\mathbf{x}^{(k)})}{\nabla f(\mathbf{x}^{(k)})^T Q \nabla f(\mathbf{x}^{(k)})}. \qquad \text{Ans}$$

Next we will show the steepest descent method with an exact line search generates the iterates $\mathbf{x}^{(k)}$ that zigzag towards the solution.

Proposition 5.2.2. *If a sequence $\{\mathbf{x}^{(k)}\}$ is obtained from minimizing a function $f : \mathbb{R}^n \to \mathbb{R}$ using the steepest descent method with an exact line search, then for each k the vector $\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ is orthogonal to the vector $\mathbf{x}^{(k+2)} - \mathbf{x}^{(k+1)}$.* ■

Proof From the steepest descent method (5.9), it follows that

$$\left(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right)^T \left(\mathbf{x}^{(k+2)} - \mathbf{x}^{(k+1)}\right) = \lambda_k \lambda_{k+1} \nabla f(\mathbf{x}^{(k)})^T \nabla f(\mathbf{x}^{(k+1)}).$$

When an exact line search is used, one determines λ_k that satisfies

$$\phi'(\lambda_k) = 0 = \nabla f(\mathbf{x}^{(k)})^T \nabla f(\mathbf{x}^{(k+1)}).$$

Hence the proof is completed. Q.E.D.

It is worth noting the rate of convergence of the steepest descent method in the ideal case (that is, when f is a quadratic function and the line searches are exact). The result is obtained by considering $f(\mathbf{x}) - f(\mathbf{x}^*)$ instead of $||\mathbf{x} - \mathbf{x}^*||$ because the analysis is simpler.

Theorem 5.2.3 ([26])**.** *Assume that $\{\mathbf{x}^{(k)}\}$ is the sequence of approximate solution obtained when the steepest descent method is applied to the quadratic function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q \mathbf{x} +$*

$\mathbf{p}^T\mathbf{x}$ ($Q > 0$), *and when an exact line search is used. Then, for any* $\mathbf{x}^{(0)}$, *the method converges to the unique minimizer* $\mathbf{x}^*$ *of* f, *and furthermore*

$$f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^*) \leq \left[\frac{A-a}{A+a}\right]^2 \left\{f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*)\right\}$$

where A *and* a *are, respectively, the largest and smallest eigenvalues of the matrix* Q. ■

Proof See [24]. Q.E.D.

For general nonlinear functions, it is possible to show that the steepest descent method (with an exact line search) also converges linearly, with a rate constant that is bounded by

$$\left[\frac{A-a}{A+a}\right]^2$$

where A and a are the largest and smallest eigenvalues of $\nabla^2 f(\mathbf{x}^*)$.

5.2.2 Newton's Method

From Chapter 3, one can see that Newton's (or Newton-Raphson) method is an efficient algorithm for determining a solution of nonlinear equations.

In solving optimization problem, Newton's method is used to find a solution of the first-order necessary conditions for a local minimizer, which is in fact a set of nonlinear equations.

Consider a function $f : \mathbb{R}^n \to \mathbb{R}$ where $f \in C^2$. To derive Newton's method for minimization, we apply Newton-Raphson method to the first order necessary conditions

$$\nabla f(\mathbf{x}) = \mathbf{0}.$$

Since the Jacobian of $\nabla f(\mathbf{x})$ is $\nabla^2 f(\mathbf{x})$, the update formula of Newton's method for minimization is given by

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \left[\nabla^2 f(\mathbf{x}^{(k)})\right]^{-1} \nabla f(\mathbf{x}^{(k)}).$$

Instead of the above form, we prefer writing Newton's formula as

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)} \quad \text{where} \quad \left[\nabla^2 f(\mathbf{x}^{(k)})\right] \mathbf{s}^{(k)} = -\nabla f(\mathbf{x}^{(k)}). \tag{5.11}$$

Formula (5.11) is sometimes called *classical* Newton method.

Next we will look at another interpretation of Newton's method; the details of which can be found in [26] or [14]. The second-order Taylor series approximant of the function f about

the point $\mathbf{x}^{(k)}$ is given by

$$q(\boldsymbol{\delta}) \approx f(\mathbf{x}^{(k)}) + \boldsymbol{\delta}^T \nabla f(\mathbf{x}^{(k)}) + \frac{1}{2}\boldsymbol{\delta}^T \nabla^2 f(\mathbf{x}^{(k)})\boldsymbol{\delta}$$

where $\boldsymbol{\delta} = \mathbf{x} - \mathbf{x}^{(k)}$. Then one obtains

$$\nabla q(\boldsymbol{\delta}) = \nabla f(\mathbf{x}^{(k)}) + \nabla^2 f(\mathbf{x}^{(k)})\boldsymbol{\delta}.$$

If $\nabla^2 f(\mathbf{x}^{(k)}) > 0$, then the function q has the minimum at

$$\boldsymbol{\delta} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} = -\left[\nabla^2 f(\mathbf{x}^{(k)})\right]^{-1} \nabla f(\mathbf{x}^{(k)}). \tag{5.12}$$

Obviously, (5.12) leads to the formula (5.11).

Now one can see that at every iteration, Newton's method approximates $f(\mathbf{x})$ by the quadratic model $q(\boldsymbol{\delta})$; minimizes q with respect to $\boldsymbol{\delta}$; and then sets $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \boldsymbol{\delta}$.

How efficiently does Newton's method perform when minimizing a quadratic function? The answer is as follows.

Proposition 5.2.4. *When Newton's method is used to minimize a quadratic function* $f(x) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{p}^T\mathbf{x}$ *where* $Q > 0$*, it takes only* one step*, regardless the staring point.* ■

Because this is quite easy, the proof is left as an exercise for students.

Newton's method has a quadratic rate of convergence except in *degenerate* cases. This property is stated rigorously as follows.

Theorem 5.2.5. *Suppose that* $f \in C^2$ *and that the Hessian matrix satisfies a Lipschitz condition* $\| \nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y}) \| \leq L \| \mathbf{x} - \mathbf{y} \|$ *in a neighbourhood of a local minimizer* $\mathbf{x}^*$*. If* $\mathbf{x}^{(k)}$ *is sufficiently close to* $\mathbf{x}^*$ *for some* k *and if* $\nabla^2 f(\mathbf{x}^*)$ *is pdf, then Newton's method is well-defined for all* k *and converges to* $\mathbf{x}^*$ *at a quadratic rate.* ■

Proof The detail of the proof can be found in [29]. Here only a sketch of the proof is given. Students should work out the details themselves.

From (5.11) and $\nabla f(\mathbf{x}^*) = \mathbf{0}$, it follows that

$$\begin{aligned}\mathbf{x}^{(k)} + \mathbf{s}^{(k)} - \mathbf{x}^* &= \mathbf{x}^{(k)} - \mathbf{x}^* - \nabla^2 f(\mathbf{x}^{(k)})^{-1} \nabla f(\mathbf{x}^{(k)}) \\ &= \nabla^2 f(\mathbf{x}^{(k)})^{-1} \Big[\nabla^2 f(\mathbf{x}^{(k)})[\mathbf{x}^{(k)} - \mathbf{x}^*] - [\nabla f(\mathbf{x}^{(k)}) - \nabla f(\mathbf{x}^*)] \Big].\end{aligned}$$

Using the fact that

$$\nabla f(\mathbf{x}) - \nabla f(\mathbf{y}) = \int_0^1 \nabla^2 f\big(\mathbf{x} + t[\mathbf{y} - \mathbf{x}]\big)[\mathbf{x} - \mathbf{y}]\, dt,$$

we have

$$\begin{aligned} & \Big\| \nabla^2 f(\mathbf{x}^{(k)})[\mathbf{x}^{(k)} - \mathbf{x}^*] - \big[\nabla f(\mathbf{x}^{(k)}) - \nabla f(\mathbf{x}^*)\big] \Big\| \\ \leq \ & \big\|\mathbf{x}^{(k)} - \mathbf{x}^*\big\|^2 \int_0^1 Lt\, dt = \frac{1}{2} L \big\|\mathbf{x}^{(k)} - \mathbf{x}^*\big\|^2, \end{aligned}$$

where L is the Lipschitz constant of $\nabla^2 f(\mathbf{x})$ for $\mathbf{x}$ near $\mathbf{x}^*$. From the above, we can deduce that

$$\big\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\big\| \leq L \big\|\nabla^2 f(\mathbf{x}^*)^{-1}\big\| \ \big\|\mathbf{x}^{(k)} - \mathbf{x}^*\big\|^2.$$

Using this inequality inductively, we deduce that if $\mathbf{x}^{(0)}$ is sufficiently near $\mathbf{x}^*$, then $\{\mathbf{x}^{(k)}\}$ converges to $\mathbf{x}^*$ and the rate of convergence is quadratic. Q.E.D.

Newton's method is rarely used in the classical form (5.11). It does not always converge; even if it does, it might not converge to a minimizer.

By introducing an additional variable $\lambda \in \mathbb{R}$ in (5.11), we have *modified* Newton's method as follows:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}, \qquad \big[\nabla^2 f(\mathbf{x}^{(k)})\big]\, \mathbf{s}^{(k)} = -\nabla f(\mathbf{x}^{(k)}) \tag{5.13}$$

where λ_k is chosen so that $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$ (or ideally to minimize f along the search direction). Indeed, the formula (5.13) is Newton's method used in practice for minimization.

It is interesting to know if Newton's method always generates a descent search direction. The answer is as follows.

Proposition 5.2.6. *Consider a function $f : \mathbb{R}^n \to \mathbb{R}$ where $f \in C^1$ and let $\mathbf{x} = \mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}$. For any positive definite matrix $H_k \in \mathbb{R}^{n \times n}$, a search direction $\mathbf{s}^{(k)} = -H_k \nabla f(\mathbf{x}^{(k)})$ always has a descent property.* ■

Proof Let $\phi(\lambda) = f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)})$. Then it can be verified that

$$\phi'(0) = \mathbf{s}^{(k)T} \nabla f(\mathbf{x}^{(k)}) = -\nabla f(\mathbf{x}^{(k)})^T H_k \nabla f(\mathbf{x}^{(k)}) \tag{5.14}$$

Since $H_k > 0$, $\phi'(0) < 0$ and therefore $f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)}) < f(\mathbf{x}^{(k)})$ for $\lambda > 0$ sufficiently small. Hence, the proposition is proved. Q.E.D.

Now one can readily deduce that at each step of Newton's method (5.13) if the matrix $\nabla^2 f(\mathbf{x}^{(k)})$ is pdf, then the search direction $\mathbf{s}^{(k)}$ has a descent property. That is, a local minimum exists for $\lambda > 0$.

There are three types of costs in implementing Newton's method:

- computation of derivatives,
- computation of $\mathbf{s}^{(k)}$,
- storage.

In its classical form, Newton's method requires second derivatives, the solution of a linear equation and the storage of a matrix. For an n-variable problem, there are $n(n+1)/2$ entries in the Hessian matrix. Once the Hessian matrix has been found, it costs $\mathcal{O}(n^3)$ arithmetic operations to solve the linear equation in (5.13). And normally, the Hessian matrix will have to be stored at a cost of $n(n+1)/2$ storage locations. As n increases, these costs grow rapidly.

Calculation of the Hessian When analytical expressions of the Hessian matrix $\nabla^2 f(\mathbf{x})$ are not available, one may compute $\nabla^2 f(\mathbf{x}^{(k)})$ by the following means. If the gradient $\nabla f(\mathbf{x})$ can be determined analytically, then an approximation to the ith column of $\nabla^2 f(\mathbf{x}^{(k)})$ is given by

$$y_i = \frac{1}{h_i}\big(\nabla f(\mathbf{x}^{(k)} + h_i\mathbf{e}_i) - \nabla f(\mathbf{x}^{(k)})\big) \qquad \text{forward difference}$$

$$\text{or} \quad y_i = \frac{1}{2h_i}\big(\nabla f(\mathbf{x}^{(k)} + h_i\mathbf{e}_i) - \nabla f(\mathbf{x}^{(k)} - h_i\mathbf{e}_i)\big) \qquad \text{central difference}$$

where $\mathbf{e}_i$ is the ith column of the identity matrix.

Alternatively, it may be possible to approximate $\nabla^2 f(\mathbf{x}^{(k)})$ by

$$\big[\nabla^2 f(\mathbf{x}^{(k)})\big]_{ij} \approx \frac{f(\mathbf{x}^{(k)} + h_i\mathbf{e}_i + h_j\mathbf{e}_j) - f(\mathbf{x}^{(k)} + h_i\mathbf{e}_i) - f(\mathbf{x}^{(k)} + h_j\mathbf{e}_j) + f(\mathbf{x}^{(k)})}{h_i h_j}.$$

If the approximant matrix Y obtained from the above is not symmetric, then the matrix H used to approximate the Hessian is given by

$$H = \frac{1}{2}(Y + Y^T)$$

One should, however, bear in mind that the use of finite difference approximation may be unsatisfactory especially if $f(\mathbf{x})$ is noisy.

LDL^T factorization If $A \in \mathbb{R}^{n\times n}$ is pdf, then A can always be written as $A = LDL^T$, where L is a lower triangular matrix and $D = \text{diag}(d_1, \ldots, d_n)$, $d_i > 0$. Notice that this is in fact a variation of LU factorization. More importantly, in this case, one is able to compute the factors L and D without having to perform partial pivoting.

To find a formula for the factorization, consider a matrix $A \in \mathbb{R}^{4\times 4}$

$$\begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{21} & a_{22} & a_{32} & a_{42} \\ a_{31} & a_{32} & a_{33} & a_{43} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} D \begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix}^T$$

From the above, one can deduce that

$$\begin{aligned}
\text{for } j = 1: \qquad & d_1 = a_{11}, \\
& l_{21} = a_{21}/d_1, \\
& l_{31} = a_{31}/d_1, \\
& l_{41} = a_{41}/d_1; \\
\text{for } j = 2: \qquad & d_2 = a_{22} - d_1 l_{21}^2, \\
& l_{32} = \Big(a_{32} - d_1 l_{21} l_{31}\Big)/d_2, \\
& l_{42} = \Big(a_{42} - d_1 l_{21} l_{41}\Big)/d_2; \\
\text{for } j = 3: \qquad & d_3 = a_{33} - d_1 l_{31}^2 - d_2 l_{32}^2, \\
& l_{43} = \Big(a_{43} - d_1 l_{31} l_{41} - d_2 l_{32} l_{42}\Big)/d_3; \\
\text{for } j = 4: \qquad & d_4 = a_{44} - d_1 l_{41}^2 - d_2 l_{42}^2 - d_3 l_{43}^2.
\end{aligned}$$

Hence, the algorithm for LDL^T factorization is deduced as follows.

Algorithm: Given a pdf matrix $A \in \mathbb{R}^{n\times n}$,

for $j = 1 : n$

$$d_j = a_{jj} - \sum_{k=1}^{j-1} d_k l_{jk}^2$$

$L_{jj} = 1$

for $i = j + 1 : n$

$$L_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} d_k l_{jk} l_{ik}\right)/d_j$$

end

end

In Newton's method, the computation of search direction involves the solution of

$$\nabla^2 f(\mathbf{x}^{(k)})\mathbf{s}^{(k)} = -\nabla f(\mathbf{x}^{(k)}).$$

If $\nabla^2 f(\mathbf{x}^{(k)})$ is pdf, then the factorization

$$\nabla^2 f(\mathbf{x}^{(k)}) = LDL^T$$

can be used, where the diagonal matrix D has positive diagonal entries. If $\nabla^2 f(\mathbf{x}^{(k)})$ is not pdf, then at some point during the computation of the factorization some diagonal entry of D will satisfy

$$d_{ii} \leq 0.$$

In which case, d_{ii} should be replaced by some positive entry; e.g. $|d_{ii}|$ or some small positive number.

Modifying the entries of D is equivalent to replacing $\nabla^2 f(\mathbf{x}^{(k)})$ by

$$\nabla^2 f(\mathbf{x}^{(k)}) \to \nabla^2 f(\mathbf{x}^{(k)}) + E$$

where E is a diagonal matrix and then factoring this matrix

$$\nabla^2 f(\mathbf{x}^{(k)}) + E = LDL^T$$

so the modified Hessian matrix is pdf. The factors L, D are then used to compute the search direction

$$(LDL^T)\mathbf{s}^{(k)} = -\nabla f(\mathbf{x}^{(k)}).$$

The following example shows that the processes of both checking the positive definiteness of the Hessian matrix and finding E can be carried out together.

Example ([26]) Suppose that

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} -1 & 2 & 4 \\ 2 & -3 & 6 \\ 4 & 6 & 22 \end{bmatrix}.$$

At the first stage of the factorization, $d_{11} = -1$. Clearly, the matrix is not pdf. Therefore, choose $e_{11} = 5$ so that $d_{11} = 4 > 0$. Then, it follows that $l_{11} = 1$, $l_{21} = \frac{1}{2}$, $l_{31} = 1$.

At the next stage, $d_{22} = -4$; choose $e_{22} = 12$ so that $d_{22} = 8 > 0$. Then $l_{22} = 1$ and $l_{32} = \frac{1}{2}$.

At the final stage, $d_{33} = 16$, so no modification is needed. The overall factorization is

$$\nabla^2 f(\mathbf{x}) + E = \begin{bmatrix} -1 & 2 & 4 \\ 2 & -3 & 6 \\ 4 & 6 & 22 \end{bmatrix} + \begin{bmatrix} 5 & 0 & 0 \\ 0 & 12 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

where

$$L = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ 4 & \frac{1}{2} & 1 \end{bmatrix}, \quad D = \text{diag}(4,\, 8,\, 16). \qquad \underline{\text{Ans}}$$

An alternative approach which is due to Levenberg and Marquardt ([23, 25]) is to modify Newton search direction (5.11) by giving it a bias towards the steepest descent direction $-\nabla f(\mathbf{x}^{(k)})$. This is most conveniently achieved by adding a multiple of the identity matrix to $\nabla^2 f(\mathbf{x}^{(k)})$ and solving the system

$$\left[\nabla^2 f(\mathbf{x}^{(k)}) + \beta I\right] \mathbf{s}^{(k)} = -\nabla f(\mathbf{x}^{(k)}), \quad \beta > 0. \tag{5.15}$$

From equation (5.15), one can see that for a small β, $\mathbf{s}^{(k)}$ is close to Newton direction whereas for a large β, $\mathbf{s}^{(k)}$ becomes parallel to the steepest descent direction.

5.2.3 Conjugate Directions

Definition 5.2.7. *Let Q be a real symmetric $n \times n$ matrix. The vectors $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m \in \mathbb{R}^n$ are said to be* Q-conjugate *if*

$$\mathbf{d}_i^T Q \mathbf{d}_j = 0, \quad \text{for all } i \neq j. \qquad \blacksquare$$

Clearly, when the m vectors are orthogonal, they are conjugate with respect to the identity matrix. Hence, orthogonality is a special case of Q-conjugacy.

Proposition 5.2.8. *Let $Q \in \mathbb{R}^{n \times n}$ be positive definite. If, for $k \leq n$, the vectors $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_k \in \mathbb{R}^n$ are nonzero and Q-conjugate, then they are linearly independent.* $\blacksquare$

Proof Let $\alpha_1, \alpha_2, \ldots, \alpha_k$ be scalars such that

$$\alpha_1 \mathbf{d}_1 + \alpha_2 \mathbf{d}_2 + \ldots + \alpha_k \mathbf{d}_k = \mathbf{0}. \tag{5.16}$$

Premultiplying (5.16) by $\mathbf{d}_j^T Q$ for $1 \leq j \leq k$ yields

$$\alpha_j \mathbf{d}_j^T Q \mathbf{d}_j = 0, \quad 1 \leq j \leq k.$$

This is because, by Q-conjugacy, the terms $\mathbf{d}_j^T Q \mathbf{d}_i = 0$ for $i \neq j$. Since $Q > 0$ and $\mathbf{d}_j \neq \mathbf{0}$, it follows that $\alpha_j = 0$ for $j = 1, 2, \ldots, k$. Therefore, the vectors $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_k$ are linearly independent. Q.E.D.

There is an important class of methods for minimizing a quadratic function called *conjugate direction methods.*

Definition 5.2.9. [Conjugate direction method] *Consider a method for minimizing a quadratic function* $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{p}^T\mathbf{x}$ *(*Q *is pdf) where*

$$\left.\begin{array}{rcl} \mathbf{x}^{(k+1)} & = & \mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)} \\ \lambda_k & = & \arg\min\{f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)}) : \lambda \in \mathbb{R}\} \end{array}\right\}, \quad k = 0, 1, 2, \ldots. \tag{5.17}$$

The method is called a conjugate direction method if all the search directions $\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \ldots$ *are conjugate to* Q*.* ■

Theorem 5.2.10. *For any starting point* $\mathbf{x}^{(0)}$*, the conjugate direction method (5.17) converges to the minimum of* $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{p}^T\mathbf{x}$ *in* n *steps; that is to say, the method terminates with* $\mathbf{x}^{(n)} = \mathbf{x}^* = -Q^{-1}\mathbf{p}$*.* ■

Proof First note that the vectors $\mathbf{s}^{(k)}$ are linearly independent. Thus, by the conjugacy of the vectors $\mathbf{s}^{(k)}$, an arbitrary vector $\mathbf{v}$ can be written in the form

$$\mathbf{v} = \sum_{k=0}^{n-1} \alpha_k \mathbf{s}^{(k)}, \quad \text{where } \alpha_k = \frac{\mathbf{s}^{(k)T} Q\mathbf{v}}{\mathbf{s}^{(k)T} Q\mathbf{s}^{(k)}}. \tag{5.18}$$

Then it is easy to verify that

$$\mathbf{x}^{(n)} = \mathbf{x}^{(0)} + \sum_{k=0}^{n-1} \lambda_k \mathbf{s}^{(k)}.$$

Since λ_k is obtained by minimizing $f(\mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)})$ in the direction $\mathbf{s}^{(k)}$, it follows that

$$\mathbf{s}^{(k)T} \nabla f(\mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}) = 0 \implies \lambda_k = \frac{-\mathbf{s}^{(k)T} \nabla f(\mathbf{x}^{(k)})}{\mathbf{s}^{(k)T} Q\mathbf{s}^{(k)}}.$$

Furthermore, by Q-conjugacy,

$$\begin{array}{rcl} \mathbf{s}^{(k)T} \nabla f(\mathbf{x}^{(k)}) & = & \mathbf{s}^{(k)T} \left\{ Q\mathbf{x}^{(k)} + \mathbf{p} \right\} \\ & = & \mathbf{s}^{(k)T} \left\{ Q \left[\mathbf{x}^{(0)} + \sum_{i=0}^{k-1} \lambda_i \mathbf{s}^{(i)} \right] + \mathbf{p} \right\} \\ & = & \mathbf{s}^{(k)T} \left\{ Q\mathbf{x}^{(0)} + \mathbf{p} \right\}. \end{array}$$

Hence,

$$\mathbf{x}^{(n)} = \mathbf{x}^{(0)} - \sum_{k=0}^{n-1} \frac{\mathbf{s}^{(k)T} (Q\mathbf{x}^{(0)} + \mathbf{p}) \mathbf{s}^{(k)}}{\mathbf{s}^{(k)T} Q\mathbf{s}^{(k)}}. \tag{5.19}$$

Using (5.18), one can verify that

$$\sum_{k=0}^{n-1} \frac{\mathbf{s}^{(k)T} Q\mathbf{x}^{(0)} \mathbf{s}^{(k)}}{\mathbf{s}^{(k)T} Q\mathbf{s}^{(k)}} = \mathbf{x}^{(0)} \tag{5.20}$$

and

$$\sum_{k=0}^{n-1} \frac{\mathbf{s}^{(k)T}\mathbf{p}\mathbf{s}^{(k)}}{\mathbf{s}^{(k)T}Q\mathbf{s}^{(k)}} = \sum_{k=0}^{n-1} \frac{\mathbf{s}^{(k)T}QQ^{-1}\mathbf{p}\mathbf{s}^{(k)}}{\mathbf{s}^{(k)T}Q\mathbf{s}^{(k)}} = Q^{-1}\mathbf{p} \tag{5.21}$$

From (5.19)–(5.21), it readily follows that

$$\mathbf{x}^{(n)} = \mathbf{x}^{(0)} - \mathbf{x}^{(0)} - Q^{-1}\mathbf{p} = -Q^{-1}\mathbf{p}.$$

Hence, the theorem is proved. Q.E.D.

