Chapter 1

INTRODUCTION

1.1 What is mathematical optimization?

Formally, *Mathematical Optimization* is the process of

(i) the *formulation* and

(ii) the *solution* of a constrained optimization problem of the general mathematical form:

$$\underset{\text{w.r.t. } \mathbf{x}}{\text{minimize}}\ f(\mathbf{x}),\ \mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$$

subject to the constraints:

$$\begin{aligned} g_j(\mathbf{x}) &\leq 0, \quad j = 1,\ 2,\ \dots,\ m \\ h_j(\mathbf{x}) &= 0, \quad j = 1,\ 2,\ \dots,\ r \end{aligned} \tag{1.1}$$

where $f(\mathbf{x})$, $g_j(\mathbf{x})$ and $h_j(\mathbf{x})$ are scalar functions of the real *column vector* $\mathbf{x}$.

The continuous components x_i of $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ are called the (*design*) *variables*, $f(\mathbf{x})$ is the *objective function*, $g_j(\mathbf{x})$ denotes the respective *inequality constraint functions* and $h_j(\mathbf{x})$ the *equality constraint functions.*

The optimum vector $\mathbf{x}$ that solves problem (1.1) is denoted by $\mathbf{x}^*$ with corresponding optimum function value $f(\mathbf{x}^*)$. If no constraints are specified, the problem is called an *unconstrained* minimization problem.

Mathematical Optimization is often also called *Nonlinear Programming, Mathematical Programming* or *Numerical Optimization.* In more general terms Mathematical Optimization may be described as the science of determining the *best* solutions to mathematically defined problems, which may be models of physical reality or of manufacturing and management systems. In the first case solutions are sought that often correspond to minimum energy configurations of general structures, from molecules to suspension bridges, and are therefore of interest to Science and Engineering. In the second case commercial and financial considerations of economic importance to Society and Industry come into play, and it is required to make decisions that will ensure, for example, maximum profit or minimum cost.

The history of the Mathematical Optimization, where functions of many variables are considered, is relatively short, spanning roughly only 70 years. At the end of the 1940s the very important simplex method for solving the special class of linear programming problems was developed. Since then numerous methods for solving the general optimization problem (1.1) have been developed, tested, and successfully applied to many important problems of scientific and economic interest. There is no doubt that the advent of the computer was essential for the development of these optimization methods. However, in spite of the proliferation of optimization methods, there is no universal method for solving all optimization problems. According to Nocedal and Wright (1999): "...there are numerous algorithms, each of which is tailored to a particular type of optimization problem. It is often the user's responsibility to choose an algorithm that is appropriate for the specific application. This choice is an important one; it may determine whether the problem is solved rapidly or slowly and, indeed, whether the solution is found at all." In a similar vein Vanderplaats (1998) states that "The author of each algorithm usually has numerical examples which demonstrate the efficiency and accuracy of the method, and the unsuspecting practitioner will often invest a great deal of time and effort in programming an algorithm, only to find that it will not in fact solve the particular problem being attempted. This often leads to disenchantment with these techniques that can be avoided if the user is knowledgeable in the basic concepts of numerical optimization." With these representative and authoritative opinions in mind, and also taking into account the present authors' personal experiences in developing algorithms and applying them to design problems in mechanics, this text has been written to provide a brief but unified introduction to optimization concepts and methods. In addition, an overview of a set of novel algorithms, developed by the authors and their students at the University of Pretoria over the past thirty years, is also given.

The emphasis of this book is almost exclusively on gradient-based methods. This is for two reasons. (i) The authors believe that the introduction to the topic of mathematical optimization is best done via the classical gradient-based approach and (ii), contrary to the current popular trend of using non-gradient methods, such as genetic algorithms (GA's), simulated annealing, particle swarm optimization and other evolutionary methods, the authors are of the opinion that these search methods are, in many cases, computationally too expensive to be viable. The argument that the presence of numerical noise and multiple minima disqualify the use of gradient-based methods, and that the only way out in such cases is the use of the above mentioned non-gradient search techniques, is not necessarily true. It is the experience of the authors that, through the judicious use of gradient-based methods, problems with numerical noise and multiple minima may be solved, and at a fraction of the computational cost of search techniques such as genetic algorithms. In this context Chapter 6, dealing with the new gradient-based methods developed by the first author and gradient-only methods developed by the authors in Chapter 8, are especially important. The presentation of the material is not overly rigorous, but hopefully correct, and should provide the necessary information to allow scientists and engineers to select appropriate optimization algorithms and to apply them successfully to their respective fields of interest.

Many excellent and more comprehensive texts on practical optimization can be found in the literature. In particular the authors wish to acknowledge the works of Wismer and Chattergy (1978), Chvatel (1983), Fletcher (1987), Bazaraa et al. (1993), Arora (1989), Haftka and Gürdal (1992), Rao (1996), Vanderplaats (1998), Nocedal and Wright (1999) and Papalambros and Wilde (2000).

1.2 Objective and constraint functions

The values of the functions $f(\mathbf{x})$, $g_j(\mathbf{x})$ and $h_j(\mathbf{x})$ at any point $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, may in practice be obtained in different ways:

(i) from *analytically* known *formulae*, e.g. $f(\mathbf{x}) = x_1^2 + 2x_2^2 + \sin x_3$;

(ii) as the *outcome* of some complicated *computational process*, e.g. $g_1(\mathbf{x}) = a(\mathbf{x}) - a_{\max}$, where $a(\mathbf{x})$ is the stress, computed by means of a finite element analysis, at some point in a structure, the design of which is specified by $\mathbf{x}$; or

(iii) from *measurements* taken of a *physical process*, e.g. $h_1(\mathbf{x}) = T(\mathbf{x}) - T_0$, where $T(\mathbf{x})$ is the temperature measured at some specified point in a reactor, and $\mathbf{x}$ is the vector of operational settings.

The first two ways of function evaluation are by far the most common. The optimization principles that apply in these cases, where computed function values are used, may be carried over directly to also be applicable to the case where the function values are obtained through physical measurements.

Much progress has been made with respect to methods for solving different classes of the general problem (1.1). Sometimes the solution may be obtained *analytically*, i.e. a closed-form solution in terms of a *formula* is obtained.

In general, especially for $n > 2$, solutions are usually obtained *numerically* by means of suitable *algorithms* (computational recipes).