5.2.4 Quasi-Newton (Variable Metric) Methods

There are many different quasi-Newton methods but they are all based on approximating the Hessian $\nabla^2 f(\mathbf{x}^{(k)})$ by another matrix B_k which is available at low cost. Then the search direction is obtained by solving

$$B_k\mathbf{s}^{(k)} = -\nabla f(\mathbf{x}^{(k)}).$$

There are several advantages to this approach.

- The approximant B_k can be found using only first derivatives.
- The vector $\mathbf{s}^{(k)}$ can be computed using $\mathcal{O}(n^2)$ operations.

There are also disadvantages but they are minor. The methods do not converge quadratically but they can converge superlinearly. In practice, there is not much difference between these two rates of convergence.

Quasi-Newton methods are generalizations of the secant method. For the case of a single variable, the idea is explained as follows. The secant method uses the approximation

$$f''(x^{(k)}) \approx \frac{f'(x^{(k)}) - f'(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}$$

in Newton's formula for minimization

$$x^{(k+1)} = x^{(k)} - f'(x^{(k)})/f''(x^{(k)}).$$

This leads to

$$x^{(k+1)} = x^{(k)} - \frac{x^{(k)} - x^{(k-1)}}{f'(x^{(k)}) - f'(x^{(k-1)})} f'(x^{(k)}).$$

In the multidimensional case, the approximation can be written as

$$\nabla^2 f(\mathbf{x}^{(k)})\left[\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right] \approx \nabla f(\mathbf{x}^{(k)}) - \nabla f(\mathbf{x}^{(k-1)}).$$

From this, the condition used to define the quasi-Newton approximation B_k is as follows:

$$B_k\Big[\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\Big] = \nabla f(\mathbf{x}^{(k)}) - \nabla f(\mathbf{x}^{(k-1)}). \tag{5.22}$$

This is known as the *secant condition*.

For an n-dimensional problem, this condition must be satisfied by B_k. However, the matrix B_k has n^2 entries (or $n(n+1)/2$ for a symmetric B_k), so this condition by itself is not sufficient to define B_k uniquely. Additional conditions must be imposed to specify a specific quasi-Newton method.

For the sake of convenience, the following notations are defined.

$$\left.\begin{array}{ll} \mathbf{g}^{(k)} \triangleq \nabla f(\mathbf{x}^{(k)}), & \Delta\mathbf{g}^{(k)} \triangleq \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}, \\ \Delta\mathbf{x}^{(k)} \triangleq \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}, & B_k^{-1} \triangleq H_k \end{array}\right\}. \tag{5.23}$$

From the above, the secant condition (5.22) can be rewritten as

$$B_{k+1}\Delta\mathbf{x}^{(k)} = \Delta\mathbf{g}^{(k)} \quad \text{or} \quad H_{k+1}\Delta\mathbf{g}^{(k)} = \Delta\mathbf{x}^{(k)}. \tag{5.24}$$

Quasi-Newton methods have the update formula of the form

$$B_{k+1} = B_k + \Big[\text{ something }\Big]. \tag{5.25}$$

For example, the *symmetric rank-1* update formula is

$$B_{k+1} = B_k + \frac{(\Delta\mathbf{g}^{(k)} - B_k\Delta\mathbf{x}^{(k)})(\Delta\mathbf{g}^{(k)} - B_k\Delta\mathbf{x}^{(k)})^T}{(\Delta\mathbf{g}^{(k)} - B_k\Delta\mathbf{x}^{(k)})^T\Delta\mathbf{x}^{(k)}}. \tag{5.26}$$

One can verify that (5.26) satisfies (5.24) and, furthermore, it preserves symmetry.

The algorithmic model of quasi-Newton methods is as follows:

Quasi-Newton model: A function $f : \mathbb{R}^n \to \mathbb{R}$, a starting point $\mathbf{x}^{(0)} \in \mathbb{R}^n$ and a positive definite matrix $B_0 \in \mathbb{R}^{n\times n}$ are given. Let $k = 0$.

1. If the termination criterion is satisfied, then stop; otherwise compute $\mathbf{s}^{(k)}$ using
$$B_k\mathbf{s}^{(k)} = -\mathbf{g}^{(k)} \quad \text{or} \quad \mathbf{s}^{(k)} = -H_k\mathbf{g}^{(k)}.$$
2. Compute λ_k such that $f_{k+1} < f_k$; or ideally compute
$$\lambda_k = \arg\min\{f(\mathbf{x}^{(k)} + \lambda\mathbf{s}^{(k)}) : \lambda \geq 0\}.$$
3. Update $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k\mathbf{s}^{(k)}$ and $B_{k+1} = B_k + \Big[\text{ something }\Big]$.
4. Set $k = k + 1$ and goto Step 1.

Better-known quasi-Newton methods are DFP (Davidon-Fletcher-Powell) method, whose formula is given by

$$H_{k+1} = H_k + \frac{\Delta \mathbf{x}^{(k)} \Delta \mathbf{x}^{(k)T}}{\Delta \mathbf{x}^{(k)T} \Delta \mathbf{g}^{(k)}} - \frac{H_k \Delta \mathbf{g}^{(k)} \Delta \mathbf{g}^{(k)T} H_k}{\Delta \mathbf{g}^{(k)T} H_k \Delta \mathbf{g}^{(k)}}, \tag{5.27}$$

and BFGS (Broyden-Fletcher-Goldfarb-Shanno) method, whose formula is given by

$$B_{k+1} = B_k + \frac{\Delta \mathbf{g}^{(k)} \Delta \mathbf{g}^{(k)T}}{\Delta \mathbf{g}^{(k)T} \Delta \mathbf{x}^{(k)}} - \frac{B_k \Delta \mathbf{x}^{(k)} \Delta \mathbf{x}^{(k)T} B_k}{\Delta \mathbf{x}^{(k)T} B_k \Delta \mathbf{x}^{(k)}}. \tag{5.28}$$

To compute the inverse matrix of H_{k+1} in (5.27) or B_{k+1} in (5.28) analytically, the following lemma is useful.

Lemma 5.2.11. [Sherman-Morrison] *Consider a nonsingular matrix $A \in \mathbb{R}^{n \times n}$ and the vectors $\mathbf{u}$, $\mathbf{v} \in \mathbb{R}^n$ such that $1 + \mathbf{v}^T A^{-1} \mathbf{u} \neq 0$. Then it follows that the matrix $A + \mathbf{u}\mathbf{v}^T$ is nonsingular and*

$$\left(A + \mathbf{u}\mathbf{v}^T\right)^{-1} = A^{-1} - \frac{\left(A^{-1}\mathbf{u}\right)\left(\mathbf{v}^T A^{-1}\right)}{1 + \mathbf{v}^T A^{-1} \mathbf{u}}.$$

■

Quasi-Newton methods generate $\{B_k : B_0 \text{ is given}\}$ such that B_k is an approximant to $\nabla^2 f(\mathbf{x}^{(k)})$. The methods that maintain the positive definiteness of the matrix B_k (or H_k) are sometimes called *variable metric* methods.

In general, a rank-1 formula has the form

$$B_{k+1} = B_k + a\mathbf{u}\mathbf{u}^T$$

while a rank-2 formula takes the form

$$B_{k+1} = B_k + a\mathbf{u}\mathbf{u}^T + b\mathbf{v}\mathbf{v}^T,$$

where $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ and $a, b \in \mathbb{R}$ are to be chosen. It is of interest to note that a rank-1 update formula can preserve the symmetry of the matrix B_k but cannot preserve the positive definiteness; on the other hand, a rank-2 formula can preserve the positive definiteness of B_k.

DFP Method The method was originally developed by Davidon [10] and later was modified by Fletcher and Powell [15]. Hence, the name DFP method was given.

Consider a rank-2 update formula

$$H_{k+1} = H_k + a\mathbf{u}\mathbf{u}^T + b\mathbf{v}\mathbf{v}^T.$$

From the secant (or quasi-Newton) condition, it follows that

$$\Delta \mathbf{x}^{(k)} = H_k \Delta \mathbf{g}^{(k)} + a\mathbf{u}\mathbf{u}^T \Delta \mathbf{g}^{(k)} + b\mathbf{v}\mathbf{v}^T \Delta \mathbf{g}^{(k)}. \tag{5.29}$$

An obvious choice in (5.29) is to try

$$\mathbf{u} = \Delta \mathbf{x}^{(k)} \quad \text{and} \quad \mathbf{v} = H_k \Delta \mathbf{g}^{(k)}. \tag{5.30}$$

From (5.29) and (5.30), it follows that

$$a\mathbf{u}^T \Delta \mathbf{g}^{(k)} = 1 \quad \text{and} \quad b\mathbf{v}^T \Delta \mathbf{g}^{(k)} = -1,$$

which are used to determine a and b. Hence, the DFP formula is obtained.

$$H_{k+1} = H_k + \frac{\Delta \mathbf{x}^{(k)} \Delta \mathbf{x}^{(k)T}}{\Delta \mathbf{x}^{(k)T} \Delta \mathbf{g}^{(k)}} - \frac{H_k \Delta \mathbf{g}^{(k)} \Delta \mathbf{g}^{(k)T} H_k}{\Delta \mathbf{g}^{(k)T} H_k \Delta \mathbf{g}^{(k)}}.$$

Using Sherman-Morrison Lemma, one can verify that

$$\begin{aligned} B_{k+1}^{DFP} &= B_k^{DFP} + \left(1 + \frac{\Delta \mathbf{x}^{(k)T} B_k \Delta \mathbf{x}^{(k)}}{\Delta \mathbf{g}^{(k)T} \Delta \mathbf{x}_k^{(k)}}\right) \frac{\Delta \mathbf{g}^{(k)} \Delta \mathbf{g}^{(k)T}}{\Delta \mathbf{g}^{(k)T} \Delta \mathbf{x}^{(k)}} \\ &\quad - \left(\frac{\Delta \mathbf{g}^{(k)} \Delta \mathbf{x}^{(k)T} B_k + B_k \Delta \mathbf{x}^{(k)} \Delta \mathbf{g}^{(k)T}}{\Delta \mathbf{g}^{(k)T} \Delta \mathbf{x}^{(k)}}\right) \end{aligned}$$

The DFP method has a useful property when an exact line search is used. The following theorem was given in [15].

Theorem 5.2.12. *For the DFP method (with an exact line search), if the matrix H_k is positive definite, then so is H_{k+1}.* ■

Proof Consider the quantity $\mathbf{x}^T H_{k+1} \mathbf{x}$ for $\mathbf{x} \neq \mathbf{0}$.

$$\mathbf{x}^T H_{k+1} \mathbf{x} = \mathbf{x}^T H_k \mathbf{x} + \frac{(\mathbf{x}^T \Delta \mathbf{x}^{(k)})^2}{\Delta \mathbf{x}^{(k)T} \Delta \mathbf{g}^{(k)}} - \frac{(\mathbf{x}^T H_k \Delta \mathbf{g}^{(k)})^2}{\Delta \mathbf{g}^{(k)T} H_k \Delta \mathbf{g}^{(k)}} \tag{5.31}$$

Since H_k is positive definite, one can easily verify that

$$H_k = H^{1/2} H^{1/2} \quad (\text{recall } H_k = LDL^T)$$

where $H^{1/2}$ is the square root of H_k. Therefore, define

$$\mathbf{a} \triangleq H^{1/2} \mathbf{x}, \quad \mathbf{b} \triangleq H^{1/2} \Delta \mathbf{g}^{(k)}$$

Substituting $\mathbf{a}$ and $\mathbf{b}$ into equation (5.31) yields

$$\mathbf{x}^T H_{k+1} \mathbf{x} = \mathbf{a}^T \mathbf{a} + \frac{(\mathbf{x}^T \Delta \mathbf{x}^{(k)})^2}{\Delta \mathbf{x}^{(k)T} \Delta \mathbf{g}^{(k)}} - \frac{(\mathbf{a}^T \mathbf{b})^2}{\mathbf{b}^T \mathbf{b}}$$

Hence,

$$\mathbf{x}^T H_{k+1}\mathbf{x} = \frac{\| \mathbf{a} \|^2 \| \mathbf{b} \|^2 - (\mathbf{a}^T\mathbf{b})^2}{\mathbf{b}^T\mathbf{b}} + \frac{(\mathbf{x}^T\Delta\mathbf{x}^{(k)})^2}{\Delta\mathbf{x}^{(k)T}\Delta\mathbf{g}^{(k)}} \tag{5.32}$$

From the Cauchy-Schwarz inequality (2.7), one can easily see that the first term of the right hand side of (5.32) is nonnegative. Since H_k is positive definite, one can easily deduce that $\Delta\mathbf{x}^{(k)T}\Delta\mathbf{g}^{(k)} > 0$ and hence the second term is also nonnegative. (WHY?).

Now one can show that if $H_k > 0$, then both terms cannot be zero at the same time. (WHY?) Therefore, $\mathbf{x}^T H_{k+1}\mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$ and the result is established. Q.E.D.

By using the same technique, the above theorem can be easily extended to the following useful proposition. For more details, see, for example, [14].

Proposition 5.2.13. *The DFP formula (5.27) preserves positive definite matrices H_k if*

$$\Delta\mathbf{x}^{(k)T}\Delta\mathbf{g}^{(k)} > 0. \tag{5.33}$$

■

In fact, the condition (5.33) is also necessary for the DFP formula to preserve the positive definiteness of H_k. See [26] for the details.

It is important to note that the condition (5.33) is realistic and can always be achieved in practice. This can be seen as follows. Note that

$$\Delta\mathbf{x}^{(k)T}\Delta\mathbf{g}^{(k)} = \lambda_k\Delta\mathbf{s}^{(k)T}\Big[\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}\Big].$$

For an line exact line search, $\mathbf{s}^{(k)T}\mathbf{g}^{(k+1)} = 0$; hence (5.33) is satisfied. For an inexact line search, if

$$|\mathbf{s}^{(k)T}\mathbf{g}^{(k+1)}| \leq \eta\,|\mathbf{s}^{(k)T}\mathbf{g}^{(k)}|, \quad \text{where } 0 < \eta < 1$$

then (5.33) is satisfied.

The DFP was found to work well in practice and has been used widely. It proved to be much more efficient than the steepest descent and also somewhat more efficient than the conjugate gradient methods. Early implementation attempted to carry out fairly accurate line searches.

However, the coming of low accuracy line searches in 1970 showed the DFP formula in a less satisfactory light than other formulae which were being introduced. Currently, the DFP method is no longer preferred.

However, the method has a number of important properties as follows:

- For $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{p}^T\mathbf{x},\ \mathbf{x} \in \mathbb{R}^n$ (with exact line searches),
  1. terminates in at most n iterations with $H_n = Q^{-1}$;
  2. generate conjugate directions and, when $H_0 = I$, conjugate gradient;
- For general functions,
  1. preserve positive definite H_k matrices—hence the descent property holds;
  2. requires $3n^2 + \mathcal{O}(n)$ multiplications per iteration;
  3. superlinear order of convergence (see [5] and also [12]);
  4. global convergence for strictly convex functions with exact line searches (see [32]).

It is interesting to note that Powell [32] shows that for strictly convex functions, the DFP method with exact line searches converges globally, with superlinear convergence if $\nabla^2 f(\mathbf{x}^*)$ is positive definite.

BFGS method The method was developed independently by Broyden [4], Fletcher [13], Goldfarb [18] and Shanno[37] around 1970. The method has been found to work well in practice and, especially when implemented with inexact line searches, performs better than the DFP method.

As to the theoretical side, the above properties of the DFP method also hold for the BFGS method. Moreover, global convergence of the BFGS method with inexact line searches satisfying certain conditions (see below) has been proved by Powell [33], a result which has not yet been shown for the DFP method.

To this end, it is interesting to note the convergence property of the Broyden class, whose update formulae are given by

$$\begin{aligned} B_{k+1} &= B_k + \frac{\Delta\mathbf{g}^{(k)}\Delta\mathbf{g}^{(k)T}}{\Delta\mathbf{g}^{(k)T}\Delta\mathbf{x}^{(k)}} - \frac{B_k\Delta\mathbf{x}^{(k)}\Delta\mathbf{x}^{(k)T}B_k}{\Delta\mathbf{x}^{(k)T}B_k\Delta\mathbf{x}^{(k)}} + \phi\Big(\Delta\mathbf{x}^{(k)T}B_k\Delta\mathbf{x}^{(k)}\Big)\mathbf{u}^{(k)}\mathbf{u}^{(k)T}, \\ \mathbf{u}^{(k)} &= \frac{\Delta\mathbf{g}^{(k)}}{\Delta\mathbf{g}^{(k)T}\Delta\mathbf{x}^{(k)}} - \frac{B_k\Delta\mathbf{x}^{(k)}}{\Delta\mathbf{x}^{(k)T}B_k\Delta\mathbf{x}^{(k)}} \end{aligned} \tag{5.34}$$

where ϕ is a scalar parameter. Note that the class (5.34) contains the DFP ($\phi = 1$) and the BFGS ($\phi = 0$) methods.

The Broyden class for $0 \le \phi < 1$, which excludes the DFP method, have the following convergence property, which is a generalization of Powell's (1976) result. See [26] for details.

Theorem 5.2.14 ([6]). *Consider a function $f : \mathbb{R}^n \to \mathbb{R}$. Let $\mathbf{x}^{(0)} \in \mathbb{R}^n$ be a starting point and let $\{\mathbf{x}^{(k)}\}$ be defined by $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}$, where $\mathbf{s}^{(k)} \in \mathbb{R}^n$ and $\lambda_k \geq 0$ is a scalar. Assume that*

(i) the set $S = \{\mathbf{x} : f(\mathbf{x}) \leq f(\mathbf{x}^{(0)})\}$ is bounded;

(ii) f, ∇f and $\nabla^2 f$ are continuous for all $\mathbf{x} \in S$;

(iii) $\nabla^2 f(\mathbf{x})$ is positive definite for all $\mathbf{x}$;

(iv) the search directions $\{\mathbf{s}^{(k)}\}$ are computed using

$$B_k \mathbf{s}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$$

where $B_0 = I$ and the matrix B_k are updated using a formula from the Broyden class (5.34) with the parameter $0 \leq \phi < 1$;

(v) the step length $\{\lambda_k\}$ satisfy

$$f(\mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}) \leq f(\mathbf{x}^{(k)}) + \mu \lambda_k \mathbf{s}^{(k)T} \nabla f(\mathbf{x}^{(k)}) \tag{5.35}$$

$$\mathbf{s}^{(k)T} \nabla f(\mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}) \geq \eta \mathbf{s}^{(k)T} \nabla f(\mathbf{x}^{(k)}) \tag{5.36}$$

with $0 < \mu < \eta < 1$, and the line search algorithm uses the step length $\lambda_k = 1$ whenever possible.

Then it follows that

$$\lim_{k \to \infty} \mathbf{x}^{(k)} = \mathbf{x}^*$$

where $\mathbf{x}^*$ is the unique global minimizer of f on S, and the rate of convergence of $\mathbf{x}^{(k)}$ is superlinear. ■

Implementation with inexact line search The above theorem gives rise to implementing the BFGS method in such a way that the line search algorithm uses the step length $\lambda_k = 1$ whenever possible.

Near the solution, one would expect that a unity step length would be acceptable and lead to a superlinear rate of convergence. Note that when $\lambda_k = 1$ is accepted, one needs to compute f and ∇f only once during line search at that iteration. Similarly, Newton's method quadratically converges to the solution with $\lambda_k = 1$. (See the theorem on page 46.) This also leads to the implementation with inexact line search.

The step length $\lambda_k = 1$ is tried first and accepted if it satisfies conditions (5.35) and (5.36), where (5.35) is called *Armijo* condition and (5.36) is called *Wolfe-Powell* (or *Wolfe*) condition.

The restriction $\mu < \eta$ ensures that acceptable points exists and can be located in a finite number of steps. Note that in practice, a more stringent two-sided test on the slope

$$\left|\mathbf{s}^{(k)T}\nabla f(\mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)})\right| \leq -\eta \mathbf{s}^{(k)T}\nabla f(\mathbf{x}^{(k)}) \tag{5.37}$$

is preferred in place of (5.36). Therefore, in practice BFGS method and Newton's method are often implemented with line searches satisfying (5.35) and (5.36).

Values of $\eta = 0.9$ for a weak line search or $\eta = 0.1$ for a fairly accurate line search have both been used satisfactorily in different circumstances. See [14] for details. Smaller values of η require substantially more effort to find an acceptable with very little return, so are rarely used, except in testing hypotheses about an exact line search.

As for μ, a value of $\mu = 0.01$ (or $\mu = 10^{-4}$ according to [13]) would be typical, although this choice is not very significant because it is usually (5.37) which limits the range of acceptable points.

One way to guarantee convergence is to make additional assumptions, two on the search directions s_k and two on the step lengths λ_k. The assumptions on the step lengths λ_k are that

- it produces a sufficient decrease in the function f
- it is not too small

whereas the assumptions on the search directions $\mathbf{s}^{(k)}$ are that

- it is gradient related (i.e. $\left\| \mathbf{s}^{(k)} \right\| \geq m \left\| \nabla f(\mathbf{x}^{(k)}) \right\|$ for all k where $m > 0$ is a constant)

- it produces sufficient descent so as to prevent $\mathbf{s}^{(k)}$ being closely orthogonal to $\nabla f(\mathbf{x}^{(k)})$ (When $\mathbf{s}^{(k)}$ is closely orthogonal to $\nabla f(\mathbf{x}^{(k)})$ while still remaining a descent direction, the algorithm would make little progress towards a solution.)

The Armijo stepsize condition and the Wolfe conditions are illustrated on the next page.

Clearly, the set of step size accepted by strong Wolfe's condition is a subset of that accepted by weak Wolfe's condition. That is,

$$\begin{aligned} &\left\{\lambda \in \mathbb{R} : |\mathbf{s}^{(k)T}\nabla f(x^{(k)} + \lambda \mathbf{s}^{(k)})| \leq -\eta \mathbf{s}^{(k)T}\mathbf{g}^{(k)}\right\} \\ \subset &\left\{\lambda \in \mathbb{R} : \mathbf{s}^{(k)T}\nabla f(x^{(k)} + \lambda \mathbf{s}^{(k)}) \geq \eta \mathbf{s}^{(k)T}\mathbf{g}^{(k)}\right\}. \end{aligned}$$

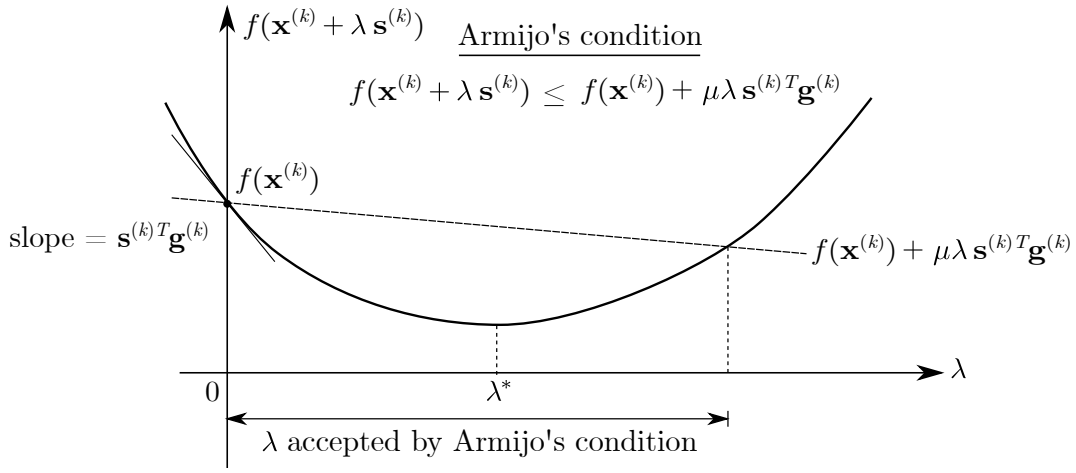

Figure 5.5: Armijo's step size rule

5.2.5 Conjugate Gradient Methods

Motivation Consider a conjugate direction method for minimizing a quadratic function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{p}^T\mathbf{x}$, where $Q > 0$ and $\mathbf{x} \in \mathbb{R}^n$, with an exact line search. Note the this minimization problem is closely related to that of solving a system of linear equations.

Given a starting point $\mathbf{x}^{(0)}$ and the vectors $\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(n-1)}$ that are Q-conjugate.

$$\left.\begin{array}{rcl} \mathbf{g}^{(k)} & = & Q\mathbf{x}^{(k)} + \mathbf{p} \\ \lambda_k & = & \dfrac{-\mathbf{g}^{(k)T}\mathbf{s}^{(k)}}{\mathbf{s}^{(k)T}Q\mathbf{s}^{(k)}} \\ \mathbf{x}^{(k+1)} & = & \mathbf{x}^{(k)} + \lambda_k\mathbf{s}^{(k)} \end{array}\right\}, \quad \text{for } k = 0, 1, 2, \ldots.$$

In this regard, an interesting question is how one can generate the search directions $\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(k)}$ that are (mutually) Q-conjugate. This is accomplished by choosing the following update formula:

$$\left.\begin{array}{rcl} \mathbf{s}^{(k+1)} & = & -\nabla f(\mathbf{x}^{(k+1)}) + \beta_k\mathbf{s}^{(k)} \\ \beta_k & = & \dfrac{\nabla f(\mathbf{x}^{(k+1)})^T Q\mathbf{s}^{(k)}}{\mathbf{s}^{(k)T}Q\mathbf{s}^{(k)}} \end{array}\right\} \tag{5.38}$$

Formula (5.38) is called a (linear) conjugate gradient method.

In proving that the formula (5.38) generates the search directions that are Q-conjugate, the following proposition is useful.

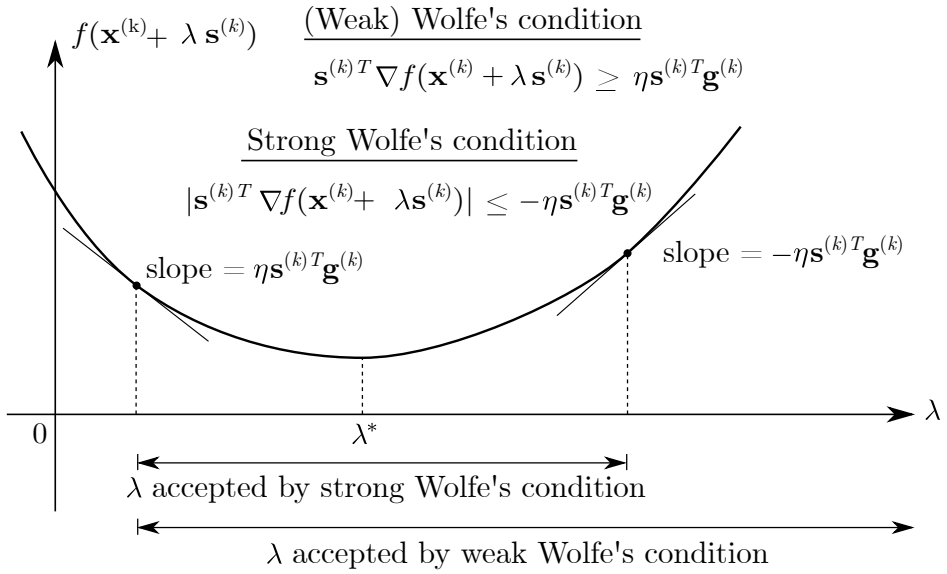

Figure 5.6: Wolfe conditions

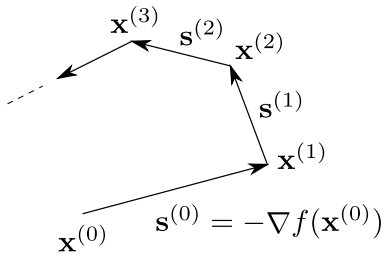

Figure 5.7: A conjugate direction method with $\mathbf{s}^{(0)} = -\nabla f(\mathbf{x}^{(0)}$

Proposition 5.2.15. *Consider a method for minimizing a quadratic function*

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{p}^T\mathbf{x}, \quad Q > 0,$$

where $\mathbf{x}^{(k)}$ *are generated by* $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}$ *($k = 0, 1, 2, \ldots$). Assume that an exact line search is used. If the vectors* $\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(k)}$ *are Q-conjugate, then*

$$\mathbf{g}^{(k+1)T}\mathbf{s}^{(i)} = 0 \quad \text{for} \;\; i = 0, 1, 2, \ldots, k$$

■

Proof First recall that since $\mathbf{g}^{(k)} = Q\mathbf{x}^{(k)} + \mathbf{p}$, we have

$$Q\Big(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\Big) = \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)};$$

and hence

$$\mathbf{g}^{(k+1)} = \mathbf{g}^{(k)} + \lambda_k Q\mathbf{s}^{(k)}. \tag{5.39}$$

Then prove the proposition by induction. For $k = 0$, it is obvious that

$$\mathbf{g}^{(1)T}\mathbf{s}^{(0)} = 0.$$

Now assume that the proposition holds for $k = j$; in other words,

$$\mathbf{g}^{(j)T}\mathbf{s}^{(i)} = 0 \quad \text{for } i = 0, 1, \ldots, j-1. \tag{5.40}$$

Then we will prove that $\mathbf{g}^{(j+1)T}\mathbf{s}^{(i)} = 0$ for $i = 0, 1, \ldots, j$. From (5.39), one can see that for $i = 0, 1, \ldots, j$,

$$\mathbf{g}^{(j+1)T}\mathbf{s}^{(i)} = \mathbf{g}^{(j)T}\mathbf{s}^{(i)} + \lambda_j \mathbf{s}^{(j)T} Q \mathbf{s}^{(i)}$$

By virtue of exact line searches, one can conclude that

$$\mathbf{g}^{(j+1)T}\mathbf{s}^{(j)} = 0$$

and from (5.40) one can deduce that

$$\mathbf{g}^{(j+1)T}\mathbf{s}^{(i)} = \mathbf{g}^{(j)T}\mathbf{s}^{(i)} + \lambda_j \mathbf{s}^{(j)T} Q \mathbf{s}^{(i)} = 0 \quad \text{for } i = 0, 1, \ldots, j-1. \tag{5.41}$$

Hence, the result is proved. Q.E.D.

We are now ready to prove the required result as follows.

Proposition 5.2.16. *Consider the minimization of a quadratic function* $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{p}^T \mathbf{x}$, $Q > 0$ *with an exact line search. The directions* $\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \ldots$ *given by the formula (5.38) are mutually* Q*-conjugate.* ■

Proof The proposition will be proved by induction and only the sketch of the proof is given as follows:

1. It is easy to prove that the proposition is true for $k = 0$.
2. Assume that the proposition is true for $k = j$; that is, assume that the directions $\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(j)}$ are mutually Q-conjugate.
3. Prove that $\mathbf{g}^{(j+1)T}\mathbf{g}^{(i)} = 0$ for $i = 0, 1, \ldots, j$.
4. Using the result from Step (3) to prove that $\mathbf{s}^{(j+1)}$ and $\mathbf{s}^{(i)}$ are Q-conjugate for $i = 0, 1, \ldots, j$. (This is equivalent to proving that $\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(j+1)}$ are mutually Q-conjugate. WHY?)
5. From Steps (1), (2) and (4), one can conclude that the proposition is true. Q.E.D.

The details of the proof is left to students as an exercise. It is important that students try to prove the above proposition themselves.

From the above proposition and the properties of conjugate direction methods, one can immediately see that the conjugate gradient methods (5.38) with an exact line search minimizes a quadratic function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{p}^T\mathbf{x},\ \mathbf{x} \in \mathbb{R}^n$ within n-steps regardless of the starting point $\mathbf{x}^{(0)}$.

Nonlinear Conjugate Gradient Methods It is clear from (5.38) that β_k is expressed in terms of Q. When applying to nonlinear objective functions, the formula (5.38) is computationally expensive. This motivates one to employ conjugate gradient methods without having to know Q explicitly.

In the following, three update formulae are derived. They are equivalent to (5.38) for the case of minimizing a quadratic function with an exact line search.

Since $\nabla f(\mathbf{x}^{(k+1)}) = Q\Big\{\mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}\Big\} + \mathbf{p}$, one can easily verify that

$$\nabla f(\mathbf{x}^{(k+1)}) = \lambda_k Q\mathbf{s}^{(k)} + \nabla f(\mathbf{x}^{(k)})$$

Hence, we have the Hestenes and Stiefel (HS) update formula:

$$\text{HS}: \quad \mathbf{s}^{(k+1)} = -\mathbf{g}^{(k+1)} + \beta_k \mathbf{s}^{(k)}, \quad \beta_k = \frac{\mathbf{g}^{(k+1)T}\Big[\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}\Big]}{\mathbf{s}^{(k)T}\Big[\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}\Big]}, \tag{5.42}$$

where $\mathbf{g}^{(k)}$ denotes $\nabla f(\mathbf{x}^{(k)})$. In fact, the formula (5.42) is originally proposed by [21] in connection with the solution of a linear system $Q\mathbf{x} = -\mathbf{p}$ by an iterative method.

Now it is ready to derive another formula. By considering the denominator of the expression for β_k in (5.42) and using the fact that $\mathbf{s}^{(k)T}\mathbf{g}^{(k+1)} = 0$, it readily follows that

$$\mathbf{s}^{(k)T}\mathbf{g}^{(k)} = -\mathbf{g}^{(k)^T}\mathbf{g}^{(k)} + \beta_{k-1}\mathbf{g}^{(k)T}\mathbf{s}^{(k-1)} = -\mathbf{g}^{(k)T}\mathbf{g}^{(k)}. \tag{5.43}$$

Using (5.43) and (5.42), one can now arrive at the **Polak-Ribiére** [30] formula:

$$\text{PR}: \quad \mathbf{s}^{(k+1)} = -\mathbf{g}^{(k+1)} + \beta_k \mathbf{s}^{(k)}, \quad \text{where } \beta_k = \frac{\mathbf{g}^{(k+1)T}\Big[\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}\Big]}{\mathbf{g}^{(k)T}\mathbf{g}^{(k)}}. \tag{5.44}$$

By starting from the Polak-Ribiére formula (5.44), one can arrive at another formula as follows. It is not difficult to verify that (using $\mathbf{g}^{(k+1)T}\mathbf{s}^{(k)} = 0$)

$$\mathbf{g}^{(k+1)T}\mathbf{g}^{(k)} = 0.$$

In other words, $\mathbf{g}^{(k+1)}$ is orthogonal to $\mathbf{g}^{(k)}$. Accordingly, the formula (5.44) becomes

$$\text{FR}: \qquad \mathbf{s}^{(k+1)} = -\mathbf{g}^{(k+1)} + \beta_k \mathbf{s}^{(k)}, \quad \text{where } \beta_k = \frac{\mathbf{g}^{(k+1)T}\mathbf{g}^{(k+1)}}{\mathbf{g}^{(k)T}\mathbf{g}^{(k)}}, \tag{5.45}$$

which is known as the **Fletcher-Reeves** [16] formula.

An algorithmic model of conjugate gradient methods for minimizing a nonlinear function $f : \mathbb{R}^n \to \mathbb{R}$ is as follows.

Conjugate Gradient Algorithm: Suppose that a starting point $\mathbf{x}^{(0)} \in \mathbb{R}^n$ are given. Let $k = 0$.

1. Compute $\mathbf{s}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$.
2. Perform the line search to obtain the step length λ_k.
3. Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}$.
4. If the termination criterion is satisfied, then terminate with $\mathbf{x}^* = \mathbf{x}^{(k)}$; else compute $\mathbf{s}^{(k+1)} = -\nabla f(\mathbf{x}^{(k+1)}) + \beta_k \mathbf{s}^{(k)}$, set $k = k + 1$ and goto step 2.

Although the HS, PR and FR update formulae for conjugate gradients methods are equivalent in the minimization of quadratic functions with exact line searches, numerous practical experiences show that the PR method seems to work somewhat better than the FR method in minimizing general nonlinear functions.

When applied to a quadratic function with an exact line search, the conjugate gradient methods (FR, PR and HS) are equivalent to the Broyden method (including DFP and BFGS methods) with exact line searches where $H_0 = B_0 = I$. See [14] for details.

Advantage of CG Methods In the implementation of CG methods, only vectors of dimension n are used. Thus, the methods require much less storage than quasi-Newton methods and Newton's method. This is a great advantage in solving very large-scale optimization problems (say, $\dim(\mathbf{x}) > 1,000$) such as in solving discrete linear optimal control problems or linear pde's.

Question How many vectors are needed in implementing the HS, PR and FR conjugate gradient methods?

Inexact Line Search Generally, conjugate gradient methods are more sensitive to accuracy in line search process than quasi-Newton methods. However, if the inexact line searches

are implemented with careful consideration, then the conjugate gradient methods can still provide satisfactory results.

In practice, step lengths λ_k are often chosen to satisfy the strong Wolfe conditions:

$$\left.\begin{array}{rcl} f(\mathbf{x}^{(k)}+\lambda\mathbf{s}^{(k)}) & \leq & f(\mathbf{x}^{(k)})+\lambda\mu\mathbf{s}^{(k)T}\nabla f(\mathbf{x}^{(k)}) \\ \left|\mathbf{s}^{(k)T}\nabla f(\mathbf{x}^{(k)}+\lambda\mathbf{s}^{(k)})\right| & \leq & -\eta\mathbf{s}^{(k)T}\nabla f(\mathbf{x}^{(k)}) \end{array}\right\}. \tag{5.46}$$

The first practical global convergence result for conjugate gradient methods is due to Al-Baali [1] and applies to the Fletcher-Reeves method. The results are as follows.

Lemma 5.2.17 ([1]). *Suppose that the Fletcher-Reeves method (5.45) is implemented with an inexact line search satisfying*

$$\left|\mathbf{s}^{(k)T}\nabla f(\mathbf{x}^{(k)}+\lambda\mathbf{s}^{(k)})\right| \leq -\eta\mathbf{s}^{(k)T}\nabla f(\mathbf{x}^{(k)}), \quad \eta\in(0,\tfrac{1}{2}].$$

It follows that the method has a descent property (that is, $\mathbf{g}^{(k)T}\mathbf{s}^{(k)}<0$) for all k. ■

Proof Recall Fletcher-Reeves CG formula:

$$\begin{array}{rcl} s_0 & = & -g_0 \\ s_{k+1} & = & -g_{k+1}+\left(\frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}\right)s_k, \quad k=1,2,3,\ldots \end{array}$$

The method of proof is to show by induction that

$$-\sum_{j=0}^{k}\eta^j \leq \frac{g_k^T s_k}{g_k^T g_k} \leq -2+\sum_{j=0}^{k}\eta^j \quad \text{for } k=0,1,2,\ldots \tag{5.47}$$

For $k=0$, we have $s_0=-g_0$. Therefore, $\frac{g_0^T s_0}{g_0^T g_0}=-1$. Clearly, (5.47) are satisfied.

Next we assume that (5.47) hold for any $k\geq 0$. It can be verified that

$$\frac{g_{k+1}^T s_{k+1}}{g_{k+1}^T g_{k+1}} = -1+\frac{g_{k+1}^T s_k}{g_k^T g_k}.$$

Since $|g_{k+1}^T s_k| \geq -\eta s_k^T g_k$ and since $s_k^T g_k<0$, it follows that

$$-1+\eta\,\frac{g_k^T s_k}{g_k^T g_k} \leq \frac{g_{k+1}^T s_{k+1}}{g_{k+1}^T g_{k+1}} \leq -1-\eta\,\frac{g_k^T s_k}{g_k^T g_k}. \tag{5.48}$$

By using (5.47) again, it follows from (5.48) that

$$-\sum_{j=0}^{k+1}\eta^j = -1-\eta\sum_{j=0}^{k}\eta^j \leq -1+\eta\,\frac{g_k^T s_k}{g_k^T g_k}.$$

$$-1-\eta\,\frac{g_k^T s_k}{g_k^T g_k} \le -1+\eta\sum_{j=0}^{k}\eta^j = -2+1+\eta\sum_{j=0}^{k}\eta^j = -2+\eta\sum_{j=0}^{k+1}\eta^j.$$

Then it follows that

$$-\sum_{j=0}^{k+1}\eta^j \le \frac{g_{k+1}^T s_{k+1}}{g_{k+1}^T g_{k+1}} \le -2+\sum_{j=0}^{k+1}\eta^j$$

The induction is thus complete. Next we note that

$$\sum_{j=0}^{k}\eta^j < \sum_{j=0}^{\infty}\eta^j = \frac{1}{1-\eta}.$$

Evidently, the right hand side of (5.47) remains negative for any $\eta \in (0, 0.5]$. Hence, the proof is complete. Q.E.D.

Lemma 5.2.17 shows that the FR conjugate gradient method preserves descent property when inexact line search is implemented with the strong Wolfe condition for $\eta < 0.5$.

The following theorem reveals the global convergence of the FR conjugate gradient method in minimizing a nonlinear function with inexact line search.

Theorem 5.2.18 ([1]). *Consider the Fletcher-Reeves method* (5.45) *for minimizing function* $f : \mathbb{R}^n \to \mathbb{R}$. *Assume that the set* $\{\mathbf{x} : f(\mathbf{x}) \le f(\mathbf{x}^{(0)})\}$ *is bounded and that* $f(\mathbf{x})$ *is twice continuously differentiable. If the step length* λ_k *(*$k = 0, 1, \ldots$*) is any value satisfying the strong Wolfe conditions* (5.46) *with* $0 < \mu < \eta < \frac{1}{2}$, *then*

$$\liminf_{k\to\infty} \| \nabla f(\mathbf{x}^{(k)}) \| = 0.$$

■

The implication of the above lemma and theorem is that the Fletcher-Reeves method with inexact line searches generates descent directions and the sequence $\{\mathbf{x}^{(k)}\}$ converging globally to a stationary point, provided the step lengths satisfy the Wolfe condition (5.46) with $0 < \mu < \eta < \frac{1}{2}$.

This result is very attractive because it applies to the algorithm as implemented in practice and because the assumptions on the function f are not restrictive. Notice that in practice the choice of $\eta \in (0, \frac{1}{2})$ is not restrictive at all.

Restart In many implementations of CG methods, the iterations (5.44) and (5.45) are restarted every n (or more) steps by setting $\beta_k = 0$; that is, taking a steepest descent direction. Alternatively, when the direction $\mathbf{s}^{(k)}$ is no longer descent or the search direction $\mathbf{s}^{(k)}$ is nearly orthogonal to $\nabla f(\mathbf{x}^{(k)})$, one may reset the direction $\mathbf{s}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$ and

continue the iteration. Restarting ensures global convergence and avoids the inefficiency that can occur in the methods.

Powell [34] demonstrates that in some cases, the Polak-Ribiére method is somehow able to reset its search direction $\mathbf{s}^{(k)}$ to the steepest descent direction whereas the Fletcher-Reeves method is not. More specifically, he considers a case where the search direction $\mathbf{s}^{(k)}$ is nearly orthogonal to the vector $-\nabla f(\mathbf{x}^{(k)})$ so that a tiny step is taken. As a consequence, $\mathbf{x}^{(k+1)} \approx \mathbf{x}^{(k)}$ and hence

$$\nabla f(\mathbf{x}^{(k)}) \approx \nabla f(\mathbf{x}^{(k+1)}).$$

The Fletcher-Reeves method gives $\beta_k \approx 1$. By using this approximation together with

$$||\nabla f(\mathbf{x}^{(k+1)})|| \approx ||\nabla f(\mathbf{x}^{(k)})|| << ||\mathbf{s}^{(k)}||,$$

we conclude that $\mathbf{s}^{(k+1)} \approx \mathbf{s}^{(k)}$, so the new search direction will improve little on the previous one. This shows the inefficient behaviour of the FR method and suggests that the method should not be implemented without some kind of restart strategy.

By contrast, the Polak-Ribiére method gives $\beta_k \approx 0$ and thereby

$$\mathbf{s}^{(k+1)} \approx -\nabla f(\mathbf{x}^{(k+1)}).$$

Obviously, the PR method essentially performs a restart after it encounters a bad direction.

Rate of Convergence Intuitively, since the conjugate gradient method accomplishes in n steps what Newton's method does in a single step, one might similarly expect from the local quadratic convergence rate of Newton's method that the n-step conjugate gradient method converges quadratically. It has been shown (Cohen [9]; see also Polak 1971) that, for a general objective function f, the rate of convergence of the conjugate gradient method with exact line searches is n-step quadratic, i.e.

$$\left\| \mathbf{x}^{(k+n)} - \mathbf{x}^* \right\| \leq C \left\| \mathbf{x}^{(k)} - \mathbf{x}^* \right\|^2$$

if the sequence $\mathbf{x}^{(k)}$ converges to $\mathbf{x}^*$ and if the function f is three times continuously differentiable in some neighbourhood of $\mathbf{x}^*$ with $\nabla^2 f(\mathbf{x}^*) > 0$.