Expertise in the *formulation* of appropriate optimization problems of the form (1.1), through which an optimum decision can be made, is gained from *experience*. This exercise also forms part of what is generally known as the *mathematical modelling* process. In brief, attempting to solve real-world problems via mathematical modelling requires the cyclic performance of the four steps depicted in Figure 1.1. The main steps are: 1) the observation and study of the real-world situation associated with a practical problem, 2) the abstraction of the problem by the construction of a mathematical model, that is described in terms of

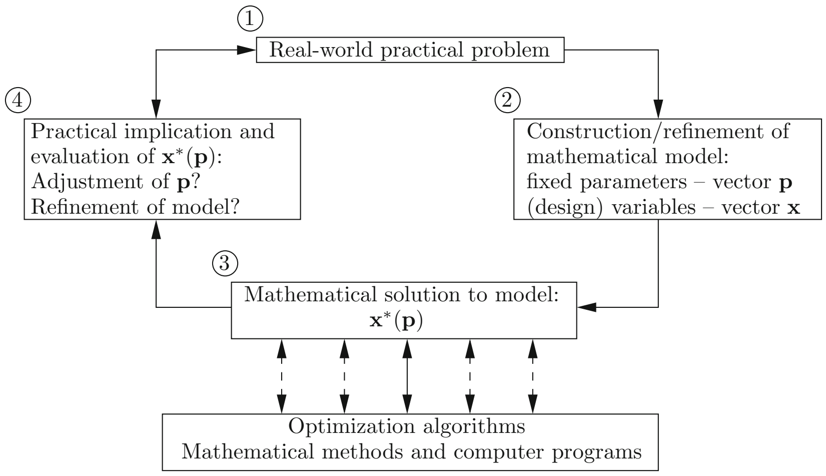
①
Real-world practical problem
②
Construction/refinement of mathematical model:
fixed parameters – vector **p**
(design) variables – vector **x**
③
Mathematical solution to model: $\mathbf{x}^*(\mathbf{p})$
④
Practical implication and evaluation of $\mathbf{x}^*(\mathbf{p})$:
Adjustment of **p**?
Refinement of model?
Optimization algorithms
Mathematical methods and computer programs

Figure 1.1: The mathematical modelling process

preliminary fixed model parameters $\mathbf{p}$, and variables $\mathbf{x}$, the latter to be determined such that model performs in an acceptable manner, 3) the solution of a resulting purely mathematical problem, that requires an analytical or numerical parameter dependent solution $\mathbf{x}^*(\mathbf{p})$, and 4) the evaluation of the solution $\mathbf{x}^*(\mathbf{p})$ and its practical implications. After step 4) it may be necessary to adjust the parameters and refine the model, which will result in a new mathematical problem to be solved and evaluated. It may be required to perform the modelling cycle a number of times, before an acceptable solution is obtained. More often than not, the mathematical problem to be solved in 3) is a *mathematical optimization problem*, requiring a numerical solution. The *formulation* of an appropriate and consistent optimization problem (or model) is probably the most important, but unfortunately, also the *most neglected* part of Practical Mathematical Optimization.

This book gives a very brief introduction to the *formulation* of optimization problems, and deals with different *optimization algorithms* in greater depth. Since no algorithm is generally applicable to all classes of problems, the emphasis is on providing sufficient information to allow for the selection of appropriate algorithms or methods for different specific problems.

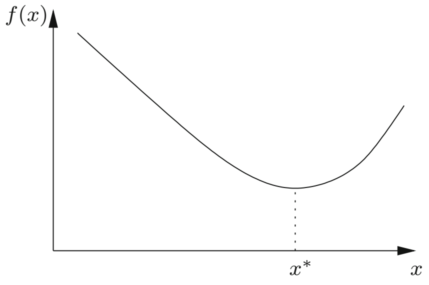

Figure 1.2: Function of single variable with optimum at x^*

1.3 Basic optimization concepts

1.3.1 Simplest class of problems: Unconstrained one-dimensional minimization

Consider the minimization of a smooth, i.e. continuous and twice continuously differentiable (C^2) function of a single real variable, i.e. the problem:

$$\underset{x}{\text{minimize}}\, f(x),\ x \in \mathbb{R},\ f \in C^2. \tag{1.2}$$

With reference to Figure 1.2, for a strong local minimum, it is required to determine a x^* such that $f(x^*) < f(x)$ for all x.

Clearly x^* occurs where the slope is zero, i.e. where

$$f'(x) = \frac{df(x)}{dx} = 0,$$

which corresponds to the first order necessary condition. In addition *non-negative curvature* is necessary at x^*, i.e. it is required that the second order condition

$$f''(x) = \frac{d^2 f(x)}{dx^2} > 0$$

must hold at x^* for a strong local minimum.

A simple *special case* is where $f(x)$ has the simple *quadratic form*:

$$f(x) = ax^2 + bx + c. \tag{1.3}$$

Since the minimum occurs where $f'(x) = 0$, it follows that the closed-form solution is given by

$$x^* = -\frac{b}{2a}, \text{ provided } f''(x^*) = 2a > 0. \tag{1.4}$$

If $f(x)$ has a *more general form*, then a closed-form solution is in general not possible. In this case, the solution may be obtained numerically via the *Newton-Raphson algorithm*:

Given an approximation x^0, iteratively compute:

$$x^{i+1} = x^i - \frac{f'(x^i)}{f''(x^i)};\ i = 0,\ 1,\ 2,\ \ldots \tag{1.5}$$

Hopefully $\lim_{i\to\infty} x^i = x^*$, i.e. the iterations converge, in which case a sufficiently accurate numerical solution is obtained after a finite number of iterations.

1.3.2 Contour representation of a function of two variables ($n = 2$)

Consider a function $f(\mathbf{x})$ of two variables, $\mathbf{x} = [x_1, x_2]^T$. The locus of all points satisfying $f(\mathbf{x}) = c =$ constant, forms a contour in the $x_1 - x_2$ plane. For each value of c there is a corresponding different contour.

Figure 1.3 depicts the contour representation for the example $f(\mathbf{x}) = x_1^2 + 2x_2^2$.

In three dimensions ($n = 3$), the contours are *surfaces of constant function* value. In more than three dimensions ($n > 3$) the contours are, of course, impossible to visualize. *Nevertheless, the contour representation in two-dimensional space will be used throughout the discussion of optimization techniques to help visualize the various optimization concepts.*

Other examples of 2-dimensional objective function contours are shown in Figures 1.4 to 1.6.

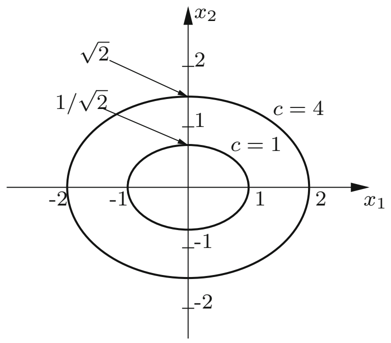

Figure 1.3: Contour representation of the function $f(\mathbf{x}) = x_1^2 + 2x_2^2$

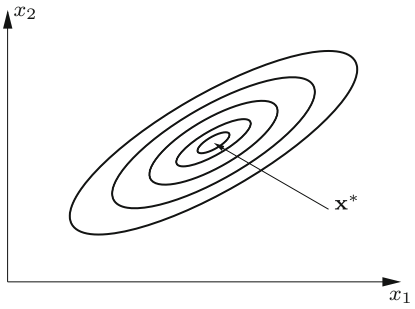

Figure 1.4: General quadratic function

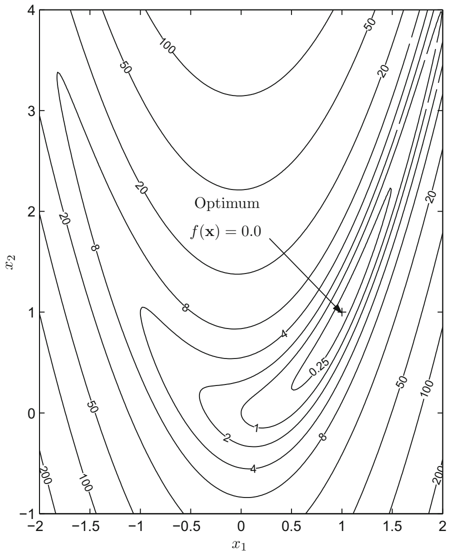

Figure 1.5: The two-dimensional Rosenbrock function $f(\mathbf{x}) = 10(x_2 - x_1^2)^2 + (1 - x_1)^2$

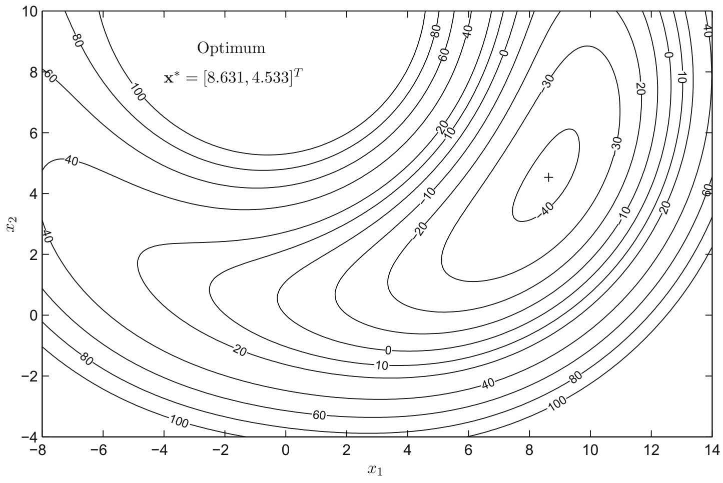

Figure 1.6: Potential energy function of a spring-force system (Vanderplaats 1998)

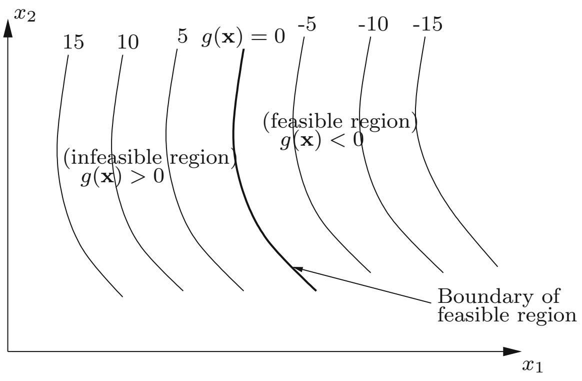

Figure 1.7: Contours within feasible and infeasible regions

1.3.3 Contour representation of constraint functions

1.3.3.1 Inequality constraint function $g(\mathbf{x})$

The contours of a typical inequality constraint function $g(\mathbf{x})$, in $g(\mathbf{x}) \leq 0$, are shown in Figure 1.7. The contour $g(\mathbf{x}) = 0$ divides the plane into a *feasible region* and an *infeasible region.*

More generally, the boundary is a surface in three dimensions and a so-called "hyper-surface" if $n > 3$, which of course cannot be visualised.

1.3.3.2 Equality constraint function $h(\mathbf{x})$

Here, as shown in Figure 1.8, only the line $h(\mathbf{x}) = 0$ is a feasible contour.

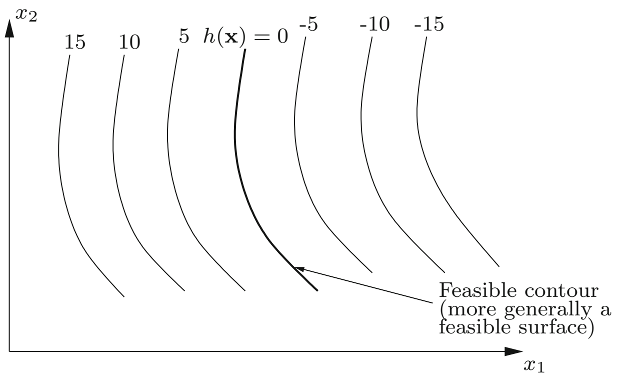

Figure 1.8: Feasible contour of equality constraint

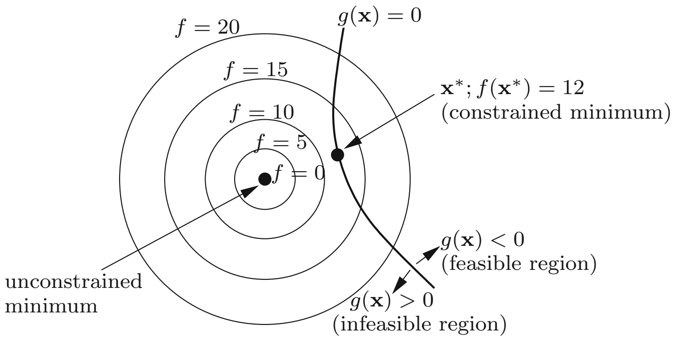
$f = 20$
$f = 15$
$f = 10$
$f = 5$
$f = 0$
$g(\mathbf{x}) = 0$
$\mathbf{x}^*; f(\mathbf{x}^*) = 12$ (constrained minimum)
$g(\mathbf{x}) < 0$ (feasible region)
$g(\mathbf{x}) > 0$ (infeasible region)
unconstrained minimum