An algorithm for finding a step length λ_k which satisfies both Armijo's and strong Wolfe's conditions is given below.

Algorithm for finding λ_k satisfying Armijo's and strong Wolfe's conditions:
Let $\mathbf{x}^{(k)} \in \mathbb{R}^n$ and $\mathbf{s}^{(k)} \in \mathbb{R}^n$ be given. Let $0 < \mu < \eta < \eta_{\max}$, $0 < \beta < 1$ and $\alpha > 1$ be specified. Define $\mathbf{g}^{(k)} = \nabla f(\mathbf{x}^{(k)})$.

1. Set $\lambda = \lambda_0$.
2. If $f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)}) \leq f(\mathbf{x}^{(k)}) + \mu \mathbf{s}^{(k)T} \mathbf{g}^{(k)} \lambda$, then goto step 5.
3. Set $\lambda = \beta \lambda$.
4. If $f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)}) > f(\mathbf{x}^{(k)}) + \mu\, \mathbf{s}^{(k)T} \mathbf{g}^{(k)} \lambda$, then goto step 3.
5. If $\left| \mathbf{s}^{(k)T} \nabla f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)}) \right| \leq -\eta\, \mathbf{s}^{(k)T} \mathbf{g}^{(k)}$, then set $\lambda_k = \lambda$ and goto step 9.
6. If $\mathbf{s}^{(k)T} \nabla f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)}) > 0$, then goto step 8.
7. Set $\lambda = \alpha \lambda$ and goto step 5.
8. By using golden section method with the intial interval $[0, \lambda]$, determine λ_k satisfying $\left| \mathbf{s}^{(k)T} \nabla f(\mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}) \right| \leq -\eta\, \mathbf{s}^{(k)T} \mathbf{g}^{(k)}$.
9. Exit.

Remarks:

- For the BFGS method, $\lambda_0 = 1$ and $\eta_{max} = 1$.
- For the Fletcher–Reeves conjugate gradient method, $\eta_{\max} = 0.5$.
- For relatively small η (say, $\eta < 0.1$), the obtained λ_k is close to the optimal step-length λ_k^*.

5.3 Exercises

5.1 Let $f : \mathbb{R}^n \to \mathbb{R}$ be given by $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} - \mathbf{x}^T\mathbf{b}$ where $\mathbf{b} \in \mathbb{R}^n$ and Q is a real symmetric positive definite $n \times n$ matrix. Consider the algorithm

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \beta\alpha_k \mathbf{g}^{(k)},$$

where $\mathbf{g}^{(k)} = Q\mathbf{x}^{(k)} - \mathbf{b}$, $\alpha_k = \mathbf{g}^{(k)T}\mathbf{g}^{(k)}/\mathbf{g}^{(k)T}Q\mathbf{g}^{(k)}$, and $\beta \in \mathbb{R}$ is a given constant. (Note that the above reduces to the steepest descent algorithm if $\beta = 1$.) Show that $\{\mathbf{x}^{(k)}\}$ converges to $\mathbf{x}^* = Q^{-1}\mathbf{b}$ for any initial condition $\mathbf{x}^{(0)}$ if and only if $0 < \beta < 2$.

5.2 Consider the problem

$$\text{minimize} \quad f(\mathbf{x}) = x_1^2 + 2x_2^2$$

where $\mathbf{x} = [x_1, x_2]^T$.

(a) If the starting point is $\mathbf{x}^{(0)} = [2, 1]^T$, use mathematical induction to show that the sequence of points generated by the steepest-descent algorithm with an exact line search is given by

$$\mathbf{x}^{(k)} = \left(\frac{1}{3}\right)^k \begin{bmatrix} 2 \\ (-1)^k \end{bmatrix}.$$

(b) Show that $f(\mathbf{x}^{(k+1)}) = f(\mathbf{x}^{(k)})/9$.

(c) Compare the results in (b) to the bounds on the convergence rate of the steepest descent method when minimizing a quadratic function. What conclusions can you draw regarding this method?

5.3 Consider the modified Newton's algorithm

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k F(\mathbf{x}^{(k)})^{-1}\mathbf{g}^{(k)}$$

where

$$\alpha_k = \arg\min_{\alpha \geq 0} f(\mathbf{x}^{(k)} - \alpha_k F(\mathbf{x}^{(k)})^{-1}\mathbf{g}^{(k)}).$$

Suppose that we apply the algorithm to a quadratic function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} - \mathbf{x}^T\mathbf{b}$, where $Q = Q^T > 0$. Recall that the standard Newton's method reaches the point $\mathbf{x}^*$ such that $\nabla f(\mathbf{x}^*) = \mathbf{0}$ in just one step starting from any initial point $\mathbf{x}^{(0)}$. Does the above modified Newton's algorithm possess the same property? Justify your answer.

5.4 A vector $\mathbf{d}$ is a *direction of negative curvature* for the function f at the point $\mathbf{x}$ if $\mathbf{d}^T\nabla^2 f(\mathbf{x})\mathbf{d} < 0$. Prove that such a direction exists if and only if at least one of the eigenvalues of $\nabla^2 f(\mathbf{x})$ is negative. Also prove that, if a direction of negative curvature

exists, then there also exists a direction of negative curvature that is also a descent direction.

5.5 Let $Q \in \mathbb{R}^{n\times n}$ be a positive definite symmetric matrix. Given an arbitrary set of linearly independent vectors $\{\mathbf{p}^{(0)}, \ldots, \mathbf{p}^{(n-1)}\}$ in $\mathbb{R}^n$, the *Gram-Schmidt* procedure generates a set of vectors $\{\mathbf{d}^{(0)}, \ldots, \mathbf{d}^{(n-1)}\}$ as follows:

$$\begin{aligned} \mathbf{d}^{(0)} &= \mathbf{p}^{(0)} \\ \mathbf{d}^{(k+1)} &= \mathbf{p}^{(k+1)} - \sum_{i=0}^{k} \frac{\mathbf{p}^{(k+1)T} Q \mathbf{d}^{(i)}}{\mathbf{d}^{(i)T} Q \mathbf{d}^{(i)}} \mathbf{d}^{(i)}, \quad k = 0, 1, 2, \ldots \end{aligned}$$

Show that the vectors $\mathbf{d}^{(0)}, \ldots, \mathbf{d}^{(n-1)}$ are Q-conjugate.

5.6 Let Q be an $n \times n$ real symmetric matrix.

(a) Show that there exists a Q-conjugate set $\{\mathbf{d}^{(1)}, \ldots, \mathbf{d}^{(n)}\}$ such that each $\mathbf{d}^{(i)}$ $(i = 1, 2, \ldots, n)$ is an eigenvector of Q.

(b) Suppose that Q is positive definite. Show that if $\{\mathbf{d}^{(1)}, \ldots, \mathbf{d}^{(n)}\}$ is a Q-conjugate set that is also orthogonal (i.e., $\mathbf{d}^{(i)T}\mathbf{d}^{(j)} = 0$ for all $i, j = 1, 2, \ldots, n$ where $i \neq j$) and if $\mathbf{d}^{(i)} \neq 0, i = 1, 2, \ldots, n$, then each $\mathbf{d}^{(i)}$ $(i = 1, 2, \ldots, n)$ is an eigenvector of Q.

5.7 Consider a quadratic function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{p}^T\mathbf{x}$, where Q is positive definite. Prove that when DFP method with an exact line search is used to minimize f, it follows that

$$H_{k+1}\Delta g^{(i)} = \Delta x^{(i)} \quad \text{for } i = 0, 1, \ldots, k.$$

Note that the above result is used to prove that the DFP method with an exact line search generates the search directions $\{\mathbf{s}^{(k)} : k = 0, 1, 2, \ldots, n-1\}$ that are Q-conjugate.

5.8 Proof that in minimizing a quadratic function

$$f(x) = \frac{1}{2}x^T Q x + p^T x \quad (Q \text{ is pdf and } Q \in \mathbb{R}^{n\times n}),$$

the DFP method with an exact line search is a conjugate direction method. That is to say, all the generated search vector $s_0, s_1, \ldots, s_{n-1}$ are Q-conjugate.

5.9 Proof that when Fletcher-Reeves conjugate gradient is used to minimize a quadratic function $f(\mathbf{x})$ given by

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{p}^T\mathbf{x} \quad (Q > 0 \text{ and } Q \in \mathbb{R}^{n\times n}),$$

with an exact line search, all the search directions $\mathbf{s}_i, i = 0, 1, \ldots, k$ $(k < n)$ are Q-conjugate. Hence, the method converges to the minimum of f within n steps for any starting point.

5.10 Proof that when minimizing $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{p}^T\mathbf{x}$ (Q is positive definite) with an exact line search, the conjugate gradient method whose search vector is given by

$$\mathbf{s}_k = \begin{cases} -\mathbf{g}_k + \left(\dfrac{\mathbf{g}_k^T Q\mathbf{s}_{k-1}}{\mathbf{s}_{k-1}^T Q\mathbf{s}_{k-1}}\right)\mathbf{s}_{k-1}, & k > 0 \\ -\mathbf{g}_0, & k = 0 \end{cases}$$

is a conjugate direction method.

5.11 Consider a conjugate gradient method whose search direction $\mathbf{s}_k$ is of the form

$$\left.\begin{array}{rcl} \mathbf{s}_{k+1} & = & -\mathbf{g}_{k+1} + \beta_k \mathbf{s}_k \\ \mathbf{s}_0 & = & -\mathbf{g}_0 \end{array}\right\}, \quad k = 0, 1, 2, \ldots$$

where the β_k is any real number satisfying

$$|\beta_k| \leq \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k}.$$

Notice that this is not Fletcher-Reeves method but a family of conjugate gradient methods that includes Fletcher-Reeves method. Suppose that, for each k, the method chooses a step length α_k satisfying the strong Wolfe condition:

$$|\mathbf{g}_{k+1}^T \mathbf{s}_k| \leq -\eta \mathbf{g}_k^T \mathbf{s}_k.$$

Prove that the method generates descent directions $\mathbf{s}_k$ that satisfies

$$-\frac{1}{1-\eta} \leq \frac{\mathbf{g}_k^T \mathbf{s}_k}{||\mathbf{g}_k||^2} \leq \frac{2\eta - 1}{1-\eta}$$

for all k. What is the maximum value of η which ensures that the method always generates search directions with descent preperty.

5.12 Use the Fletcher–Reeves conjugate gradient method with exact line search to minimize the following objective function.

$$f(x) = 5x_1^2 + x_1 x_2 + x_1 - x_2 + \frac{5}{2}x_2^2.$$

Use exact arithmetic (without rounding). Explain all the formulae that you use in the computation. You must provide the result for every iteration.

5.13 Use the DFP method with exact line search to minimize the following objective function.

$$f(x) = 5x_1^2 + x_1x_2 + x_1 - x_2 + \frac{5}{2}x_2^2.$$

Use exact arithmetic (without rounding). Explain all the formulae that you use in the computation. You must provide the result for every iteration.

Chapter 6

Optimality Conditions for Constrained Optimization

Here, we pay attention to constrained optimization problems of the form

$$\left.\begin{array}{lll} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & f(\mathbf{x}) & \\ \text{subject to} & h_i(\mathbf{x}) = \mathbf{0}, & i = 1, 2, \ldots, m \\ & g_j(\mathbf{x}) \leq \mathbf{0}, & j = 1, 2, \ldots, p \end{array}\right\}. \tag{6.1}$$

Problem (6.1) can be written as

$$\min\ \{f(\mathbf{x}) :\ \mathbf{x} \in \Omega \subset \mathbb{R}^n\},$$

where Ω is called a *feasible set* and defined as

$$\Omega \triangleq \{\mathbf{x} \in \mathbb{R}^n : h_i(\mathbf{x}) = 0,\ 1 \leq i \leq m;\ g_j(\mathbf{x}) \leq 0,\ 1 \leq j \leq p\}.$$

Example

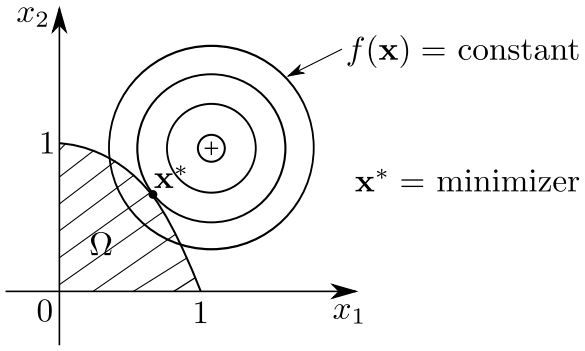

Figure 6.1: xxx

$$\begin{array}{lll} \min & f(\mathbf{x}) = (x_1 - 1)^2 + (x_2 - 1)^2 & \\ \text{subject to} & g_1(\mathbf{x}) = -x_1 \leq 0 & (x_1 \geq 0) \\ & g_2(\mathbf{x}) = -x_2 \leq 0 & (x_2 \geq 0) \\ & g_3(\mathbf{x}) = x_1^2 + x_2 - 1 \leq 0. & \end{array}$$

In vector notation, Problem (6.1) can be represented by the following standard form:

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{h}(\mathbf{x}) = \mathbf{0} \ \text{ and } \ \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \end{array}$$

where $\mathbf{h} : \mathbb{R}^n \to \mathbb{R}^m$ and $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^p$. Ans

6.1 Equality-Constrained Problems

Before considering optimization with equality and inequality constraints, we will consider the case of equality constraints:

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{h}(\mathbf{x}) = \mathbf{0} \end{array} \tag{6.2}$$

where h_i is assumed to be continuously differentiable.

Consider the following example.

$$\begin{array}{lll} \min & f(\mathbf{x}) = & -x_1\, x_2\, x_3 \\ \text{subject to} & h_1(\mathbf{x}) = & x_1 + x_2 + x_3 - 1 = 0. \end{array}$$

Eliminating x_3 with the help of $h_1(\mathbf{x}) = 0$, we get an unconstrained optimization problem:

$$\min_{(x_1,x_2)\in\mathbb{R}^2} \; x_1\, x_2\, (1 - x_1 - x_2).$$

Notice that this method is applicable as long as the equality constraints can be solved explicitly for a given set of independent variables $\mathbf{x}$.

In the presence of several equality constraints, the elimination process may become unwieldy. Moreover, in certain situation, it may not be possible to solve the constraints explicitly to eliminate a variable. For example,

$$h_1(\mathbf{x}) = x_1^2\, x_3 + x_2\, x_3^2 + \frac{x_1}{x_2} = 0.$$

In problem involving several complex equality constraints, it is better to use the method of Lagrange multipliers, which is systematic, for handling the constraints

Definition 6.1.1. *A point $\mathbf{x}^*$ satisfying the constraints $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ is said to be a* regular point *of the constraints if the vectors $\nabla h_1(\mathbf{x}^*),\ \nabla h_2(\mathbf{x}^*), \ldots, \nabla h_m(\mathbf{x}^*)$ are linearly independent.* ■

In connection with the above definition, let $\nabla\mathbf{h}(\mathbf{x})$ be the Jacobian matrix of $\mathbf{h}$, given by

$$\nabla\mathbf{h}(\mathbf{x}) \triangleq \begin{bmatrix} \nabla h_1(\mathbf{x})^T \\ \vdots \\ \nabla h_m(\mathbf{x})^T \end{bmatrix}.$$

One can now see that the point $\mathbf{x}$ is regular if and only if

$$\operatorname{rank}\Big[\nabla\mathbf{h}(\mathbf{x})\Big] = m.$$

Notice that the set of equality constraints $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ describes a surface S given by

$$S \triangleq \{\mathbf{x} \in \mathbb{R}^n : \mathbf{h}(\mathbf{x}) = \mathbf{0}\}.$$

When all the points in S are regular, the dimension of S is $n - m$.

Example Let $n = 3$ and $m = 1$. If all points in S are regular, then the set S is a two-dimensional surface. For example, let

$$h_1(\mathbf{x}) = x_2 - x_3^2 = 0.$$

Note that $\nabla h_1(\mathbf{x}) = [0,\ 1,\ -2x_3]^T$. Hence, for any $\mathbf{x} \in \mathbb{R}^3$,

$$\nabla h_1(\mathbf{x}) \neq \mathbf{0}.$$

In this case,

$$\dim S = \dim\{\mathbf{x} \in \mathbb{R}^3 : h_1(\mathbf{x}) = 0\} = n - m = 2.$$

Example In connection with the previous example, assume that $n = 3$ and $m = 2$. Let

$$h_1(\mathbf{x}) = x_1,$$

$$h_2(\mathbf{x}) = x_2 - x_3^2.$$

In this case, $\nabla h_1(\mathbf{x}) = [1,\ 0,\ 0]^T$ and $\nabla h_2(\mathbf{x}) = [0,\ 1,\ -2x_3]^T$. Hence, the vectors $\nabla h_1(\mathbf{x})$ and $\nabla h_2(\mathbf{x})$ are linearly independent in $\mathbb{R}^3$. Therefore,

$$\dim S = \dim\{\mathbf{x} \in \mathbb{R}^3 : h_1(\mathbf{x}) = 0,\ h_2(\mathbf{x}) = 0\} = 1.$$

6.1.1 Tangent plane

In the following, we consider a representation of the tangent plane at a regular point on the surface S defined by the equality constraints $\mathbf{h}(\mathbf{x}) = \mathbf{0}$.

Definition 6.1.2. *A* curve C *on a surface* S *is a set of points* $\{\mathbf{x}(t) \in S : t \in (a, b)\}$, *continuously parameterized by* $t \in (a, b)$. ■

The definition of a curve implies that all the points on the curve satisfy the equation describing the surface. The curve C passes through a point $\mathbf{x}^*$ is there exists $t^* \in (a, b)$ such that $\mathbf{x}(t^*) = \mathbf{x}^*$. Intuitively, we can think of a curve $C = \{\mathbf{x}(t) : t \in (a, b)\}$ as the path traversed by a point $\mathbf{x}$ travelling on the surface S.

At regular points, it is possible to characterize the tangent plane in terms of the gradients of the constraint functions.

Theorem 6.1.3. *At a regular point* $\mathbf{x}^*$ *of the surface* S *defined by* $\mathbf{h}(\mathbf{x}) = \mathbf{0}$*, the tangent plane is equal to*

$$M = \{\mathbf{y} : \nabla h_i(\mathbf{x}^*)^T \mathbf{y} = 0,\ i = 1, 2, \ldots, m\}.$$

■

It is important to recognize that the condition of being a regular point is not a condition on the constraint surface S itself but on its representation in terms of an $\mathbf{h}$. The tangent plane is defined independently of the representation, while M is not.

First-Order Necessary Conditions Once the representation of the tangent plane is known, it is fairly simple to derive a necessary condition for a point to be a local minimum point subject to the equality constraints.

Lemma 6.1.4. *Let* $\mathbf{x}^*$ *be a regular point of* $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ *and be a local extremum of* f *subject to these constraints. Then all* $\mathbf{y} \in \mathbb{R}^n$ *satisfying*

$$\nabla h_i(\mathbf{x}^*)^T \mathbf{y} = 0, \quad i = 1, 2, \ldots, m$$

must also satisfy

$$\nabla f(\mathbf{x}^*)^T \mathbf{y} = 0.$$

■

The lemma simply says that $\nabla f(\mathbf{x}^*)$ is orthogonal to the tangent plane. And it will be used next to conclude that $\nabla f(\mathbf{x}^*)$ is a linear combination of $\nabla h_i(\mathbf{x}^*)$ at a local minimizer $\mathbf{x}^*$.

Theorem 6.1.5. [Lagrange multiplier] *Let* $\mathbf{x}^*$ *be a local extremum point of* f *subject to the constraint* $\mathbf{h}(\mathbf{x}) = \mathbf{0}$*. Assume that* $\mathbf{x}^*$ *is a regular point of these constraints. Then there is a* $\boldsymbol{\lambda} \in \mathbb{R}^m$ *such that*

$$f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) = \mathbf{0}.$$

■

The conditions $\nabla f(\mathbf{x}^*) + \sum_{i=1}^m \lambda_i \nabla h_i(\mathbf{x}^*) = \mathbf{0}$, together with the constraints $\mathbf{h}(\mathbf{x}^*) = \mathbf{0}$, give a total of $n+m$ equations in the $n+m$ variables comprising $(\mathbf{x}^*, \boldsymbol{\lambda})$. Thus, the conditions are a complete set since they determine a unique solution.

In connection with the first-order necessary conditions, define the Lagrangian $\mathcal{L}$ associated with the constrained problem as follows.

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) \triangleq f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{x}). \tag{6.3}$$

From (6.3), one can easily see that the necessary conditions can be expressed as

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0} \quad \text{and} \quad \nabla_{\boldsymbol{\lambda}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0}. \tag{6.4}$$

The condition $\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\mathbf{x},\boldsymbol{\lambda}) = \mathbf{0}$ is simply a restatement of $\mathbf{h}(\mathbf{x}) = \mathbf{0}$.

By means of Lagrange multiplier, Problem (6.2), which is

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & f(\mathbf{x}), \\ \text{subject to} & \mathbf{h}(\mathbf{x}) = \mathbf{0}, \end{array}$$

can be transformed into an equivalent problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n,\ \boldsymbol{\lambda}\in\mathbb{R}^m} \mathcal{L}(\mathbf{x},\boldsymbol{\lambda}) \stackrel{\triangle}{=} f(\mathbf{x}) + \boldsymbol{\lambda}^T\mathbf{h}(\mathbf{x}) \tag{6.5}$$

The first order necessary conditions for optimality of (6.5) are

$$\begin{array}{lll} \dfrac{\partial\mathcal{L}}{\partial\mathbf{x}} = \mathbf{0} & \Longrightarrow & \dfrac{\partial f}{\partial\mathbf{x}} + \displaystyle\sum_{i=1}^{m}\lambda_i\frac{\partial h_i}{\partial\mathbf{x}} = \mathbf{0}\,; \\ \dfrac{\partial\mathcal{L}}{\partial\boldsymbol{\lambda}} = \mathbf{0} & \Longrightarrow & \mathbf{h}(\mathbf{x}) = \mathbf{0}. \end{array}$$

Example Find the dimensions of a cylindrical tin (with top and bottom) made up of sheet metal to maximize its volume such that the total surface area is equal to $A_0 = 24\pi$.

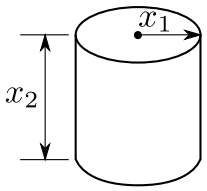

Figure 6.2: A Cylindrical tin with radius x_1 and height x_2

Solution: It is easy to obtain

$$\begin{aligned} \text{Volume} &= \pi x_1^2 x_2. \\ \text{Total area} &= 2\pi x_1^2 + 2\pi x_1 x_2 = 24\pi. \end{aligned}$$

The desired solution is determined by solving the following problem.

$$\begin{array}{ll} \max & f(x_1,x_2) = \pi x_1^2 x_2, \\ \text{subject to} & h(\mathbf{x}) = 2\pi x_1^2 + 2\pi x_1 x_2 - 24\pi = 0. \end{array}$$

Define the Lagrangian as follows.

$$\begin{aligned} \mathcal{L}(\mathbf{x},\lambda) &\stackrel{\triangle}{=} f(\mathbf{x}) + \lambda h(\mathbf{x}) \\ &= \pi x_1^2 x_2 + \lambda(2\pi x_1^2 + 2\pi x_1 x_2 - 24\pi). \end{aligned}$$

The necessary conditions for the maximum of f are

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial x_1} &= 2\pi x_1 x_2 + 4\pi\lambda x_1 + 2\pi\lambda x_2 &= 0; \\
\frac{\partial \mathcal{L}}{\partial x_2} &= \pi x_1^2 + 2\pi\lambda x_1 &= 0; \\
\frac{\partial \mathcal{L}}{\partial \lambda} &= 2\pi x_1^2 + 2\pi x_1 x_2 - 24\pi &= 0.
\end{aligned}$$

Therefore,

$$x_1^* = 2, \quad x_2^* = 4, \quad \lambda^* = -1, \quad f^* = 16\pi \qquad \underline{\text{Ans}}$$

Second-Order Conditions When considering second-order conditions, we assume as usual that f and $\mathbf{h}$ are twice continuously differentiable (that is, f, $h_i \in \mathcal{C}^2$).

Define $L(\mathbf{x}, \boldsymbol{\lambda})$ as the Hessian matrix of the Lagrangian $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$ with respect to $\mathbf{x}$; that is,

$$L(\mathbf{x}, \boldsymbol{\lambda}) \triangleq \nabla^2 f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \nabla^2 h_i(\mathbf{x}).$$

A necessary and a sufficient conditions for optimization problem (6.2) are given in the following.

Theorem 6.1.6 (necessary condition)**.** *Let $\mathbf{x}^*$ be a local minimum of f subject to $\mathbf{h}(\mathbf{x}) = \mathbf{0}$. Suppose that $\mathbf{x}^*$ is a regular point of these constraints. Then there is a $\boldsymbol{\lambda} \in \mathbb{R}^m$ such that*

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) = \mathbf{0}.$$

Furthermore, $\mathbf{y}^T L(\mathbf{x}^*, \boldsymbol{\lambda})\mathbf{y} \geq 0$ for all $\mathbf{y} \in M$, where the tangent plane $M = \{\mathbf{y} \in \mathbb{R}^n : \nabla h_i(\mathbf{x}^*)^T \mathbf{y} = 0,\ i = 1, 2, \ldots, m\}$. ■

Observe that $L(\mathbf{x}, \boldsymbol{\lambda})$ plays a similar role as the Hessian $\nabla^2 f(\mathbf{x})$ of the objective function f did in the case of unconstrained minimization. However, we now require that $L(\mathbf{x}^*, \boldsymbol{\lambda}) \geq 0$ only on the tangent subspace rather than in $\mathbb{R}^n$.

It is clear that if $L \geq 0$ in $\mathbb{R}^n$, then $L \geq 0$ in any subset of $\mathbb{R}^n$ including M. If, on the other hand, L is not psdf in $\mathbb{R}^n$, then it may or may not be psdf on M.

Example Let $M = \{\mathbf{y} \in \mathbb{R}^3 : y_1 + y_2 + y_3 = 0\}$, and let

$$L = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

One can easily find that L is indefinite. (HOW?)

On the subspace M we note that

$$\mathbf{y}^T L\mathbf{y} = y_1(y_2 + y_3) + y_2(y_1 + y_3) + y_3(y_1 + y_2) = -(y_1^2 + y_2^2 + y_3^2).$$

As a consequence, L is nsdf on M. Ans

Theorem 6.1.7 (sufficient condition)**.** *Suppose there is a point* $\mathbf{x}^*$ *satisfying* $\mathbf{h}(\mathbf{x}^*) = \mathbf{0}$ *and a* $\boldsymbol{\lambda} \in \mathbb{R}^m$ *such that*

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) = \mathbf{0}$$

Suppose also that the matrix

$$L(\mathbf{x}^*) = \nabla^2 f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla^2 h_i(\mathbf{x}^*)$$

is pdf on $M = \{\mathbf{y} : \nabla h_i(\mathbf{x}^*)^T \mathbf{y} = 0,\ i = 1, 2, \ldots, m\}$. *Then* $\mathbf{x}^*$ *is a strict local minimum of* f *subject to* $\mathbf{h}(\mathbf{x}) = \mathbf{0}$. ■

6.2 Inequality-Constrained Problems

In the previous section, we analysed constrained optimization problems involving only equality constraints. Here, we discuss extremum problems involving inequality constraints.

Consider an optimization problem having only inequality constraints given by

$$\left.\begin{array}{ll} \min & f(\mathbf{x}) \\ \text{subject to} & g_i(\mathbf{x}) \le 0, \quad i = 1, 2, \ldots, m \end{array}\right\}. \tag{6.6}$$

The inequality constraints can be transformed to equality constraints by adding nonnegative slack variables y_i^2 as follows:

$$g_i(\mathbf{x}) + y_i^2 = 0, \quad i = 1, 2, \ldots, m,$$

where the values of the slack variables are yet unknown.

Hence, Problem (6.6) becomes

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ \text{subject to} & g_i(\mathbf{x}) + y_i^2 = 0, \quad i = 1, 2, \ldots, m. \end{array}$$

Using Lagrange multiplier techniques, one can verify that the necessary conditions for the optimality are

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \mathbf{0}, \qquad \frac{\partial \mathcal{L}}{\partial \mathbf{y}} = \mathbf{0}, \qquad \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = \mathbf{0},$$

where

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \left(g_i(\mathbf{x}) + y_i^2 \right).$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \mathbf{0} \quad &\Rightarrow \quad \frac{\partial f}{\partial x_i}(\mathbf{x}) + \sum_{j=1}^{m} \lambda_j \frac{\partial g_j}{\partial x_i}(\mathbf{x}) = 0, \quad i = 1, 2, \ldots, n \\ \frac{\partial \mathcal{L}}{\partial \mathbf{y}} = \mathbf{0} \quad &\Rightarrow \quad \lambda_j \, y_j = 0, \quad j = 1, 2, \ldots, m \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = \mathbf{0} \quad &\Rightarrow \quad g_i(\mathbf{x}) + y_i^2 = 0, \quad i = 1, 2, \ldots, m \end{aligned}$$

From the above, we have the following observations at the optimum point.

- If $g_i(\mathbf{x})$ is active, then $y_i = 0$ and hence $\lambda_i \neq 0$.
- If $g_i(\mathbf{x})$ is inactive, then $y_i \neq 0$ and hence $\lambda_i = 0$.
- If $\lambda_i = 0$ and $y_i = 0 \Rightarrow g_i(\mathbf{x})$ is weakly active.

In connection with the observations, the following definition is given.

Definition 6.2.1. *An inequality constraint $g_j \leq 0$ is said to be active at a feasible point $\mathbf{x} \in \mathbb{R}^n$ if $g_j(\mathbf{x}) = 0$. It is inactive at $\mathbf{x}$ if $g_j(\mathbf{x}) < 0$. By convention, we consider an equality constraint $h_i(\mathbf{x}) = 0$ to be always active.* ■

We are now ready to consider more general optimization problems of the form

$$\begin{array}{lll} \min & f(\mathbf{x}) & \\ \text{subject to} & \mathbf{h}(\mathbf{x}) = \mathbf{0} & (\mathbf{h} : \mathbb{R}^n \to \mathbb{R}^m) \\ & \mathbf{g}(\mathbf{x}) \leq \mathbf{0} & (\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^p). \end{array} \tag{6.7}$$

Assume as before that f, $\mathbf{h}$ and $\mathbf{g}$ are smooth.

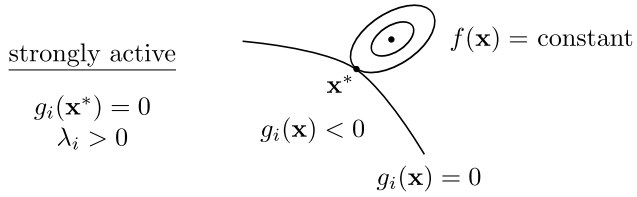

Figure 6.3: Strongly-active inequality constraint

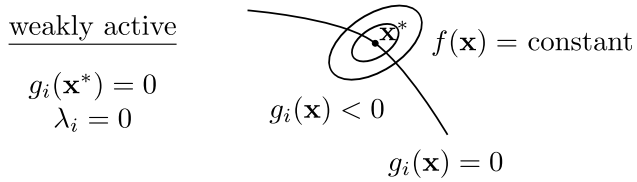

Figure 6.4: Weakly-active inequality constraint

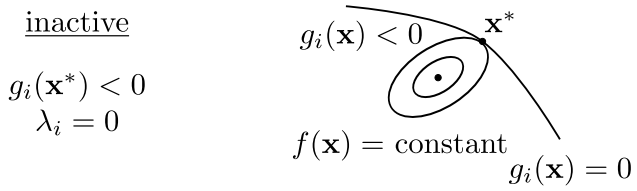

Figure 6.5: Inactive inequality constraint

Definition 6.2.2. *Let* $\mathbf{x}^*$ *be a point satisfying the constraints*

$$\mathbf{h}(\mathbf{x}^*) = \mathbf{0}, \quad \mathbf{g}(\mathbf{x}^*) \leq \mathbf{0}, \tag{6.8}$$

and let J *be the set of indices* j *for which* $g_j(\mathbf{x}^*) = 0$. *Then* $\mathbf{x}^*$ *is said to be a regular point of the constraints (6.8) if the vectors* $\nabla h_i(\mathbf{x}^*)$ $(1 \leq i \leq m)$ *and the vectors* $\nabla g_j(\mathbf{x}^*)$ $(j \in J)$ *are linearly independent.* ■

6.2.1 First-Order Necessary Condition

A first order necessary condition for optimization (6.7) are given below, which is known as Karush–Kuhn–Tacker (KKT) condition.

Theorem 6.2.3. [Karush–Kuhn–Tucker] *Let* $\mathbf{x}^*$ *be a relative minimum point for the problem*

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ \text{subject to} & \mathbf{h}(\mathbf{x}) = \mathbf{0}, \qquad \mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \end{array}$$

Suppose $\mathbf{x}^*$ *is a regular point for the constraints. Then there is a vector* $\boldsymbol{\lambda} \in \mathbb{R}^m$ *and a vector* $\boldsymbol{\mu} \in \mathbb{R}^p$ *with* $\boldsymbol{\mu} \geq \mathbf{0}$ *such that*

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) + \sum_{j \in J} \mu_j \nabla g_j(\mathbf{x}^*) = \mathbf{0}$$
$$\boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}^*) = 0.$$

■

In the above theorem, we refer $\boldsymbol{\lambda}$ as the Lagrange multiplier vector and $\boldsymbol{\mu}$ as the Karush–Kuhn–Tucker (KKT) multiplier vector. Observe that $\boldsymbol{\mu} \geq \mathbf{0}$ and $g_j(\mathbf{x}^*) \leq 0$. Therefore, the condition

$$\boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}^*) = \mu_1 g_1(\mathbf{x}^*) + \mu_2 g_2(\mathbf{x}^*) + \ldots + \mu_p g_p(\mathbf{x}^*) = 0$$

implies that if $g_j(\mathbf{x}^*) < 0$, then $\mu_j = 0$.

Let J be the index set of active inequality constraints at $\mathbf{x}^*$. From the above, it follows that

$$\mu_j = 0 \quad \text{for all } j \notin J.$$

In other words, the KKT multipliers μ_j corresponding to inactive constraints are zero.

Example Consider the problem

$$\begin{array}{ll} \min & 2x_1^2 + 2x_1x_2 + x_2^2 - 10x_1 - 10x_2 \\ \text{subject to} & x_1^2 + x_2^2 \leq 5 \\ & 3x_1 + x_2 \leq 6. \end{array}$$

The first order necessary conditions are

$$\begin{gathered} 4x_1 + 2x_2 - 10 + 2\mu_1 x_1 + 3\mu_2 = 0, \\ 2x_1 + 2x_2 - 10 + 2\mu_1 x_2 + \mu_2 = 0, \\ \mu_1 \geq 0, \quad \mu_2 \geq 0, \\ \mu_1(x_1^2 + x_2^2 - 5) = 0, \\ \mu_2(3x_1 + x_2 - 6) = 0. \end{gathered}$$

To find the solution, we assume various combinations of active constraints and then check the signs of the resulting multipliers. Here, we can try setting none, one, or two constraints active.

By assuming the 1st constraint is active and the 2nd one inactive, we have

$$\begin{gathered} 4x_1 + 2x_2 - 10 + 2\mu_1 x_1 = 0, \\ 2x_1 + 2x_2 - 10 + 2\mu_1 x_2 = 0, \\ x_1^2 + x_2^2 = 5. \end{gathered}$$

which has the solution

$$x_1 = 1, \quad x_2 = 2, \quad \mu_1 = 1.$$

This yields $3x_1 + x_2 = 5$ and hence the second constraint is satisfied. Since $\mu_1 > 0$, we conclude that the solution satisfies the KKT conditions. Ans

6.2.2 Second-Order Conditions

Let M denote the tangent subspace of the active constraints at $\mathbf{x}^*$, i.e.,

$$M \triangleq \{\mathbf{y} \in \mathbb{R}^n : \nabla h_i(\mathbf{x}^*)^T \mathbf{y} = 0 \ \ \forall i, \ \ \nabla g_j(\mathbf{x}^*)^T \mathbf{y} = 0 \ \ \forall j \in J\}.$$

Second order conditions for optimization problem (6.7) are given below.

Theorem 6.2.4. [necessary conditions] *Suppose the functions $f, g_i, h_i \in \mathcal{C}^2$ and that $\mathbf{x}^*$ is a regular point of the constraints (6.8). If $\mathbf{x}^*$ is a relative minimum point for the problem (6.7), then there is a $\boldsymbol{\lambda} \in \mathbb{R}^m$, $\boldsymbol{\mu} \in \mathbb{R}^p$, $\boldsymbol{\mu} \geq \mathbf{0}$ such that*

$$L(\mathbf{x}^*) = \nabla^2 f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla^2 h_i(\mathbf{x}^*) + \sum_{i=1}^{p} \mu_i \nabla^2 g_i(\mathbf{x}^*)$$

is psdf *on M.* ■

Notice that $L(\mathbf{x}^*)$ is the Hessian of the Lagrangian $\mathcal{L}$ which involves equality constraints and inequality constraints that are active at $\mathbf{x}^*$ where

$$\mathcal{L}(\mathbf{x}^*) = f(\mathbf{x}^*) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x}^*) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}^*).$$

Theorem 6.2.5. [sufficient conditions] *Let $f, g_i, h_i \in \mathcal{C}^2$. A point $\mathbf{x}^* \in \mathbb{R}^n$ satisfying (6.8) is a strict local minimum point of problem (6.7) if there exist $\boldsymbol{\lambda} \in \mathbb{R}^m$, $\boldsymbol{\mu} \in \mathbb{R}^p$, such that*

$$\boldsymbol{\mu} \geq \mathbf{0}, \quad \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}^*) = 0,$$

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) + \sum_{i=1}^{p} \mu_i \nabla g_i(\mathbf{x}^*) = \mathbf{0},$$

and the Hessian matrix

$$L(\mathbf{x}^*) = \nabla^2 f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla^2 h_i(\mathbf{x}^*) + \sum_{i=1}^{p} \mu_i \nabla^2 g_i(\mathbf{x}^*)$$

is pdf on the subspace M. ■

6.3 Convex Programs

In practice, many optimizations called convex programs can be found. A convex optimization problem is defined as

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in \Omega \end{array}$$

where Ω is a convex set and f is convex function on Ω.

An example of convex programs is

$$\begin{array}{lll} \min & f(\mathbf{x}) & \\ \text{subject to} & g_i(\mathbf{x}) \geq 0, & i = 1, 2, \ldots . m \end{array}$$

where f is convex and g_i is concave.

Proposition 6.3.1. *f is concave if, and only if, $-f$ is convex.* ■

Theorem 6.3.2. [global solutions of convex programs] *Let $\mathbf{x}^*$ be a local minimizer of a convex programming problem. Then $\mathbf{x}^*$ is also a global minimizer. If the objective function is strictly convex, then $\mathbf{x}^*$ is the unique global minimizer.* ■

Proof See [26], Chapter 2, p. 22. Q.E.D.

Proposition 6.3.3. *A feasible set*

$$\Omega \stackrel{\triangle}{=} \{\mathbf{x} : A\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \in \mathbb{R}^n,\ A \in \mathbb{R}^{m \times n}\}$$

is a convex set. ■

Proposition 6.3.4. *Let $g_1, g_2, \ldots, g_m$ be concave functions on $\mathbb{R}^n$. Then the set $\Omega \stackrel{\triangle}{=} \{\mathbf{x} : g_i(\mathbf{x}) \geq 0,\ i = 1, 2, \ldots, m\}$ is convex.* ■

Exercise Prove Propositions 6.3.3 and 6.3.4.

6.4 Exercises

6.1 Solve the following optimization problems using the second-order sufficient conditions:

$$\begin{array}{rrcl} \min & x_1^2 + x_2^2 & & \\ \text{subject to} & 4 - x_1 - x_2^2 & \leq & 0 \\ & 3x_2 - x_1 & \leq & 0 \\ & -3x_2 - x_1 & \leq & 0. \end{array}$$

6.2 Consider the linear program

$$\begin{array}{rl} \min & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} \geq \mathbf{b}. \end{array}$$

(a) Write the first- and second-order necessary conditions for a local solution.

(b) Show that the second-order sufficient conditions do not hold anywhere, but that any point $\mathbf{x}^*$ satisfying the first-order necessary conditions is a global minimizer.

6.3 Let Q be an $n \times n$ symmetric matrix.

(a) Compute all stationary points of the problem

$$\begin{array}{rl} \max & \mathbf{x}^T Q\mathbf{x} \\ \text{subject to} & \mathbf{x}^T\mathbf{x} = 1. \end{array}$$

(b) Determine which of the stationary points are global maximizers.

(c) How do your results in part (a) change if the constraint is replaced by $\mathbf{x}^T A\mathbf{x} \leq 1$ where A is positive definite.

6.4 Consider the problem

$$\begin{array}{rl} \min & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} \leq \mathbf{0}. \end{array}$$

where $A \in \mathbb{R}^{m\times n}$ is of full rank and $m < n$. Use the KKT theorem to show that if there exists a solution, then the optimal objective function value is zero.

6.5 Consider the optimization problem

$$\begin{array}{rl} \min & \frac{1}{2}\mathbf{x}^T Q\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{b} \end{array}$$

where $Q \in \mathbb{R}^{n\times n}$ is positive definite, $A \in \mathbb{R}^{m\times n}$ and $\text{rank}(A) = m$. Find all points satisfying the Lagrange condition for the problem in terms of Q, A and $\mathbf{b}$. Are they global minimizers for the problem?

Chapter 7

Representation of Linear Constraints

Many optimization problems found in practice have only linear constraints. In this chapter, we examine ways of representing linear constraints. The goal is to write the constraints in a form that makes it easy to move from one feasible point to another.

Consider the problem

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ \text{subject to} & \mathbf{a}_i^T\mathbf{x} = b_i, \quad i \in \mathcal{E} \\ & \mathbf{a}_i^T\mathbf{x} \geq b_i, \quad i \in \mathcal{I}. \end{array}$$

where $\mathcal{E}$, $\mathcal{I}$ are the index set for equality and inequality constraints.

Example Consider the problem

$$\begin{array}{ll} \min & f(\mathbf{x}) = x_1^2 + x_2^4 + x_3^4 \\ \text{subject to} & x_1 + 2x_2 + 3x_3 = 6 \\ & x_1,\ x_2,\ x_3 \geq 0. \end{array}$$

From the above we have

$$\begin{array}{lllllll} \mathbf{a}_1 & = & \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^T, & & \mathbf{a}_2 & = & \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T, \\ \mathbf{a}_3 & = & \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T, & & \mathbf{a}_4 & = & \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T, \\ \mathcal{E} & = & \{1\}, & & \mathcal{I} & = & \{2,3,4\} \end{array}$$ Ans

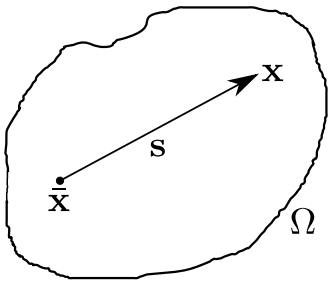

Figure 7.1: A feasible direction $\mathbf{s}$ at $\bar{\mathbf{x}}$ where Ω is a feasible set.

Let Ω denote a feasible set. Then we define a feasible direction vector.

Definition 7.1. *A vector* $\mathbf{s} \in \mathbb{R}^n$ *is called a* feasible direction *at the point* $\bar{\mathbf{x}}$ *if there exists some* $\epsilon > 0$ *such that*

$$\mathbf{x} \stackrel{\triangle}{=} \bar{\mathbf{x}} + \alpha\mathbf{s} \ \in \Omega \quad \forall\, 0 \leq \alpha \leq \epsilon.$$

■

In other words, $\mathbf{s}$ is a feasible direction at $\bar{\mathbf{x}}$ if any small step taken along $\mathbf{s}$ leads to a feasible point in Ω.

In many applications, it is useful to maintain feasibility at every iteration. This idea motivates a class of methods called *feasible point methods*.

Algorithmic model of feasible point methods

1. Specify an initial guess of the solution, $\mathbf{x}^{(0)} \in \Omega$.
2. For $k = 0, 1, 2, \ldots$ { Determine a feasible direction of descent $\mathbf{s}^{(k)}$ at the point $\mathbf{x}^{(k)}$.
   If none exists, then terminate.
   Otherwise, determine a new feasible point $\mathbf{x}^{(k+1)} \triangleq \mathbf{x}^{(k)} + \alpha_k \mathbf{s}^{(k)}$ such that $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$. }

7.1 Equality Constraints

Let us begin by considering the following example.

Example

$$\begin{array}{ll} \min & f(x_1, x_2) \\ \text{subject to} & x_1 + x_2 = 1. \end{array}$$

Let $\bar{\mathbf{x}} = [0,\ 1]^T$ so that $\bar{\mathbf{x}}$ satisfies the constraint. One can easily deduce that $\bar{\mathbf{x}} + \alpha\mathbf{s}$ will be a feasible point if and only if

$$s_1 + s_2 = 0 \qquad \underline{\text{Ans}}$$

Next we move on to consider a general optimization problem with linear equality constraints.

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ \text{subject to} & A\mathbf{x} = \mathbf{b}. \qquad (A \text{ is the constraint matrix}) \end{array}$$

It follows that a vector $\mathbf{s}$ is a feasible direction Feasible direction!linear equality constraints for the linear equality constraint if and only if

$$A\mathbf{s} = \mathbf{0}.$$

Thus, a direction $\mathbf{s}$ is feasible for the linear equality constraint if and only if it lies in the null space of A.

Recall that the null space of A is defined as the set of all vector satisfying $A\mathbf{s} = \mathbf{0}$. Moreover, the null space of A is a subspace of $\mathbb{R}^n$.

7.2 Inequality Constraints

Let us consider the following simple example.

Example

$$\begin{array}{ll} \min & f(x_1, x_2) \\ \text{subject to} & x_1 + x_2 \geq 1. \end{array}$$

Let $\bar{\mathbf{x}} = [0,\ 2]^T$. Then $\bar{\mathbf{x}}$ is feasible and the constraint is inactive at $\bar{\mathbf{x}}$. As a result, any $\mathbf{s} \in \mathbb{R}^2$ is a feasible direction.