Figure 1.9: Contour representation of inequality constrained problem

1.3.4 Contour representations of constrained optimization problems

1.3.4.1 Representation of inequality constrained problem

Figure 1.9 graphically depicts the inequality constrained problem:

$$\begin{gathered} \min f(\mathbf{x}) \\ \text{such that}\, g(\mathbf{x}) \leq 0. \end{gathered}$$

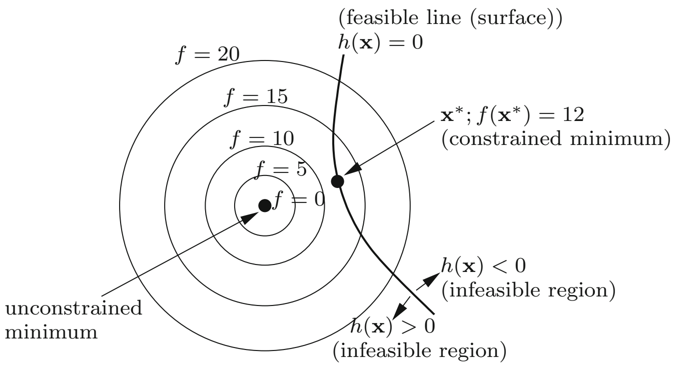

Figure 1.10: Contour representation of equality constrained problem

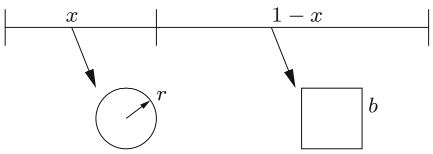

Figure 1.11: Wire divided into two pieces with $x_1 = x$ and $x_2 = 1 - x$

1.3.4.2 Representation of equality constrained problem

Figure 1.10 graphically depicts the equality constrained problem:

$$\begin{aligned} &\min f(\mathbf{x}) \\ &\text{such that } h(\mathbf{x}) = 0. \end{aligned}$$

1.3.5 Simple example illustrating the formulation and solution of an optimization problem

Problem: A length of wire 1 meter long is to be divided into two pieces, one in a circular shape and the other into a square as shown in Figure 1.11. What must the individual lengths be so that the total area is a minimum?

Formulation 1

Set length of first piece $= x$, then the area is given by $f(x) = \pi r^2 + b^2$. Since $r = \frac{x}{2\pi}$ and $b = \frac{1-x}{4}$ it follows that

$$f(x) = \pi \left(\frac{x^2}{4\pi^2} \right) + \frac{(1-x)^2}{16}.$$

The problem therefore reduces to an unconstrained minimization problem:

$$\text{minimize } f(x) = 0.1421x^2 - 0.125x + 0.0625.$$

Solution of Formulation 1

The function $f(x)$ is quadratic, therefore an analytical solution is given by the formula $x^* = -\frac{b}{2a}$ $(a > 0)$:

$$x^* = -\frac{-0.125}{2(0.1421)} = 0.4398 \text{ m},$$

and

$$1 - x^* = 0.5602 \text{ m with } f(x^*) = 0.0350 \text{ m}^2.$$

Formulation 2

Divide the wire into respective lengths x_1 and x_2 $(x_1 + x_2 = 1)$. The area is now given by

$$f(\mathbf{x}) = \pi r^2 + b^2 = \pi \left(\frac{x_1^2}{4\pi^2} \right) + \left(\frac{x_2}{4} \right)^2 = 0.0796x_1^2 + 0.0625x_2^2.$$

Here the problem reduces to an *equality constrained* problem:

$$\begin{aligned} \text{minimize } f(\mathbf{x}) &= 0.0796x_1^2 + 0.0625x_2^2 \\ \text{such that } h(\mathbf{x}) &= x_1 + x_2 - 1 = 0. \end{aligned}$$

Solution of Formulation 2

This constrained formulated problem is more difficult to solve. The closed-form analytical solution is not obvious and special constrained optimization techniques, such as the *method of Lagrange multipliers* to be discussed later, must be applied to solve the constrained problem analytically. The graphical solution is sketched in Figure 1.12.

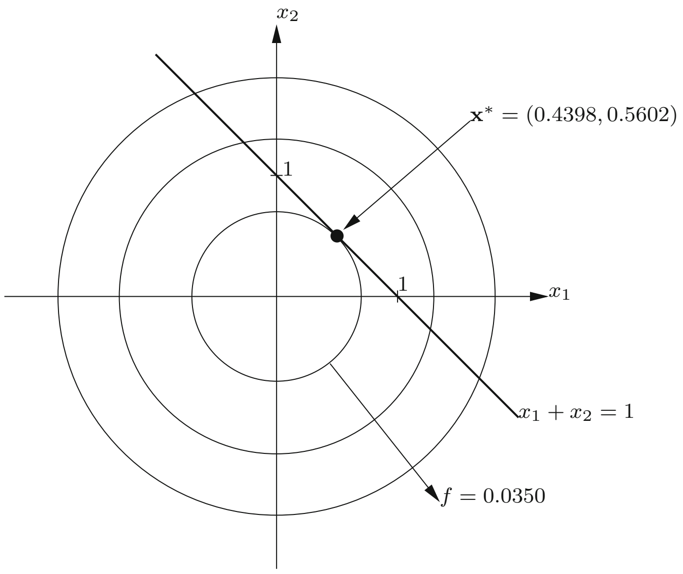

Figure 1.12: Graphical solution of Formulation 2

1.3.6 Maximization

The maximization problem: $\max_{\mathbf{x}} f(\mathbf{x})$ can be cast in the standard form (1.1) by observing that $\max_{\mathbf{x}} f(\mathbf{x}) = -\min_{\mathbf{x}}\{-f(\mathbf{x})\}$ as shown in Figure 1.13. Therefore in applying a minimization algorithm set $F(\mathbf{x}) = -f(\mathbf{x})$.

Also if the inequality constraints are given in the non-standard form: $g_j(\mathbf{x}) \geq 0$, then set $\tilde{g}_j(\mathbf{x}) = -g_j(\mathbf{x})$. In standard form the problem then becomes:

$$\text{minimize } F(\mathbf{x}) \text{ such that } \tilde{g}_j(\mathbf{x}) \leq 0.$$

Once the minimizer $\mathbf{x}^*$ is obtained, the maximum value of the original maximization problem is given by $-F(\mathbf{x}^*)$.

1.3.7 The special case of Linear Programming

A very important special class of the general optimization problem arises when both the objective function and all the constraints are linear func-

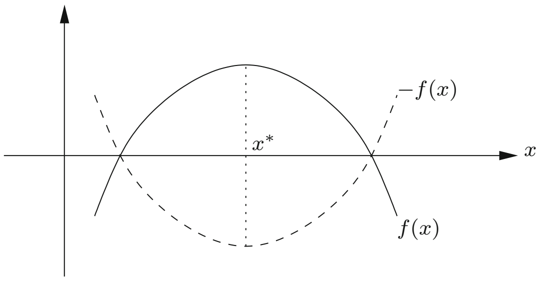

Figure 1.13: Maximization problem transformed to minimization problem

tions of $\mathbf{x}$. This is called a *Linear Programming* problem and is usually stated in the following form:

$$\begin{aligned} &\min_{\mathbf{x}} f(\mathbf{x}) = \mathbf{c}^T\mathbf{x} \\ &\text{such that} \\ &\mathbf{Ax} \leq \mathbf{b};\ \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{1.6}$$

where $\mathbf{c}$ is a real n-vector and $\mathbf{b}$ is a real m-vector, and $\mathbf{A}$ is a $m \times n$ real matrix. A linear programming problem in two variables is graphically depicted in Figure 1.14.

Special methods have been developed for solving linear programming problems. Of these the most famous are the simplex method proposed by Dantzig in 1947 (Dantzig 1963) and the interior-point method (Karmarkar 1984). A short introduction to the simplex method, according to Chvatel (1983), is given in Appendix A.

1.3.8 Scaling of design variables

In formulating mathematical optimization problems, great care must be taken to ensure that the scale of the variables are more or less of the same order. If not, the formulated problem may be relatively insensitive to the variations in one or more of the variables, and any optimization algorithm will struggle to converge to the true solution, because of extreme distortion of the objective function contours as result of the poor scaling. In particular it may lead to difficulties when selecting step lengths and

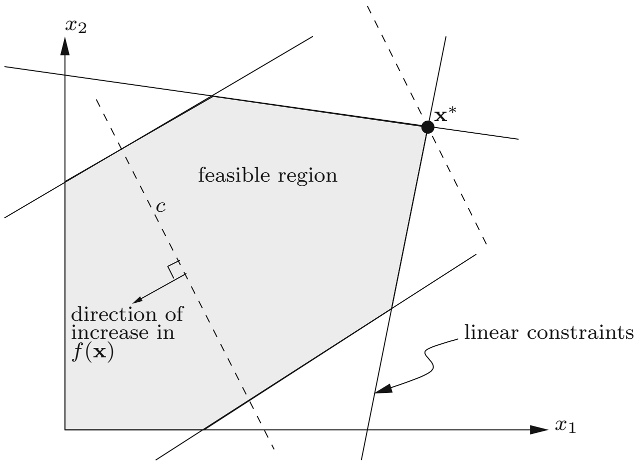

Figure 1.14: Graphical representation of a two-dimensional linear programming problem

calculating numerical gradients. Scaling difficulties often occur where the variables are of different dimension and expressed in different units. Hence it is good practice, if the variable ranges are very large, to scale the variables so that all the variables will be dimensionless and vary between 0 and 1 approximately. For scaling the variables, it is necessary to establish an approximate range for each of the variables. For this, take some estimates (based on judgement and experience) for the lower and upper limits. The values of the bounds are not critical. Another related matter is the scaling or normalization of constraint functions. This becomes necessary whenever the values of the constraint functions differ by large magnitudes.

1.4 Further mathematical concepts

1.4.1 Convexity

A line through the points $\mathbf{x}^1$ and $\mathbf{x}^2$ in $\mathbb{R}^n$ is the set

$$L = \{\mathbf{x} \big| \mathbf{x} = \mathbf{x}^1 + \lambda(\mathbf{x}^2 - \mathbf{x}^1), \text{ for all } \lambda \in \mathbb{R}\}. \tag{1.7}$$

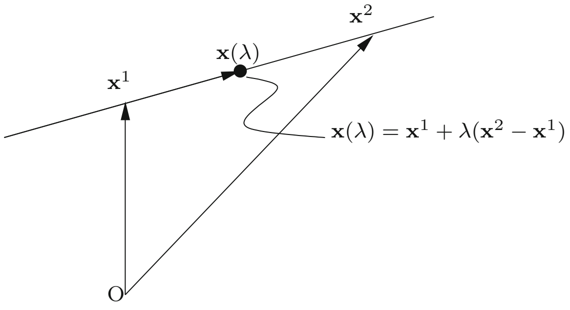
$\mathbf{x}^2$
$\mathbf{x}(\lambda)$
$\mathbf{x}^1$
$\mathbf{x}(\lambda) = \mathbf{x}^1 + \lambda(\mathbf{x}^2 - \mathbf{x}^1)$
O

Figure 1.15: Representation of a point on the straight line through $\mathbf{x}^1$ and $\mathbf{x}^2$

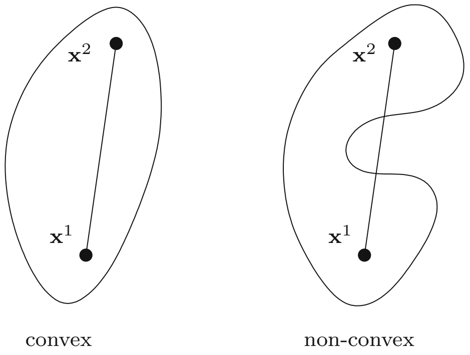

Figure 1.16: Examples of a convex and a non-convex set

Equivalently for any point $\mathbf{x}$ on the line there exists a λ such that $\mathbf{x}$ may be specified by $\mathbf{x} = \mathbf{x}(\lambda) = \lambda\mathbf{x}^2 + (1-\lambda)\mathbf{x}^1$ as shown in Figure 1.15.