Now consider the point $\bar{\mathbf{x}} = [0,\ 1]^T$. Then $\bar{\mathbf{x}} + \alpha\mathbf{s}$ will be a feasible point if and only if

$$s_1 + s_2 \geq 0 \qquad \underline{\text{Ans}}$$

Next we consider a general optimization problem with linear inequality constraints.

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ \text{subject to} & A\mathbf{x} \geq \mathbf{b} \end{array}$$

Let $\bar{\mathbf{x}}$ be a feasible point for this problem and let $\hat{A}$ be the submatrix of A corresponding to the rows of the active constraints at $\bar{\mathbf{x}}$. Then, a direction $\mathbf{s}$ is a feasible direction for $A\mathbf{x} \geq \mathbf{b}$ at $\bar{\mathbf{x}}$ if and only if

$$\hat{A}\mathbf{s} \geq \mathbf{0}.$$

In summary, the feasible directions at a point $\bar{\mathbf{x}}$ are determined by the equality constraints and the active inequalities at that point. ■

7.3 Mixed Equality-Inequality Constraints

Let $\hat{\mathcal{I}}$ denote the index set of active inequalities constraints at $\bar{\mathbf{x}}$. Then $\mathbf{s}$ is a feasible direction with respect to the feasible set at $\bar{\mathbf{x}}$ if and only if

$$\begin{array}{lll} & \mathbf{a}_i^T\mathbf{s} = 0, & i \in \mathcal{E} \\ \text{and} & \mathbf{a}_i^T\mathbf{s} \geq 0, & i \in \hat{\mathcal{I}}. \end{array}$$

Once a feasible direction $\mathbf{s}$ is determined, the new estimate of the solution is given by $\mathbf{x} = \bar{\mathbf{x}} + \alpha\mathbf{s},\ \alpha \geq 0$. Since the new point $\mathbf{x}$ must be feasible, in general there is an upper limit on how large α can be.

For an equality constraint, $(\mathbf{a}_i^T\mathbf{s} = 0)$

$$\mathbf{a}_i^T\mathbf{x} = \mathbf{a}_i^T(\bar{\mathbf{x}} + \alpha\mathbf{s}) = b_i \ \text{ for all } \alpha \geq 0.$$

For an active inequality constraint, $(\mathbf{a}_i^T\mathbf{s} \geq 0)$

$$\mathbf{a}_i^T\mathbf{x} = \mathbf{a}_i^T(\bar{\mathbf{x}} + \alpha\mathbf{s}) \geq \mathbf{a}_i^T\bar{\mathbf{x}} \geq b_i \text{ for all } \alpha \geq 0.$$

Therefore, only the inactive constraints are relevant when determining an upper bound of α. Now we consider the constraints that are inactive at $\bar{\mathbf{x}}$. These can be divided into two cases:

- If $\mathbf{a}_i^T\bar{\mathbf{x}} > b_i$ and if $\mathbf{a}_i^T\mathbf{s} \geq 0$, then the constraint $\mathbf{a}_i^T\mathbf{x} \geq b_i$ remains satisfied for all $\alpha \geq 0$. As α increases, the movement is away from the boundary of the constraint.
- If $\mathbf{a}_i^T\bar{\mathbf{x}} > b_i$ and if $\mathbf{a}_i^T\mathbf{s} < 0$, then the inequality will remain valid for

  $$\alpha_i \leq \frac{(\mathbf{a}_i^T\bar{\mathbf{x}} - b_i)}{-\mathbf{a}_i^T\mathbf{s}}.$$

  A positive step along $\mathbf{s}$ is a move towards the boundary and any step larger than α will violate the constraint.

Hence, it readily follows from the above that the maximum step length $\bar{\alpha}$ that maintain feasibility is obtained from

$$\bar{\alpha} = \min_i \left\{ \frac{(\mathbf{a}_i^T\bar{\mathbf{x}} - b_i)}{-\mathbf{a}_i^T\mathbf{s}} : \mathbf{a}_i^T\mathbf{s} < 0 \right\}.$$

Definition 7.2. *A subspace of a vector space is a nonempty subset that satisfies 2 requirements:*

1. *If we add any vectors $\mathbf{x}, \mathbf{y}$ in the subspace, then their sum is still in the subspace.*
2. *If we multiply any vector $\mathbf{x}$ in the subspace by any scalar c, then the multiple $c\mathbf{x}$ is still in the subspace.* ■

Null Spaces Let $A \in \mathbb{R}^{m\times n},\ m \leq n$. Let $\mathcal{N}(A)$ be defined as the null space of A; *i.e.*

$$\mathcal{N}(A) = \{\mathbf{s} \in \mathbb{R}^n : A\mathbf{s} = \mathbf{0}\}.$$

Notice that he null space of A is the set of vectors orthogonal to the rows of the matrix A. It is easy to see that any linear combination of two vectors in $\mathcal{N}(A)$ is also in $\mathcal{N}(A)$ and therefore the null space is a subspace of $\mathbb{R}^n$.

It can be shown that the dimension of this subspace is $n - \text{rank}(A)$. Further, if A has full rank (*i.e.* $\text{rank}(A) = m$), then

$$\dim[\mathcal{N}(A)] = n - m.$$

Range Spaces The range space of a matrix is the set of vectors spanned by the columns of the matrix. In particular, we are interested in the range space of A^T, defined by

$$\mathcal{R}(A^T) = \{\mathbf{q} \in \mathbb{R}^n : \mathbf{q} = A^T\boldsymbol{\lambda} \text{ for some } \boldsymbol{\lambda} \in \mathbb{R}^m\}.$$

It is important to note that

$$\mathcal{N}(A) \quad \text{is orthogonal to} \quad \mathcal{R}(A^T).$$

That is to say, any vector in one subspace is orthogonal to any vector in the other.

How to Represent Vectors in $\mathcal{N}(A)$

Define Z as a null space matrix of A if any vector in $\mathcal{N}(A)$ can be expressed as a linear combination of the columns of Z.

The representation of a null space matrix is not unique. If A has full row rank m, any matrix Z of dimension $n \times v$ and rank $n - m$ that satisfies $AZ = 0$ is a null space matrix. The column dimension r must be at least $(n - m)$.

In the special case where $r = n - m$, the columns of Z are linearly independent; in which case, Z is called a basis matrix for the null space of A.

If Z is an $n \times v$ null space matrix, the null space can be represented as

$$\mathcal{N}(A) = \{\mathbf{p} : \mathbf{p} = Z\mathbf{v} \text{ for some } \mathbf{v} \in \mathbb{R}^r\},$$

and hence

$$\mathcal{N}(A) = \mathcal{R}(Z).$$

This representation of the null space gives us a practical way to generate feasible points. If $\bar{\mathbf{x}}$ is any point satisfying $A\mathbf{x} = \mathbf{b}$, then all other feasible points can be written as

$$\mathbf{x} = \bar{\mathbf{x}} + Z\mathbf{v} \qquad \text{for some } \mathbf{v}.$$

Example $A = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$.

$\mathcal{N}(A)$ is the set of all vectors $\mathbf{p} \in \mathbb{R}^4$ such that

$$A\mathbf{p} = \begin{bmatrix} p_1 - p_2 \\ p_3 + p_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

That is, the vector $\mathbf{p}$ must satisfy

$$p_1 = p_2 \qquad \text{and} \qquad p_3 = -p_4.$$

Thus, any null-space vector must have the form

$$\mathbf{p} = \begin{bmatrix} v_1 & v_1 & v_2 & -v_2 \end{bmatrix}^T$$

for some scalar v_1, v_2.

A possible basis matrix for $\mathcal{N}(A)$ is

$$Z = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}.$$

and $\mathcal{N}(A) = \{\mathbf{p} : \mathbf{p} = Z\mathbf{v} \text{ for some } \mathbf{v} \in \mathbb{R}^2\}$. Ans

7.4 Exercises

7.1 Consider the set defined by the constraints $x_1 + x_2 = 1$, $x_1 \geq 0$ and $x_2 \geq 0$. Determine the set of feasible directions at each of the following points $\bar{x}$: (a) $\bar{x} = (0, 1)^T$; (b) $\bar{x} = (1, 0)^T$; (c) $\bar{x} = (0.5, 0.5)^T$.

7.2 Let $S = \{x : Ax \leq b\}$. Derive the conditions that must be satisfied by a feasible direction at a point $\bar{x} \in S$.

Chapter 8

Linear Programming

In this chapter, we examine linear programming problems, which are optimization problems where the objective function and all the constraint functions are linear. Many problems found in practical applications can be cast as linear programs.

In solving such problems, we will concentrate our attention here only to the simplex method, due to George B. Dantzig. As will be seen later, the solution of a linear program can be found by searching through a finite number of feasible points, known as basic feasible solutions.

Formally, a linear program is expressed as

$$\begin{array}{lrcl} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & f(\mathbf{x}) & = & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{a}_i^T\mathbf{x} & = & b_i, \quad i \in \mathcal{E} \\ & \mathbf{a}_i^T\mathbf{x} & \geq & b_i, \quad i \in \mathcal{I} \end{array}$$

where $\mathcal{E}$ and $\mathcal{I}$ are, respectively, the index sets for equality and inequality constraints.

Next we show that a two dimensional linear problem can be solved graphically. Consider the problem

$$\begin{array}{lrcl} \min & z = -x_1 - 2x_2 & & \\ \text{subject to} & -2x_1 + x_2 & \leq & 2 \\ & -x_1 + x_2 & \leq & 3 \\ & x_1 & \leq & 3 \\ & x_1, x_2 & \geq & 0. \end{array} \tag{8.1}$$

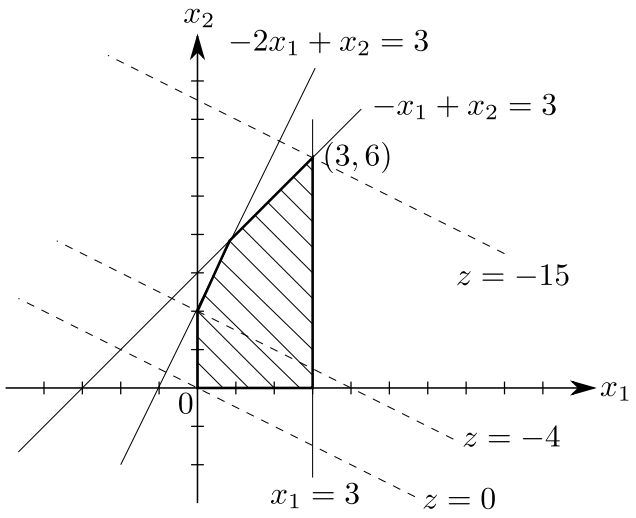

Figure 8.1: Geometric solution of a linear program in $\mathbb{R}^2$.

8.1 Standard Form

There are many different ways to represent a linear program. It is sometimes more convenient to use one instead of another, at times to make a property of the linear program more apparent, at other times to simplify the description of an algorithm.

One such representation, called *standard form*, will be used in connection with the simplex method. As will be seen later, rules for converting linear programs to standard form are simple and can be performed automatically by computer.

In vector-matrix notation, a linear program in standard form will be written as

$$\left.\begin{array}{rl} \min & z = \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{array}\right\}, \tag{8.2}$$

where $\mathbf{b} \geq \mathbf{0}$, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$. $A \in \mathbb{R}^{m \times n}$ is called the constraint matrix.

It is important to note the following.

1. The LP (8.2) is a minimization problem;
2. All the variables (x_i) are constrained to be non-negative;
3. All the other constraints are represented as equations;
4. All b_i are nonnegative.

This will be the form of a linear program used within the simplex method. In other settings, other forms of a linear program may be more convenient.

The following is an example of linear program in standard form.

$$\begin{array}{rl} \min & 4x_1 - 5x_2 + 3x_3 \\ \text{subject to} & 3x_1 - 2x_2 + 7x_3 = 7 \\ & 8x_1 + 6x_2 + 6x_3 = 5 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \\ & x_3 \geq 0. \end{array}$$

All linear programs can be converted to the standard form. In the following examples, we show how to do this.

- $\max\ \mathbf{c}^T\mathbf{x} \quad \rightarrow \quad \min\ -\mathbf{c}^T\mathbf{x}$
  (The optimal values of the variables are the same for both objective functions.)

- $\exists_i\ b_i < 0 \quad \rightarrow \quad$ multiply the constraint by -1.

  $$(\mathbf{a}_i^T\mathbf{x} \leq b_i \quad \rightarrow \quad -\mathbf{a}_i^T\mathbf{x} \geq -b_i)$$

- $x_1 \geq 5 \qquad \rightarrow \qquad x_1' = x_1 - 5$
  $$x_1' \geq 0$$

- free variable (unrestricted variable)
  If x_2 is unrestricted, then it is replaced by
  $$x_2 = x_2' - x_2''$$
  $$\text{and} \qquad x_2' \geq 0, \qquad x_2'' \geq 0$$

- inequality constraint with $\leq$
  $$2x_1 + 7x_2 - 3x_3 \leq 10$$
  $$\Longrightarrow \qquad \begin{cases} 2x_1 + 7x_2 - 3x_3 + s_1 = 10 \\ \text{and} \qquad s_1 \geq 0 \end{cases}$$
  s_1 is called a *slack* variable.

- inequality constraint with $\geq$
  $$6x_1 - 2x_2 + 4x_3 \geq 15$$
  $$\Longrightarrow \qquad \begin{cases} 6x_1 - 2x_2 + 4x_3 - e_2 = 15 \\ \text{and} \qquad e_2 \geq 0 \end{cases}$$
  e_2 is called an excess variable.

8.2 Extreme Points & Basic Feasible Solutions

An extreme point (or a vertex) is defined geometrically using convexity.

Definition 8.2.1. *A point $\mathbf{x} \in S$, where S is a convex set, is called an* extreme point *(or a vertex) of S if it cannot be expressed in the form*

$$\mathbf{x} = \alpha\mathbf{y} + (1-\alpha)\mathbf{z},$$

with $\mathbf{y}, \mathbf{z} \in S$, $\alpha \in (0,1)$ and $\mathbf{y}, \mathbf{z} \neq \mathbf{x}$. ■

That is to say, $\mathbf{x}$ cannot be expressed as a convex combination of other feasible points $\mathbf{y}, \mathbf{z}$.

Basic Solutions A basic solution is defined algebraically using the standard form of the constraints.

Definition 8.2.2. *A point* $\mathbf{x}$ *is called a* basic solution *(of* $A\mathbf{x} = \mathbf{b}$*) if*

- $\mathbf{x}$ *satisfies* $A\mathbf{x} = \mathbf{b}$
- *the columns of the matrix* A *corresponding to the nonzero components of* $\mathbf{x}$ *are linearly independent.* ■

As A has full row rank ($= m$), we can separate $\mathbf{x}$ into two subvectors $\mathbf{x}_B$ and $\mathbf{x}_N$, where $\mathbf{x}_N$ consists of $n - m$ nonbasic variables (all of which are zero) and $\mathbf{x}_B$ consists of m basic variables. The constraint coefficients of $\mathbf{x}_B$ correspond to an invertible $m \times m$ basis matrix B.

Definition 8.2.3. *A point* $\mathbf{x}$ *is a* basic feasible solution *(bfs) if it is a basic solution and also satisfies the constraint* $\mathbf{x} \geq \mathbf{0}$. ■

Consider LP on page 97 again.

$$\left.\begin{array}{rrcl} \min & z = -x_1 - 2x_2 & & \\ \text{subject to} & -2x_1 + x_2 & \leq & 2 \\ & -x_1 + x_2 & \leq & 3 \\ & x_1 & \leq & 3 \\ & x_1, x_2 & \geq & 0. \end{array}\right\} \tag{8.3}$$

In standard form, LP (8.3) can be rewritten as

$$\left.\begin{array}{rrcccccl} \min & z = -x_1 - 2x_2 & & & & & & \\ \text{subject to} & -2x_1 & +x_2 & +s_1 & & & = & 2 \\ & -x_1 & +x_2 & & +s_2 & & = & 3 \\ & x_1 & & & & +s_3 & = & 3 \\ & x_1, & x_2, & s_1, & s_2, & s_3 & \geq & 0. \end{array}\right\} \tag{8.4}$$

Notice that in the standard form, the problem has 5 variables. In (8.4), the basis $\{x_2, s_1, s_3\}$ produce the basic solution

$$\begin{bmatrix} x_1 & x_2 & s_1 & s_2 & s_3 \end{bmatrix}^T = \begin{bmatrix} 0 & 3 & -1 & 0 & 3 \end{bmatrix}^T,$$

where $x_2,\ s_1,\ s_3$ are basic variables.

The basis $\{s_1, s_2, s_3\}$ produces the basic feasible solution

$$\begin{bmatrix} x_1 & x_2 & s_1 & s_2 & s_3 \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 2 & 3 & 3 \end{bmatrix}^T.$$

Similarly, if the basis $\{x_1, x_2, s_1\}$ is chosen, then we obtain the basic feasible solution

$$\begin{bmatrix} x_1 & x_2 & s_1 & s_2 & s_3 \end{bmatrix}^T = \begin{bmatrix} 3 & 6 & 2 & 0 & 0 \end{bmatrix}^T,$$

which is the optimal basic feasible solution of this problem.

Let $\mathbf{x}$ be any basic feasible solution. Once a set of basic variables has been selected, it is possible to reorder the variables so that the basic variables are listed first.

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} \begin{matrix} \} \, m \text{ components} \\ \} \, n - m \text{ components} \end{matrix}$$

The constraint matrix can then be written as

$$A = \begin{bmatrix} B & N \end{bmatrix},$$

where B and N are the coefficient matrix for $\mathbf{x}_B$ and $\mathbf{x}_N$, respectively.

For a basic solution, we have $\mathbf{x}_N = \mathbf{0}$, so that the set of constraints $A\mathbf{x} = \mathbf{b}$ simplifies to $B\mathbf{x}_B = \mathbf{b}$; that is to say,

$$\begin{aligned} A\mathbf{x} &= \begin{bmatrix} B & N \end{bmatrix} \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} \\ &= B\mathbf{x}_B = \mathbf{b}. \end{aligned}$$

Hence, $\mathbf{x}_B$ is determined by B and $\mathbf{b}$.

The number of basic feasible solution is finite, and is bounded by the number of ways that the m variables $\mathbf{x}_B$ can be selected from among the n variables $\mathbf{x}$. This number, which is the number of all basic solutions, is

$$\begin{pmatrix} n \\ m \end{pmatrix} = \frac{n!}{m!(n-m)!},$$

where $n! = n(n-1)(n-2)\cdots 3 \cdot 2 \cdot 1$.

The concept of an extreme point is equivalent to that of a basic feasible solution. This is stated as follows.

Theorem 8.2.4. *A point* $\mathbf{x}$ *is an extreme point of the set*

$$\{\mathbf{x} : A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$$

if and only if it is a basic feasible solution. ■

Proof See [26], pp. 80–81. ■

It is possible that one or more of the basic variables in a basic feasible solution will be zero. In which case, the point is called a degenerate vertex (basic feasible solution) and the linear program is said to be degenerate.

At a degenerate vertex, several different bases may correspond to the same basic feasible solution. Degeneracy can arise when a linear program contains a redundant constraint. For example,

$$\begin{aligned} 2x_1 &\leq 6, \\ 3x_1 &\leq 13, \\ 4x_1 &\leq 12. \end{aligned}$$

The first and third constraints are equivalent, so either of them could be moved from the problem without changing the solution.

Geometrically, two extreme points (vertices) are adjacent if they are connected by and edge of the feasible region.

It is important to note that for a linear program in standard form with m equality constraints, two bases will be adjacent if they have $m - 1$ variables in common.

We now arrive at a fundamental result in LP, which is based on the *representation theorem*[1]. See [26] for further details.

Theorem 8.2.5. *If a linear programme in standard form* (8.2) *has a finite optimal solution, then it has an optimal basic feasible solution.* ■

Proof See [26], pp. 90–91. ■

[1]Any feasible point can be represented as a convex combination of extreme points plus, possibly, a direction of unboundedness.

The above theorem implies that a solution to a linear program (if one exists) can always be chosen from among the vertices of the feasible region.

8.3 Simplex Method

The simplex method is a feasible direction method for solving a linear programming problem written in standard form.

It moves from one extreme point (or basic feasible solution) to another. At each iteration, the method tests to see if the current basis is optimal. If not, it selects a feasible direction along which the objective function improves and moves to an adjacent basic feasible solution along that direction.

Simplex Tableau The tableaus are merely notational devices that help us explain the simplex method.

$$\begin{array}{rl} \min & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{array}$$

basic	x_1 x_2 ... x_n	rhs
x_3 x_4 x_5	A	$\mathbf{b}$
$-z$	$\mathbf{c}^T$	0

Example Consider the following LP.

$$\begin{array}{rlrrrrrrl} \min & z = -x_1 - 2x_2 & & & & & & & \\ \text{subject to} & -2x_1 & +x_2 & +x_3 & & & = & 2 \\ & -x_1 & +2x_2 & & +x_4 & & = & 7 \\ & x_1 & & & & +x_5 & = & 3 \\ & x_1,\ x_2,\ x_3,\ x_4,\ x_5 & \geq & 0. & & & & & \end{array}$$

Step 1 Choose $\{x_3,\ x_4,\ x_5\}$ as the basis (set of basic variables).

leaving variable $\Rightarrow$

basic	x_1	x_2	x_3	x_4	x_5	rhs	ratio
x_3	–2	1	1	0	0	2	2/1=2
x_4	–1	2	0	1	0	7	7/2=3.5
x_5	1	0	0	0	1	3	
$-z$	–1	–2	0	0	0	0	

$$\Uparrow \text{ entering variable}$$

$$\text{basic feasible solution} = \begin{bmatrix} 0 & 0 & 2 & 7 & 3 \end{bmatrix}^T \quad \text{and} \quad z = 0.$$

Choose x_2 as the entering variable (*i.e.* use x_2 as a basic var. in step 2).

Choose x_3 as the leaving variable (*i.e.* let x_3 be a non-basic var. in the next step).

Step 2 choose $\{x_2,\ x_4,\ x_5\}$ as the basis. Then perform row operations to the tableau in step 1 so that the column of x_2 becomes $\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$.

basic	x_1	x_2	x_3	x_4	x_5	rhs	ratio
x_2	-2	1	1	0	0	2	
x_4	3	0	-2	1	0	3	$3/3 = 1$
x_5	1	0	0	0	1	3	$3/1 = 3$
$-z$	-5	0	2	0	0	4	

leaving variable $\Rightarrow$ (x_4)

$$\Uparrow \text{ entering variable}$$

$$\therefore \quad \text{basic feasible solution} = \begin{bmatrix} 0 & 2 & 0 & 3 & 3 \end{bmatrix}^T \quad \text{and} \quad z = -4.$$

Choose x_1 as the entering variable and x_4 as the leaving variable.