1.4.1.1 Convex sets

A set X is convex if for all $\mathbf{x}^1,\ \mathbf{x}^2 \in X$ it follows that

$$\mathbf{x} = \lambda\mathbf{x}^2 + (1-\lambda)\mathbf{x}^1 \in X \text{ for all } 0 \leq \lambda \leq 1.$$

If this condition does not hold the set is non-convex (see Figure 1.16).

1.4.1.2 Convex functions

Given two points $\mathbf{x}^1$ and $\mathbf{x}^2$ in $\mathbb{R}^n$, then any point $\mathbf{x}$ on the straight line connecting them (see Figure 1.15) is given by

$$\mathbf{x} = \mathbf{x}(\lambda) = \mathbf{x}^1 + \lambda(\mathbf{x}^2 - \mathbf{x}^1),\ 0 < \lambda < 1. \tag{1.8}$$

A function $f(\mathbf{x})$ is a *convex function* over a *convex set* X if for all $\mathbf{x}^1$, $\mathbf{x}^2$ in X and for all $\lambda \in [0,1]$:

$$f(\lambda\mathbf{x}^2 + (1-\lambda)\mathbf{x}^1) \leq \lambda f(\mathbf{x}^2) + (1-\lambda)f(\mathbf{x}^1). \tag{1.9}$$

The function is strictly convex if $<$ applies. Concave functions are similarly defined.

Consider again the line connecting $\mathbf{x}^1$ and $\mathbf{x}^2$. Along this line, the function $f(\mathbf{x})$ is a function of the single variable λ:

$$F(\lambda) = f(\mathbf{x}(\lambda)) = f(\mathbf{x}^1 + \lambda(\mathbf{x}^2 - \mathbf{x}^1)). \tag{1.10}$$

This is equivalent to $F(\lambda) = f(\lambda\mathbf{x}^2 + (1-\lambda)\mathbf{x}^1)$, with $F(0) = f(\mathbf{x}^1)$ and $F(1) = f(\mathbf{x}^2)$. Therefore (1.9) may be written as

$$F(\lambda) \leq \lambda F(1) + (1-\lambda)F(0) = F_{int}$$

where F_{int} is the linearly interpolated value of F at λ as shown in Figure 1.17.

Graphically $f(\mathbf{x})$ is *convex* over the convex set X if $F(\lambda)$ has the convex form shown in Figure 1.17 for any two points $\mathbf{x}^1$ and $\mathbf{x}^2$ in X.

1.4.2 Gradient vector of $f(\mathbf{x})$

For a function $f(\mathbf{x}) \in C^2$ there exists, at any point $\mathbf{x}$ a vector of first order partial derivatives, or gradient vector:

$$\boldsymbol{\nabla} f(\mathbf{x}) = \begin{bmatrix} \dfrac{\partial f}{\partial x_1}(\mathbf{x}) \\[2ex] \dfrac{\partial f}{\partial x_2}(\mathbf{x}) \\ \vdots \\ \dfrac{\partial f}{\partial x_n}(\mathbf{x}) \end{bmatrix}. \tag{1.11}$$

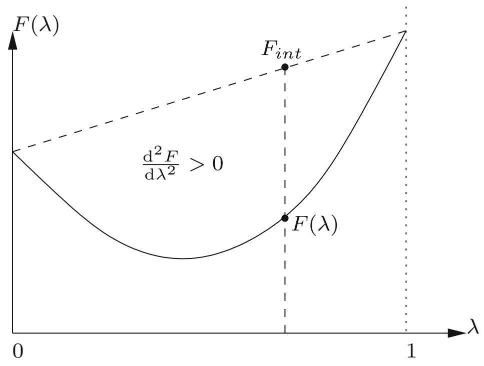

Figure 1.17: Convex form of $F(\lambda)$

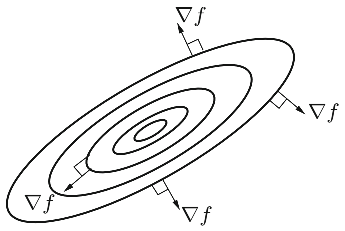

Figure 1.18: Directions of the gradient vector

It can easily be shown that if the function $f(\mathbf{x})$ is smooth, then at the point $\mathbf{x}$ the gradient vector $\boldsymbol{\nabla} f(\mathbf{x})$ is always perpendicular to the contours (or surfaces of constant function value) and is in the *direction of maximum increase* of $f(\mathbf{x})$, as depicted in Figure 1.18.

1.4.3 Hessian matrix of $f(\mathbf{x})$

If $f(\mathbf{x})$ is twice continuously differentiable then at the point $\mathbf{x}$ there exists a matrix of second order partial derivatives or *Hessian matrix*:

$$\begin{aligned}\mathbf{H}(\mathbf{x}) &= \left\{\frac{\partial^2 f}{\partial x_i \partial x_j}(\mathbf{x})\right\} = \boldsymbol{\nabla}^2 f(\mathbf{x}) \qquad (1.12)\\ &= \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(\mathbf{x}) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(\mathbf{x}) & \dots & \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(\mathbf{x}) & & & \\ \vdots & & & \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(\mathbf{x}) & & \dots & \frac{\partial^2 f}{\partial x_n^2}(\mathbf{x}) \end{bmatrix}.\end{aligned}$$

Clearly $\mathbf{H}(\mathbf{x})$ is a $n \times n$ symmetrical matrix.

1.4.3.1 Test for convexity of $f(\mathbf{x})$

If $f(\mathbf{x}) \in C^2$ is defined over a convex set X, then it can be shown (see Theorem 5.1.3 in Chapter 5) that if $\mathbf{H}(\mathbf{x})$ is positive-definite for all $\mathbf{x} \in X$, then $f(\mathbf{x})$ is strictly convex over X.