Step 3 Choose $\{x_2,\ x_1,\ x_5\}$ as the basis.

basic	x_1	x_2	x_3	x_4	x_5	rhs	ratio
x_2	0	1	$-1/3$	$2/3$	0	4	
x_1	1	0	$-2/3$	$1/3$	0	1	
x_5	0	0	$2/3$	$-1/3$	1	2	$\frac{2}{2/3}$
$-z$	0	0	$-4/3$	$5/3$	0	9	

leaving variable $\Rightarrow$ (x_5)

$$\Uparrow \text{ entering variable}$$

$$\therefore \quad \text{basic feasible solution} = \begin{bmatrix} 1 & 4 & 0 & 0 & 2 \end{bmatrix}^T \quad \text{and} \quad z = -9.$$

Choose x_3 as the entering variable and x_5 as the leaving variable.

Step 4 Choose $\{x_2,\ x_1,\ x_3\}$ as the basis.

basic	x_1	x_2	x_3	x_4	x_5	rhs
x_2	0	1	0	1/2	1/2	5
x_1	1	0	0	0	1	3
x_3	0	0	1	$-1/2$	3/2	3
$-z$	0	0	0	1	2	13
	↑	↑	↑	↑	↑	

All the coefficients are non-negative. Therefore, this step yields the optimal solution.

$\therefore$ optimal basic feasible solution $= \begin{bmatrix} 3 & 5 & 3 & 0 & 0 \end{bmatrix}^T$ and $z^* = -13$. Ans

8.4 Artificial Variables

The simplex method moves from one basic feasible solution to another until either a solution is found or until it is determined that the problem is unbounded.

How to find a basic feasible solution? The previous example clearly shows that if there are only the constraints $\mathbf{a}_i^T\mathbf{x} \leq b_i$ and $\mathbf{x} \geq \mathbf{0}$, then it is very easy to obtain a basic feasible solution.

Unfortunately, general problems will not have this property, raising the question of how to find a basic feasible solution in a systematic fashion.

Two standard approaches used in connection with the simplex method are **Two-Phase** and **Big-M** methods. Both use the device of artificial variables, *i.e.*, extra variables that are temporarily added to the problem.

To illustrate the idea of artificial variables, consider the folowing LP.

$$\begin{aligned} \min \quad & z = 2x_1 + 3x_2 \\ \text{subject to} \quad & 3x_1 + 2x_2 = 14, \\ & 2x_1 - 4x_2 \geq 2, \\ & 4x_1 + 3x_2 \leq 19, \\ & x_1, x_2 \geq 0. \end{aligned} \tag{8.5}$$

Arranging (8.5) to standard form and adding an artificial variable a_i to every constraint

that does not have a slack variable, we have

$$\begin{array}{lllll} \min & z = 2x_1 + 3x_2 & & & \\ \text{subject to} & 3x_1 + 2x_2 & + a_1 & = 14, & \\ & 2x_1 - 4x_2 - x_3 & + a_2 & = 2, & \\ & 4x_1 + 3x_2 + x_4 & & = 19, & \\ & x_1, x_2, x_3, x_4, a_1, a_2 \geq 0. & & & \end{array} \tag{8.6}$$

Clearly, it is now possible to initialize the simplex method with $\mathbf{x}_B = [a_1,\ a_2,\ x_4]^T$, where

$$a_1 = 14, \quad a_2 = 2, \quad x_4 = 19.$$

Since the a_i's are not part of the original problem, this choice of basis does not correspond to a basic feasible solution to the LP (8.5).

To obtain a basic feasible solution to (8.5), we need to move to a basic feasible solution that does not include any a_i as a basic variable. Such a basic feasible solution indeed corresponds to a basic feasible solution of (8.5). If the a_i's cannot be driven to zero, then the constraints for the original problem (8.5) are infeasible and the problem has no solution.

8.4.1 Two-phase method

The artificial variables a_i's are used to create an auxiliary LP, called *the phase I problem*, whose only purpose is to compute a basic feasible solution of the original problem. Now the objective function to be minimized is

$$\sum_i a_i.$$

If the original LP is feasible, then the phase I problem will have optimal value $z'_* = 0$. Otherwise, $z'_* > 0$. This leads to a basis that does not contain all the a_i's and hence a basic feasible solution to the original LP.

Then the basic feasible solution that is obtained from solving the phase I problem can be used as an initial basic feasible solution for the original LP. This is called the *phase II* problem.

Example Consider the problem (8.6). The phase I problem is as follows.

Step 1 Choose $\{a_1,\ a_2,\ x_4\}$ as the basis.

basic	x_1	x_2	x_3	x_4	a_1	a_2	rhs
a_1	3	2	0	0	1	0	14
a_2	2	-4	-1	0	0	1	2
x_4	4	3	0	1	0	0	19
$-z'$	0	0	0	0	1	1	0

Notice that the top row entries of a_1 and a_2 are not zero, so that z' is not expressed only in terms of the nonbasic variables. These entries must be removed by elimination:

	basic	x_1	x_2	x_3	x_4	a_1	a_2	rhs
	a_1	3	2	0	0	1	0	14
leaving variable $\Rightarrow$	a_2	2	-4	-1	0	0	1	2
	x_4	4	3	0	1	0	0	19
	$-z'$	-5	2	1	0	0	0	-16

$\Uparrow$ entering variable

Elimination must be applied to the top row of a tableau whenever the entries for the initial basic variables are not zero.

The reduced cost of x_1 is negative so this basis is not optimal. The ratio test indicates that a_2 is the leaving variable. We would like to remove a_2 from the problem completely. The artificial variables were added to constraints where there was no obvious choice for a basic variable. In the current basis x_1 serves that function for the second constraint, and a_2 is no longer required. For this reason a_2 (or any other artificial variable that has left the basis) can be removed from the problem.

Step 2 Choose $\{a_1,\ x_1,\ x_4\}$ as the basis.

	basic	x_1	x_2	x_3	x_4	a_1	rhs
	a_1	0	8	$\frac{3}{2}$	0	1	11
	x_1	1	-2	$-\frac{1}{2}$	0	0	1
leaving variable $\Rightarrow$	x_4	0	11	2	1	0	15
	$-z'$	0	-8	$-\frac{3}{2}$	0	0	-11

$\Uparrow$ entering variable

The reduced costs for x_2 and x_3 are negative, so this basis is not optimal. Since the coefficient of x_2 is larger in magnitude, x_2 will be selected as the entering variable. Then x_4 is the leaving variable.

Step 3 Choose $\{a_1,\ x_1,\ x_2\}$ as the basis.

leaving variable $\Rightarrow$

basic	x_1	x_2	x_3	x_4	a_1	rhs
a_1	0	0	$\frac{1}{22}$	$-\frac{8}{11}$	1	$\frac{1}{11}$
x_1	1	0	$-\frac{3}{22}$	$\frac{2}{11}$	0	$\frac{41}{11}$
x_2	0	1	$\frac{2}{11}$	$\frac{1}{11}$	0	$\frac{15}{11}$
$-z'$	0	0	$-\frac{1}{22}$	$\frac{8}{11}$	0	$-\frac{1}{11}$

$\Uparrow$ entering variable

The reduced cost for x_3 is negative so this basis is not optimal and x_3 is the entering variable. The ratio test shows that a_1 is the leaving variable.

Step 4 Choose $\{x_3,\ x_1,\ x_2\}$ as the basis.

basic	x_1	x_2	x_3	x_4	rhs
x_3	0	0	1	-16	2
x_1	1	0	0	-2	4
x_2	0	1	0	3	1
$-z'$	0	0	0	0	0

The current basis does involve any artificial variables and the objective value is zero, so this is a feasible point for the constraints of the original problem.

The solution of the phase I problem only gives a basic feasible solution for the original problem; it is not optimal. It can be used as an initial basic feasible solution for the original problem with objective $z = 2x_1 + 3x_2$. In the tableau:

basic	x_1	x_2	x_3	x_4	rhs
x_3	0	0	1	-16	2
x_1	1	0	0	-2	4
x_2	0	1	0	3	1
$-z$	2	3	0	0	0

If the simplex method is implemented without a tableau, then all that is necessary is to retain the final basis from the phase I problem as the initial basis of the phase II problem.

The reduced costs for the basic variables x_1 and x_2 are not zero, and must be eliminated before the simplex method can be used:

basic	x_1	x_2	x_3	x_4	rhs
x_3	0	0	1	-16	2
x_1	1	0	0	-2	4
x_2	0	1	0	3	1
$-z$	0	0	0	-5	-11

The reduced cost for x_4 is negative so this basis is not optimal. Only x_2 is a candidate for the ratio test, so it is the leaving variable. Pivoting gives

basic	x_1	x_2	x_3	x_4	rhs
x_3	0	$\frac{16}{3}$	1	0	$\frac{22}{3}$
x_1	1	$\frac{2}{3}$	0	0	$\frac{14}{3}$
x_4	0	$\frac{1}{3}$	0	1	$\frac{1}{3}$
$-z$	0	$\frac{5}{3}$	0	0	$-\frac{28}{3}$
		$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$

All the coefficients are non-negative. Therefore, this step yields the optimal solution.
$\therefore$ optimal basic feasible solution $= \begin{bmatrix} 14/3 & 0 & 22/3 & 1/3 \end{bmatrix}^T$ and $z^* = 28/3$. Ans

8.4.2 Big-M Method

In contrast to the two-phase method, penalty terms are added to the objective function that are designed to push the a_i's out of the basis. To this end, the objective function is changed to

$$\min z' = \mathbf{c}^T\mathbf{x} + M\sum_i a_i,$$

where M denotes a large positive number.

If M is large, then any basis that contains an $a_i > 0$ will lead to a large positive value of z'. If there is any basic feasible solution to the original LP, then the corresponding basis will not include any a_i; hence the value of z' gets smaller.

Any basic feasible solution to the penalized problem in which all the a_i's are nonbasic is also a basic feasible solution to the original problem. The corresponding basis can be used as an initial basis for the original problem.

Example Consider the problem (8.5) again.

Step 1 The initial tableau for the penalized problem with artificial variables is

basic	x_1	x_2	x_3	x_4	a_1	a_2	rhs
a_1	3	2	0	0	1	0	14
a_2	2	-4	-1	0	0	1	2
x_4	4	3	0	1	0	0	19
$-z'$	2	3	0	0	M	M	0

As before, the reduced costs for the artificial variables must be zeroed via elimination:

leaving variable $\Rightarrow$

basic	x_1	x_2	x_3	x_4	a_1	a_2	rhs
a_1	3	2	0	0	1	0	14
a_2	2	-4	-1	0	0	1	2
x_4	4	3	0	1	0	0	19
$-z'$	$-5M+2$	$2M+3$	M	0	0	0	$-16M$

$\Uparrow$ entering variable

At the first iteration, x_1 is the entering variable and a_2 is the leaving variable. As in the two-phase method, once an artificial leaves the basis it becomes irrelevant and can be removed from the problem.

Step 2 Choose $\{a_1,\ x_1,\ x_4\}$ as the basis.

basic	x_1	x_2	x_3	x_4	a_1	rhs
a_1	0	8	$\frac{3}{2}$	0	1	11
x_1	1	-2	$-\frac{1}{2}$	0	0	1
x_4	0	11	2	1	0	15
$-z'$	0	$-8M+7$	$-\frac{3}{2}M+1$	0	0	$-11M-2$

leaving variable $\Rightarrow$

$\Uparrow$ entering variable

Choose x_2 as the entering variable and x_4 as the leaving variable.

Step 3 Choose $\{a_1,\ x_1,\ x_2\}$ as the basis.

basic	x_1	x_2	x_3	x_4	a_1	rhs
a_1	0	0	$\frac{1}{22}$	$-\frac{8}{11}$	1	$\frac{1}{11}$
x_1	1	0	$-\frac{3}{22}$	$\frac{2}{11}$	0	$\frac{41}{11}$
x_2	0	1	$\frac{2}{11}$	$\frac{1}{11}$	0	$\frac{15}{11}$
$-z'$	0	0	$-\frac{M+6}{22}$	$\frac{8M-7}{11}$	0	$-\frac{M+127}{11}$

leaving variable $\Rightarrow$

$\Uparrow$ entering variable

Choose x_3 as the entering variable and a_1 as the leaving variable.

Step 4 Choose $\{x_3,\ x_1,\ x_2\}$ as the basis.

basic	x_1	x_2	x_3	x_4	rhs
x_3	0	0	1	-16	2
x_1	1	0	0	-2	4
x_2	0	1	0	3	1
$-z'$	0	0	0	-5	-11

leaving variable $\Rightarrow$

$\Uparrow$ entering variable

The current basis does not involve any artificial variables, so this is a feasible point for the original problem. With the artificial variables gone, the objective function is now that of the original linear program.

Choose x_4 as the entering variable and x_2 as the leaving variable.

Step 5 Choose $\{x_3,\ x_1,\ x_4\}$ as the basis.

basic	x_1	x_2	x_3	x_4	rhs
x_3	0	$\frac{16}{3}$	1	0	$\frac{22}{3}$
x_1	1	$\frac{2}{3}$	0	0	$\frac{14}{3}$
x_4	0	$\frac{1}{3}$	0	1	$\frac{1}{3}$
$-z$	0	$\frac{5}{3}$	0	0	$-\frac{28}{3}$

$\uparrow \quad \uparrow \quad \uparrow \quad \uparrow$

All the coefficients are non-negative. Therefore, this step yields the optimal solution.

$\therefore$ optimal basic feasible solution $= \begin{bmatrix} 14/3 & 0 & 22/3 & 1/3 \end{bmatrix}^T$ and $z^* = 28/3$. Ans

As expected, it is the same as the optimal basis obtained using the two-phase method.

8.5 Exercises

8.1 Consider a linear program with the constraints in standard form

$$A\mathbf{x} = \mathbf{b} \quad \text{and} \quad \mathbf{x} \geq \mathbf{0}.$$

Prove that if $\mathbf{d} \neq \mathbf{0}$ satisfies

$$A\mathbf{d} = \mathbf{0} \quad \text{and} \quad \mathbf{d} \geq \mathbf{0}$$

then $\mathbf{d}$ is a direction of unboundedness.

8.2 Prove that the set $S = \{\mathbf{x} : A\mathbf{x} < \mathbf{b}\}$ does not contain any extreme points.

8.3 Solve the problem

$$\begin{aligned} \min \quad & z = -4x_1 - 2x_2 - 8x_3 \\ \text{subject to} \quad & 2x_1 - x_2 + 3x_3 \leq 30 \\ & x_1 + 2x_2 + 4x_3 = 40 \\ & x_1, x_2, x_3 \geq 0, \end{aligned}$$

using the two-phase method.

8.4 Solve the problem

$$\begin{aligned} \min \quad & z = -4x_1 - 2x_2 \\ \text{subject to} \quad & 3x_1 - 2x_2 \geq 4 \\ & -2x_1 + x_2 = 2 \\ & x_1, x_2 \geq 0, \end{aligned}$$

using the two-phase method.

8.5 Consider the linear programming problem

$$\begin{aligned} \min \quad & z = \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & A\mathbf{x} \geq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

where $\mathbf{b} \geq \mathbf{0}$. It is possible to use a single artificial variable to obtain an initial basic feasible solution to this problem. Let $\mathbf{s}$ be the vector of surplus variables, $\mathbf{e} = [1, \ldots, 1]^T$ and a be an artificial variable. Consider the phase I problem

$$\begin{aligned} \min \quad & z' = a \\ \text{subject to} \quad & A\mathbf{x} - \mathbf{s} + a\mathbf{e} = \mathbf{b} \\ & \mathbf{x}, \mathbf{s} \geq \mathbf{0}, a \geq 0. \end{aligned}$$

(a) Assume for simplicity that $b_1 = \max\{b_i\}$. Prove that $\{a, s_2, s_3, \ldots, s_n\}$ is a feasible basis for the new problem.

(b) Prove that if the original problem is feasible, then the phase I problem will have optimal objective value $z'_* = 0$, and if the original problem is infeasible it will have optimal objective value $z'_* > 0$.

8.6 Use the two-phase method to check whether the following linear program:

$$\begin{array}{lrcl} \text{minimize} & z = x_1 + 2x_2 - 8x_3 & & \\ \text{subject to} & -2x_1 + x_2 + 4x_3 & = & -5, \\ & x_1 + 2x_2 + 4x_3 & = & 20, \\ & -x_1 - x_2 + x_3 & \leq & 2, \\ & x_1, x_2, x_3 & \geq & 0. \end{array}$$

has a bounded optimal solution. If so, find the solution. Provide the detail of your calculation.

8.7 Use the big M method to check whether the following linear program:

$$\begin{array}{lrcl} \text{minimize} & z = x_1 + 2x_2 - 8x_3 & & \\ \text{subject to} & -2x_1 + x_2 + 4x_3 & = & -5, \\ & x_1 + 2x_2 + 4x_3 & = & 20, \\ & -x_1 - x_2 + x_3 & \leq & 2, \\ & x_1, x_2, x_3 & \geq & 0. \end{array}$$

has a bounded optimal solution. If so, find the solution. Provide the detail of your calculation.

Chapter 9

Quadratic Programming

Like linear programming problems, a number of practical optimization problems are posed as quadratic programming problems, where the objective function is quadratic and the constraints are linear. It is worth noting that convex quadratic programming problems can be solved in a finite number of steps.

The general quadratic program can be expressed as

$$\left.\begin{array}{lll} \min\limits_{\mathbf{x}} & \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} & \\ \text{subject to} & \mathbf{a}_i^T\mathbf{x} = b_i, & i \in \mathcal{E} \\ & \mathbf{a}_i^T\mathbf{x} \leq b_i, & i \in \mathcal{I} \end{array}\right\} \tag{9.1}$$

where $\mathcal{E}$ and $\mathcal{I}$ are, respectively, index sets for equality and inequality constraints, $Q = Q^T$.

In this course, we focus our attention only to problem (9.1) with $Q \geq 0$. In this case, problem (9.1) is convex and it can be shown that a local minimizer $\mathbf{x}^*$ is a global one. Furthermore, if $Q > 0$, then it follows that $\mathbf{x}^*$ is a unique global solution. For the case where Q is indefinite, it follows that (9.1) is difficult and will not be covered here.

9.1 Problems with Equality Constraints

Consider the following quadratic program

$$\left.\begin{array}{ll} \min & \frac{1}{2}\,\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{b} \qquad (A\mathbf{x} - \mathbf{b} = \mathbf{0}) \end{array}\right\} \tag{9.2}$$

where $A \in \mathbb{R}^{m\times n}$, $m \leq n$, $Q > 0$.

In this case, it is clear that a unique solution exists if A is of full rank and Q is *pdf* on the subspace $M \triangleq \{\mathbf{x} : A\mathbf{x} = \mathbf{0}\}$. (Note that if Q were only positive semidefinite on M, the solution would not be unique.)

The Lagrangian for (9.2) is

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \frac{1}{2}\,\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} + \boldsymbol{\lambda}^T(A\mathbf{x} - \mathbf{b}) \tag{9.3}$$

The stationary point (or KKT point) of (9.3) satisfies the following equations:

$$\left.\begin{array}{lll} \nabla_{\mathbf{x}}\,\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0} & \Rightarrow & Q\mathbf{x} + A^T\boldsymbol{\lambda} + \mathbf{c} = \mathbf{0} \\ \nabla_{\boldsymbol{\lambda}}\,\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0} & \Rightarrow & A\mathbf{x} = \mathbf{b} \end{array}\right\} \tag{9.4}$$

$$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{c} \\ \mathbf{b} \end{bmatrix}$$

Proposition 9.1.1. *Let Q and A be $n \times n$ and $m \times n$ matrices, respectively. Suppose that A has rank m and that Q is pdf on the subspace $M = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{0}\}$. Then the matrix $\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$ is nonsingular.* ■

Proof See [14], pp. 236–237), or [24]. Q.E.D.

From the first equation in (9.4) we have

$$\mathbf{x} = -Q^{-1}A^T\boldsymbol{\lambda} - Q^{-1}\mathbf{c}.$$

Substituting this into the second equation yields

$$\boldsymbol{\lambda} = -(AQ^{-1}A^T)^{-1}\left[AQ^{-1}\mathbf{c} + \mathbf{b}\right],$$

and also

$$\begin{aligned} \mathbf{x} &= Q^{-1}A^T(AQ^{-1}A^T)^{-1}\left[AQ^{-1}\mathbf{c} + \mathbf{b}\right] - Q^{-1}\mathbf{c} \\ &= -Q^{-1}\left[I - A^T(AQ^{-1}A^T)^{-1}AQ^{-1}\right]\mathbf{c} \\ &\quad +Q^{-1}A^T(AQ^{-1}A^T)^{-1}\mathbf{b} \end{aligned}$$

9.2 Active Set Method

Most QP problems involve inequality constraints and can be expressed as

$$\begin{array}{lll} \min & \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} & \\ \text{subject to} & \mathbf{a}_i^T\mathbf{x} = b_i, & i \in \mathcal{E} \\ & \mathbf{a}_i^T\mathbf{x} \leq b_i, & i \in \mathcal{I}. \end{array}$$

They are almost always solved by an active set method.

The idea underlying active set methods is to partition inequality constraints into two groups:

- those that are to be treated as active and
- those that are to be treated as inactive.

And the constraints treated as inactive are essentially ignored.

The index set W_k always includes the equality constraints $\mathcal{E}$ and possibly some of the inequality constraints $\mathcal{I}$.

At iteration k, let the iterate $\mathbf{x}^{(k)}$ be a point that is feasible for all constraints and that satisfies all the equality constraints in the current working set W_k.

By translating to the point $\mathbf{x}^{(k)}$, the quadratic program corresponding to the working set is then defined in the form

$$\left.\begin{array}{lll} \min\limits_{\mathbf{d}_k} & \frac{1}{2}\,\mathbf{d}^{(k)T} Q \mathbf{d}^{(k)} + \mathbf{g}^{(k)T}\mathbf{d}^{(k)} & \\ \text{subject to} & \mathbf{a}_i^T \mathbf{d}^{(k)} = 0, & i \in W_k \end{array}\right\} \tag{9.5}$$

where $\mathbf{g}^{(k)} \triangleq \mathbf{c} + Q\mathbf{x}^{(k)}$.

The program in (9.5) has only equality constraints and therefore can be solved for $\mathbf{d}^{(k)}$ by the previous method or by other numerically efficient methods.

- If $\mathbf{d}^{(k)} = \mathbf{0}$, then the current point $\mathbf{x}^{(k)}$ is optimal with respect to the current working set W_k.
- If $\mathbf{d}^{(k)} \neq \mathbf{0}$ and $\mathbf{x}^{(k)} + \mathbf{d}^{(k)}$ is feasible for all constraints, then $\mathbf{x}^{(k)} + \mathbf{d}^{(k)}$ becomes the new $\mathbf{x}^{(k+1)}$.
- If $\mathbf{x}^{(k)} + \mathbf{d}^{(k)}$ is not feasible, then a search of the form $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$ is made and α_k is selected as large as possible to maintain feasibility. At this point, a new inequality constraint is satisfied by equality, and this constraint is adjoined to the working set W_{k+1}. The general move is therefore given by $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$, where

$$\alpha_k = \min\left[1, \min_{i\,:\,i \notin W_k,\, \mathbf{a}_i^T \mathbf{d}^{(k)} > 0} \left\{ \frac{b_i - \mathbf{a}_i^T \mathbf{x}^{(k)}}{\mathbf{a}_i^T \mathbf{d}^{(k)}} \right\}\right].$$

The process may proceed in this fashion (repeatedly adjoining constraints to the working set), but eventually a point is obtained that is a minimum over the current working set (since there are only a finite number of constraints).