To test for convexity, i.e. to determine whether $\mathbf{H}(\mathbf{x})$ is positive-definite or not, apply Sylvester's Theorem or any other suitable numerical method (Fletcher 1987). For example, a convenient numerical test for positive-definiteness at $\mathbf{x}$ is to show that all the eigenvalues for $\mathbf{H}(\mathbf{x})$ are positive.

1.4.4 The quadratic function in $\mathbb{R}^n$

The quadratic function in n variables may be written as

$$f(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x} + c \qquad (1.13)$$

where $c \in \mathbb{R}$, $\mathbf{b}$ is a real n-vector and $\mathbf{A}$ is a $n \times n$ real matrix that can be chosen in a non-unique manner. It is usually chosen symmetrical in which case it follows that

$$\boldsymbol{\nabla} f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b};\ \ \mathbf{H}(\mathbf{x}) = \mathbf{A}. \tag{1.14}$$

The function $f(\mathbf{x})$ is called positive-definite if $\mathbf{A}$ is positive-definite since, by the test in Section 1.4.3.1, a function $f(\mathbf{x})$ is convex if $\mathbf{H}(\mathbf{x})$ is positive-definite.

1.4.5 The directional derivative of $f(\mathbf{x})$ in the direction $\mathbf{u}$

It is usually assumed that $\|\mathbf{u}\| = 1$. Consider the differential:

$$df = \frac{\partial f}{\partial x_1} dx_1 + \cdots + \frac{\partial f}{\partial x_n} dx_n = \boldsymbol{\nabla}^T f(\mathbf{x}) d\mathbf{x}. \tag{1.15}$$

A point $\mathbf{x}$ on the line through $\mathbf{x}'$ in the direction $\mathbf{u}$ is given by $\mathbf{x} = \mathbf{x}(\lambda) = \mathbf{x}' + \lambda\mathbf{u}$, and for a small change $d\lambda$ in λ, $d\mathbf{x} = \mathbf{u}d\lambda$. Along this line $F(\lambda) = f(\mathbf{x}' + \lambda\mathbf{u})$ and the differential at any point $\mathbf{x}$ on the given line in the direction $\mathbf{u}$ is therefore given by $dF = df = \boldsymbol{\nabla}^T f(\mathbf{x})\mathbf{u}d\lambda$. It follows that the *directional derivative* at $\mathbf{x}$ in the *direction* $\mathbf{u}$ is

$$\frac{dF(\lambda)}{d\lambda} = \left.\frac{df(\mathbf{x})}{d\lambda}\right|_{\mathbf{u}} = \boldsymbol{\nabla}^T f(\mathbf{x})\mathbf{u}. \tag{1.16}$$

1.5 Unconstrained minimization

In considering the unconstrained problem: $\min\limits_{\mathbf{x}} f(\mathbf{x}),\ \mathbf{x} \in X \subseteq \mathbb{R}^n$, the following questions arise:

(i) what are the conditions for a minimum to exist,

(ii) is the minimum unique,

(iii) are there any relative minima?

Figure 1.19 (after Farkas and Jarmai 1997) depicts different types of minima that may arise for functions of a single variable, and for functions of two variables in the presence of inequality constraints. Intuitively,

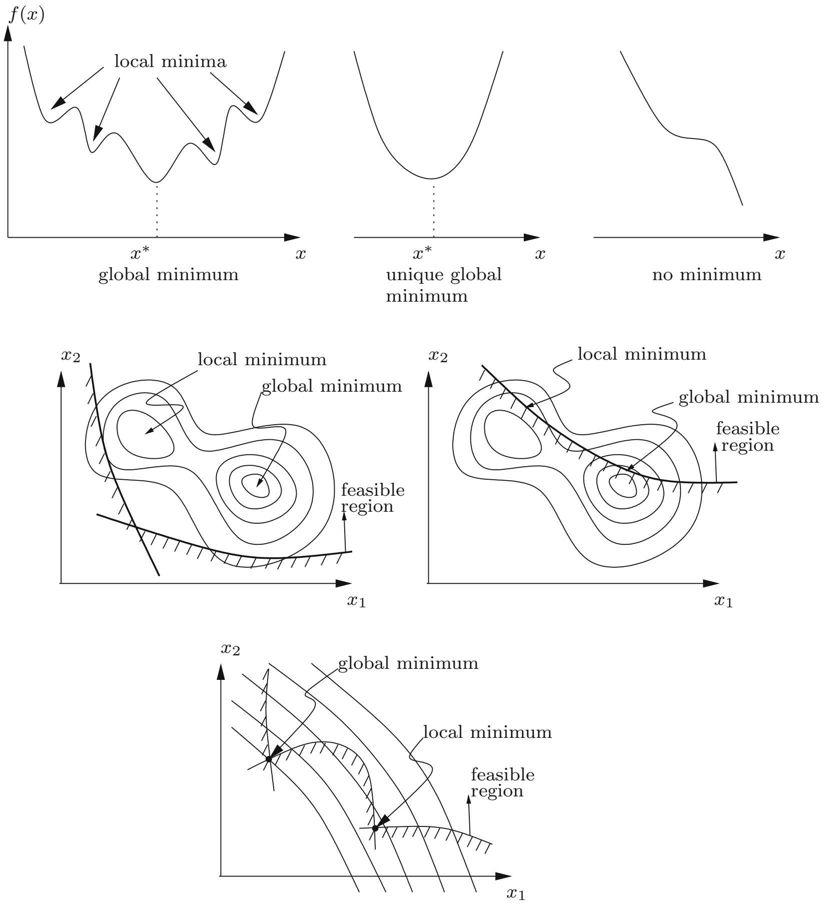

Figure 1.19: Types of minima

with reference to Figure 1.19, one feels that a general function may have a single unique global minimum, or it may have more than one local minimum. The function may indeed have no local minimum at all, and in two dimensions the possibility of saddle points also comes to mind. Thus, in order to answer the above questions regarding the nature of any given function more analytically, it is necessary to give more precise meanings to the above mentioned notions.

1.5.1 Global and local minima; saddle points

1.5.1.1 Global minimum

$\mathbf{x}^*$ is a global minimum over the set X if $f(\mathbf{x}) \geq f(\mathbf{x}^*)$ for all $\mathbf{x} \in X \subset \mathbb{R}^n$.