At this point, the corresponding multipliers λ_i for the constraints in W_k are examined.

- If they are nonnegative for all members of $\mathcal{I}$, the current point is optimal.
- If at least one λ_i is negative for $i \in \mathcal{I}$, the one such i (usually the most negative) is dropped from the working set.

From the above, an algorithm for solving QP with inequality constraints can be stated as follows.

Algorithm Start with a feasible point $\mathbf{x}^{(0)}$ and a working set W_0. Set $k = 0$.

1. Solve the equality constrained quadratic program

$$\left.\begin{array}{ll} \min\limits_{\mathbf{d}^{(k)}} & \frac{1}{2}\,\mathbf{d}^{(k)T} Q \mathbf{d}^{(k)} + \mathbf{g}^{(k)T}\mathbf{d}^{(k)} \\ \text{subject to} & \mathbf{a}_i^T \mathbf{d}^{(k)} = 0, \quad i \in W_k \end{array}\right\} \tag{9.6}$$

2. If $\mathbf{d}^{(k)} = \mathbf{0}$, go to step 6.
3. Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$, where

$$\alpha_k = \min\left[1, \min_i \left\{ \frac{b_i - \mathbf{a}_i^T \mathbf{x}^{(k)}}{\mathbf{a}_i^T \mathbf{d}^{(k)}} : i \notin W_k, \mathbf{a}_i^T \mathbf{d}^{(k)} > 0 \right\}\right]. \tag{9.7}$$

4. If $\alpha_k < 1$, then adjoin the minimizing index in (9.7) to W_k to form W_{k+1}.
5. Set $k = k + 1$ and return to step 1.
6. Compute the Lagrange multipliers of (9.6).
7. Set $\lambda_q = \min\limits_{i \in \mathcal{I} \cap W_k} \lambda_i$.
8. If $\lambda_q \geq 0$, then stop and $\mathbf{x}^{(k)}$ is optimal. Otherwise, drop q from W_k to define W_{k+1}.
9. Set $k = k + 1$ and return to step 1.

Example Solve the following quadratic programming problem using the active set method.

$$\left.\begin{array}{ll} \min & 2x_1^2 + x_1x_2 + x_2^2 - 12x_1 - 10x_2 \\ \text{subject to} & x_1 + x_2 \leq 4 \\ & -x_1 \leq 0 \\ & -x_2 \leq 0 \end{array}\right\}. \tag{9.8}$$

Solution

Let $f(\mathbf{x}) \triangleq 2x_1^2 + x_1x_2 + x_2^2 - 12x_1 - 10x_2$ where $\mathbf{x} \triangleq [x_1, x_2]^T$. Then Problem (9.8) is rewritten as

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ \text{subject to} & \mathbf{a}_1^T \mathbf{x} \leq b_1 \\ & \mathbf{a}_2^T \mathbf{x} \leq b_2 \\ & \mathbf{a}_3^T \mathbf{x} \leq b_3 \end{array} \tag{9.9}$$

where

$$\mathbf{a}_1 = [1, 1]^T, \quad \mathbf{a}_2 = [-1, 0]^T, \quad \mathbf{a}_3 = [0, -1]^T, \quad b_1 = 4, \quad b_2 = 0, \quad b_3 = 0,$$

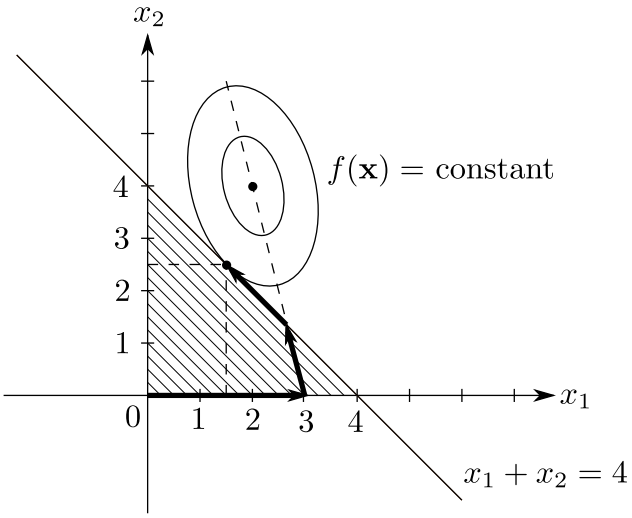

Figure 9.1: Geometric solution of the considered QP

$$Q = \begin{bmatrix} 4 & 1 \\ 1 & 2 \end{bmatrix} \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} -12 \\ -10 \end{bmatrix}.$$

Note that without the constraints in (9.8),

$$\arg\min_{\mathbf{x}\in\mathbb{R}^2} f(\mathbf{x}) = -Q^{-1}\mathbf{c} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}.$$

Choose an initial point $\mathbf{x}^{(0)} = [0, 0]^T$, which satisfies all the constraints in (9.8). Consequently, the second and third constraints are both active while the first one is not. Hence, the initial working index set is $W_0 = \{2, 3\}$.

Iteration 0 $(k = 0)$.
Let $\mathbf{d}^{(0)} = [d_1, d_2]^T$.

$$\text{Solve} \quad \left.\begin{array}{ll} \min\limits_{\mathbf{d}^{(0)}} & \frac{1}{2}\mathbf{d}^{(0)T} Q \mathbf{d}^{(0)} + \mathbf{g}^{(0)T}\mathbf{d}^{(0)} \\ \text{subject to} & \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{array}\right\}. \tag{9.10}$$

The minimizer $[d_1, d_2]^T$ of Problem (9.10) and the associated Lagrange multipliers $(\lambda_2, \lambda_3)^T$ are obtained by solving the following equations.

$$\begin{bmatrix} 4 & 1 & -1 & 0 \\ 1 & 2 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 10 \\ 0 \\ 0 \end{bmatrix}.$$

Therefore, $d_1 = 0, \quad d_2 = 0, \quad \lambda_2 = -12, \quad \text{and} \quad \lambda_3 = -10.$ Ans

It is clear that $\mathbf{d}^{(0)} = [0, 0]^T$ is optimal for Problem (9.10). That is to say, $\mathbf{x}^{(0)} = [0, 0]^T$ is optimal for

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{a}_2^T\mathbf{x} = b_2 \\ & \mathbf{a}_3^T\mathbf{x} = b_3 \end{array}$$

However, $\lambda_2 = -12 < 0$ and $\lambda_3 = -10 < 0$. Since λ_2 is most negative, the second constraint (i.e. index 2) is dropped from the working set and hence $W_1 = \{3\}$.

Set $\mathbf{x}^{(1)} = \mathbf{x}^{(0)}$.

Iteration 1 $(k = 1)$.

Let $\mathbf{d}^{(1)} = [d_1, d_2]^T$. Set $\mathbf{g}^{(1)} = \mathbf{g}^{(0)}$ and $W_1 = \{3\}$.

$$\left.\begin{array}{lll} \text{Solve} & \min\limits_{\mathbf{d}^{(1)}} & \frac{1}{2}\mathbf{d}^{(1)T} Q\mathbf{d}^{(1)} + \mathbf{g}^{(1)T}\mathbf{d}^{(1)} \\ & \text{subject to} & \begin{bmatrix} 0 & -1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = 0 \end{array}\right\}. \tag{9.11}$$

The minimizer $[d_1, d_2]^T$ of Problem (9.11) and the associated Lagrange multiplier λ_3 are obtained by solving the following equations.

$$\begin{bmatrix} 4 & 1 & 0 \\ 1 & 2 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 10 \\ 0 \end{bmatrix}.$$

Therefore, $d_1 = 3, \quad d_2 = 0, \quad \text{and} \quad \lambda_3 = -7.$ Ans

Since $\mathbf{x}^{(1)} + \mathbf{d}^{(1)} = [3, 0]^T$ is feasible, we update $\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \mathbf{d}^{(1)} = [3, 0]^T$ and set $W_2 = W_1$.

Iteration 2 $(k = 2)$.

Let $\mathbf{d}^{(2)} = [d_1, d_2]^T$. Set $\mathbf{g}^{(2)} = \mathbf{c} + Q\mathbf{x}^{(2)} = [0, -7]^T$ and $W_2 = \{3\}$.

$$\left.\begin{array}{lll} \text{Solve} & \min\limits_{\mathbf{d}^{(2)}} & \frac{1}{2}\mathbf{d}^{(2)T} Q\mathbf{d}^{(2)} + \mathbf{g}^{(2)T}\mathbf{d}^{(2)} \\ & \text{subject to} & \begin{bmatrix} 0 & -1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = 0 \end{array}\right\}. \tag{9.12}$$

The minimizer $[d_1, d_2]^T$ of Problem (9.12) and the associated Lagrange multiplier λ_3 are obtained by solving the following equations.

$$\begin{bmatrix} 4 & 1 & 0 \\ 1 & 2 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 7 \\ 0 \end{bmatrix}.$$

Therefore, $d_1 = 0, \quad d_2 = 0, \quad$ and $\quad \lambda_3 = -7.$ Ans

Now it is clear that $\mathbf{x}^{(2)} = [3, 0]^T$ is optimal for

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{a}_3^T\mathbf{x} = 0. \end{array}$$

Since $\lambda_3 = -7 < 0$ and since λ_3 is the multiplier for the third inequality in the original problem, the third constraint (i.e. index 3) is dropped from the working set. Hence, $W_3 = \{\ \}$. Set $\mathbf{x}^{(3)} = \mathbf{x}^{(2)} = [3, 0]^T$.

Iteration 3 $\quad (k = 3)$.
Set $\mathbf{g}^{(3)} = \mathbf{g}^{(2)} = [0, -7]^T$. Note that $W_3 = \{\ \}$ if and only if there is no constraint.

$$\text{Solve} \qquad \min_{\mathbf{d}^{(3)}} \ \frac{1}{2}\mathbf{d}^{(3)T} Q\mathbf{d}^{(3)} + \mathbf{g}^{(3)T}\mathbf{d}^{(3)} \tag{9.13}$$

Therefore, $\quad \mathbf{d}^{(3)} = -Q^{-1}\mathbf{g}^{(3)} = [-1, 4]^T.$

Since $\mathbf{x}^{(3)} + \mathbf{d}^{(3)} = [3 - 1, 0 + 4]^T = [2, 4]^T$ is not feasible for the original constraints in (9.8), $\mathbf{x}^{(4)}$ is updated by $\mathbf{x}^{(4)} = \mathbf{x}^{(3)} + \alpha_3\mathbf{d}^{(3)}$ (so that the first constraint becomes active at $\mathbf{x}^{(4)}$) where

$$\alpha_3 = \frac{b_1 - \mathbf{a}_1^T\mathbf{x}^{(3)}}{\mathbf{a}_1^T\mathbf{d}^{(3)}} = \frac{4 - 3}{3} = \frac{1}{3}. \qquad \text{Ans}$$

Therefore, $\mathbf{x}^{(4)} = [3, 0]^T + \frac{1}{3}[-1, 4]^T = [8/3,\ 4/3]^T.$ Ans
Furthermore, $W_4 = \{1\}$.

Iteration 4 $\quad (k = 4)$.
Let $\mathbf{d}^{(4)} = [d_1, d_2]^T$. Set $\mathbf{g}^{(4)} = \mathbf{c} + Q\mathbf{x}^{(4)} = [0, -14/3]^T \quad$ and $\quad W_4 = \{1\}$.

$$\left.\begin{array}{lll} \text{Solve} & \min\limits_{\mathbf{d}^{(4)}} & \frac{1}{2}\mathbf{d}^{(4)T} Q\mathbf{d}^{(4)} + \mathbf{g}^{(4)T}\mathbf{d}^{(4)} \\ & \text{subject to} & \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = 0 \end{array}\right\}. \tag{9.14}$$

The minimizer $[d_1, d_2]^T$ of Problem (9.14) and the associated Lagrange multiplier λ_1 are obtained by solving the following equations.

$$\begin{bmatrix} 4 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \lambda_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 14/3 \\ 0 \end{bmatrix}.$$

Therefore, $\quad d_1 = -7/6, \quad d_2 = 7/6, \quad$ and $\quad \lambda_1 = 7/2.$ Ans

Since $\mathbf{x}^{(4)} + \mathbf{d}^{(4)} = [8/3 - 7/6,\, 4/3 + 7/6]^T = [3/2,\, 5/2]^T$ is feasible, update $\mathbf{x}^{(5)} = \mathbf{x}^{(4)} + \mathbf{d}^{(4)} = [3/2,\, 5/2]^T$ and set $W_5 = W_4$.

Iteration 5 $(k = 5)$.
Let $\mathbf{d}^{(5)} = [d_1, d_2]^T$. Set $\mathbf{g}^{(5)} = \mathbf{c} + Q\mathbf{x}^{(5)} = [-7/2, -7/2]^T$ and $W_5 = \{1\}$.

$$\begin{aligned} \text{Solve} \qquad & \min_{\mathbf{d}^{(5)}} \quad \tfrac{1}{2}\mathbf{d}^{(5)T} Q \mathbf{d}^{(5)} + \mathbf{g}^{(5)T}\mathbf{d}^{(5)} \\ & \text{subject to} \quad \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = 0. \end{aligned} \tag{9.15}$$

The minimizer $[d_1, d_2]^T$ of Problem (9.15) and the associated Lagrange multiplier λ_1 are obtained by solving the following equations.

$$\begin{bmatrix} 4 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \lambda_1 \end{bmatrix} = \begin{bmatrix} 7/2 \\ 7/2 \\ 0 \end{bmatrix}.$$

Therefore, $d_1 = 0$, $d_2 = 0$, and $\lambda_1 = 7/2$. Ans
Hence, $\mathbf{x}^{(5)} = \mathbf{x}^{(4)} = [3/2, 5/2]^T$.

It is clear that $\mathbf{x}^{(5)}$ is optimal for

$$\begin{aligned} & \min_{\mathbf{x}} \quad \tfrac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T\mathbf{x} \\ & \text{subject to} \quad \mathbf{a}_1^T\mathbf{x} = b_1. \end{aligned}$$

Since λ_1 is now positive, it follows that $\mathbf{x}^{(5)} = [3/2,\, 5/2]^T$ and $\mu_1 = \lambda_1 = 7/2$, $\mu_2 = 0$, $\mu_3 = 0$ are the KKT points of the original problem.

From the above, we have $\mathbf{x}^{(5)} = [3/2,\, 5/2]^T$ is the minimizer of the original QP (9.8). Ans

9.3 Exercises

9.1 Solve the following quadratic programming problem using the active set method.

$$\begin{aligned} \min \quad & x_1^2 + 2x_2^2 - 2x_1 - 6x_2 - 2x_1x_2 \\ \text{subject to} \quad & -x_1 + 2x_2 \leq 2 \\ & x_1 + x_2 \leq 2 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{aligned}$$

Choose the starting point at $x_1 = 0$ and $x_2 = 0$.

9.2 Solve the following convex quadratic program by the active set method.

$$\begin{aligned} \text{minimize} \quad & 2x_1^2 + x_1x_2 + x_2^2 - 12x_1 - 10x_2 \\ \text{subject to} \quad & 2x_1 + x_2 \leq 4, \\ & -x_1 \leq 0, \quad -x_2 \leq 0. \end{aligned}$$

Start at the initial point $x_0 = [0, 1]^T$.

References

[1] M. Al-Baali, Descent property and global convergence of the Fletcher-Reeves method with inexact line search, *IMA Journal of Numerical Analysis*, vol. 5, pp. 121–124, 1985.

[2] R. Baldick, *Applied Optimization: Formulation and Algorithms for Engineering Systems*, Cambridge University Press, Canabridge, 2006.

[3] M. Bartholomew-Biggs, *Nonlinear Optimization with Engineering Applications*, Springer Verlag, New York, 2008.

[4] C. G. Broyden, The convergence of a class of double-rank minimization algorithms Parts I & II, *IMA Journal of Applied Mathematics*, vol. 6, pp. 76–90 & 222-231, 1970.

[5] C. G. Broyden, J. E. Dennis and J. J. Moré, On the local and superlinear convergence of quasi-Newton methods, *IMA Journal of Applied Mathematics*, vol. 12, pp. 223–245, 1973.

[6] R. H. Byrd, J. Nocedal and Y.-X. Yuan, Global convergence of a class of quasi-Newton methods on convex problems, *SIAM Journal on Numerical Analysis*, vol. 24, pp. 1171–1190, 1987.

[7] W. Cheney & D. Kincaid, *Numerical Mathematics and Computing*, 6th Edition, Thompson Brooks/Cole, Belmont, 2008.

[8] E. K. P. Chong & S. H. Zak, *An Introduction to Optimization*, 4th Edition, John Wiley & Sons, Chichester, 2013.

[9] A. Cohen, Rate of convergence of several conjugate gradient algorithms, *SIAM Journal on Numerical Analysis*, vol. 9, pp. 248–259, 1972.

[10] W. C. Davidon, *Variable metric method of minimization*, Technical Report No. ANL-5990, Argonne National Laboratory, Illinois, U.S.A, 1959.

[11] W. C. Davidon, Variable metric method of minimization, *SIAM Journal on Optimization*, vol. 1, pp. 1–17, 1991.

[12] J. E. Dennis and J. J. Moré, Quasi-Newton methods, motivation and theory, *SIAM Review*, vol. 19, pp.46–89, 1977.

[13] R. Fletcher, A new approach to variable metric algorithms, *Computer Journal*, vol. 13, pp. 317–322, 1970.

[14] R. Fletcher, *Practical Methods of Optimization*, 2nd Edition, John Wiley & Sons, Chichester, 1987.

[15] R. Fletcher and M. J. D. Powell, A rapidly convergent descent method for minimization, *Computer Journal*, vol. 6, pp. 163–168, 1963.

[16] R. Fletcher and C. Reeves, Function minimization by conjugate gradients, *Computer Journal*, vol. 7, pp. 149–154, 1964.

[17] R. L. Fox, *Optimization Methods for Engineering Design*, Addison-Wesley, Reading, Mass, 1971.

[18] D. Goldfarb, A family of variable metric methods derived by variational means, *Mathematics of Computation*, vol. 24, pp. 23–26, 1970.

[19] G. H. Golub & C. F. van Loan, *Matrix Computation*, 3rd edition, John Hopkins, 1996.

[20] W. W. Hager, *Applied Numerical Linear Algebra*, Prentice-Hall, 1988.

[21] M. R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 449–464, 1952.

[22] W. Kaplan, *Advanced Calculus*, 5th edition, Addison-Wesley, 2003.

[23] K. Levenberg, A method for the solution of certain problems in least squares, *Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, 1944.

[24] D. G. Luenberger & Y. Ye, *Linear and Nonlinear Programming*, 3rd Edition, Springer Verlag, New York, 2003.

[25] D. W. Marquardt, An algorithm for least-square estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, vol. 11, pp. 431–441, 1963.

[26] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*, McGraw-Hill, New York, 1996.

[27] W, K. Nicholson, *Linear Algebra with Applications*, McGraw-Hill, New York, 2013.

[28] B. Noble & J. W. Daniel, *Applied Linear Algebra*, Prentice-Hall, 1988.

[29] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd edition, Springer Verlag, New York, 2006.

[30] E. Polak and G. Ribiére, Note sur la convergence de methodes de directions conjugées, *Rev. Française Informat Recherche Operationelle*, vol. 16, pp. 35–43, 1969.

[31] M. J. D. Powell, An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, vol. 7, pp. 155–162, 1964.

[32] M. J. D. Powell, On the convergence of the variable metric method, *IMA Journal of Applied Mathematics*, vol. 7, pp. 21–36, 1971.

[33] M. J. D. Powell, Some global convergence properties of a variable metric algorithm for minimization without exact line searches, in *SIAM-AMS Proceedings, Vol. IX* (Eds: R.W. Cottle and C.E. Lemke), SIAM Publications, Philadelphia, 1976.

[34] M. J. D. Powell, Restart procedures of the conjugate gradient method, *Mathematical Programming*, vol. 12, pp. 241–254, 1977.

[35] G. V. Reklaitis, A. Ravindran and K. M. Ragsdell, *Engineering Optimization: Methods and Applications*, John Wiley & Sons, New York, 1983.

[36] S. S. Rao, *Optimization: Theory and Applications*, 2nd Edition, Wiley Eastern, New Delhi, 1984.

[37] D. F. Shanno, Conditioning of quasi-Newton methods for function minimization, *Mathematics of Computation*, vol. 24, pp. 647–656, 1970.

[38] G. Strang, *Linear Algebra and Its Applications* 4th edition, Brooks/Cole, 2006.

Index

LDL^T factorization, 48
Q conjugate, 51

Armijo condition, 59

Biconditional statement, 5
Bisection method, 16
Broyden class, 58

Cauchy-Schwarz inequality, 7
Concave function, 30
Conditional statement, 4
Conjugate direction methods, 51
Conjugate gradient method, 61
Conjugate gradient methods, 61
    Fletcher-Reeves formula, 65
    Hestenes and Stiefel formula, 64
    inexact line search, 65
    Polak-Ribiére formula, 64
Constrained optimization
    equality constraints, 76
Convergence rate
    linear, 21
    quadratic, 21
    superlinear, 22
Convex function, 30
Convex quadratic programs, 116
    active set method, 117
    equality constraints, 116
    stationary point, 116
Convex set, 30

Directional derivative, 31

Feasible direction, 89
    linear inequalitty constraints, 91

Global minimizer, 29
    strict, 29

Hölder inequality, 7

Inequality constraints
    active, 82
    inactive, 82

Karush–Kuhn–Tucker (KKT)
    condition, 83
    multiplier, 84

Lagrange multiplier, 78
Line search
    cubic interpolation method, 39
    golden section method, 40
    quadractic interpolation method, 38
Linear programs
    artificial variable, 105
    basic feasible solution, 100
    basic solution, 100
    big M method, 105, 110
    extreme point, 99
    phase-I problem, 106
    phase-II problem, 106
    standard form, 98
    two-phase method, 105, 106
Local minimizer, 29
    strict, 29

Mathematical induction, 5
Matrix, 7
    eigenvalues, 8
    eigenvector, 8
    indefinite, 9
    negative definite, 9
    negative semidefinite, 9
    positive definite, 9
    positive semidefinite, 9
    rank, 8
Mean value theorem, 12

Newton's method, 45
    classical, 45
    modified, 47
Newton-Raphson
    multivariable, 24
    single variable, 18

Quadractic function, 42
Quadratic programs, 116
Quasi-Newton condition, 54
Quasi-Newton methods, 53
    BFGS formula, 55
    Broyden class, 58
    DFP formula, 55
    inexact line search, 59

Regular point, 76

Secant condition, 54
Secant method, 21
Sherman-Morrison formula, 55
Simplex
    tableau, 103
Simplex method, 103
Steepest descent method, 43

Tangent plane, 77
Taylor series, 12
Triangle inequality, 6

Variable metric methods, 55
Vectors, 6
    linear independence, 7
    norms, 6

Wolfe condition, 59