1.5.1.2 Strong local minimum

$\mathbf{x}^*$ is a strong local minimum if there exists an $\varepsilon > 0$ such that

$$f(\mathbf{x}) > f(\mathbf{x}^*) \text{ for all } \{\mathbf{x} \big| \|\mathbf{x} - \mathbf{x}^*\| < \varepsilon\}$$

where $|| \cdot ||$ denotes the Euclidean norm. This definition is sketched in Figure 1.20.

1.5.1.3 Test for unique local global minimum

It can be shown (see Theorems 5.1.4 and 5.1.5 in Chapter 5) that if $f(\mathbf{x})$ is strictly convex over X, then a strong local minimum is also the global minimum.

The global minimizer can be difficult to find since the knowledge of $f(\mathbf{x})$ is usually only local. Most minimization methods seek only a local minimum. An approximation to the global minimum is obtained in practice by the multi-start application of a local minimizer from randomly selected different starting points in X. The lowest value obtained after a sufficient number of trials is then taken as a good approximation to the global solution (see Snyman and Fatti 1987; Groenwold and Snyman

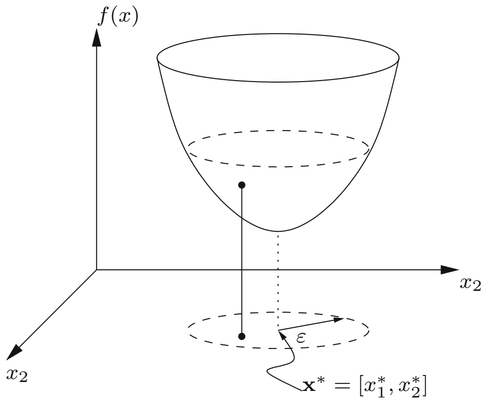

Figure 1.20: Graphical representation of the definition of a local minimum

2002). If, however, it is known that the function is strictly convex over X, then only one trial is sufficient since only one local minimum, the global minimum, exists.

1.5.1.4 Saddle points

$f(\mathbf{x})$ has a saddle point at $\overline{\mathbf{x}} = \begin{bmatrix} \mathbf{x}^0 \\ \mathbf{y}^0 \end{bmatrix}$ if there exists an $\varepsilon > 0$ such that for all $\mathbf{x}$, $\|\mathbf{x} - \mathbf{x}^0\| < \varepsilon$ and all $\mathbf{y}$, $\|\mathbf{y} - \mathbf{y}^0\| < \varepsilon$: $f(\mathbf{x}, \mathbf{y}^0) \leq f(\mathbf{x}^0, \mathbf{y}^0) \leq f(\mathbf{x}^0, \mathbf{y})$.

A contour representation of a saddle point in two dimensions is given in Figure 1.21.

1.5.2 Local characterization of the behaviour of a multi-variable function

It is assumed here that $f(\mathbf{x})$ is a smooth function, i.e., that it is a twice continuously differentiable function ($f(\mathbf{x}) \in C^2$). Consider again the line $\mathbf{x} = \mathbf{x}(\lambda) = \mathbf{x}' + \lambda\mathbf{u}$ through the point $\mathbf{x}'$ in the direction $\mathbf{u}$.

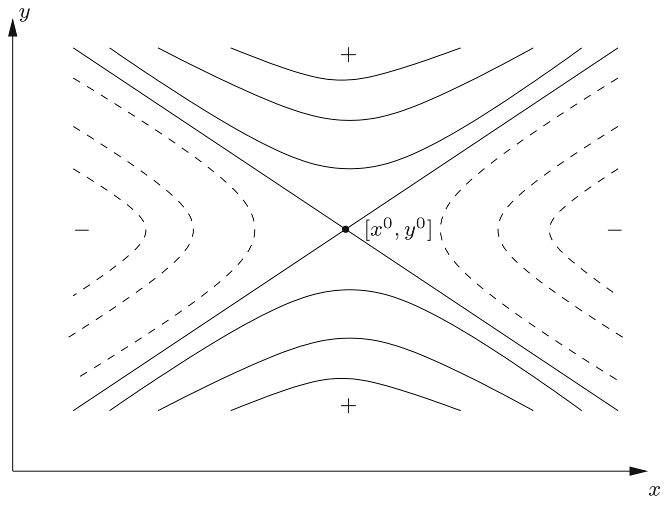

Figure 1.21: Contour representation of saddle point

Along this line a single variable function $F(\lambda)$ may be defined:

$$F(\lambda) = f(\mathbf{x}(\lambda)) = f(\mathbf{x}' + \lambda\mathbf{u}).$$

It follows from (1.16) that

$$\frac{dF(\lambda)}{d\lambda} = \left.\frac{df(\mathbf{x}(\lambda))}{d\lambda}\right|_{\mathbf{u}} = \boldsymbol{\nabla}^T f(\mathbf{x}(\lambda))\mathbf{u} = g(\mathbf{x}(\lambda)) = G(\lambda)$$

which is also a single variable function of λ along the line $\mathbf{x} = \mathbf{x}(\lambda) = \mathbf{x}' + \lambda\mathbf{u}$.

Thus similarly it follows that

$$\begin{aligned} \frac{d^2F(\lambda)}{d\lambda^2} = \frac{dG(\lambda)}{d\lambda} = \left.\frac{dg(\mathbf{x}(\lambda))}{d\lambda}\right|_{\mathbf{u}} &= \boldsymbol{\nabla}^T g(\mathbf{x}(\lambda))\mathbf{u} \\ &= \boldsymbol{\nabla}^T \left(\boldsymbol{\nabla}^T f(\mathbf{x}(\lambda))\mathbf{u}\right)\mathbf{u} \\ &= \mathbf{u}^T\mathbf{H}(\mathbf{x}(\lambda))\mathbf{u}. \end{aligned}$$

Summarising: the first and second order derivatives of $F(\lambda)$ with respect to λ at any point $\mathbf{x} = \mathbf{x}(\lambda)$ on any line (any $\mathbf{u}$) through $\mathbf{x}'$ is given by

$$\frac{dF(\lambda)}{d\lambda} = \boldsymbol{\nabla}^T f(\mathbf{x}(\lambda))\mathbf{u}, \tag{1.17}$$

$$\frac{d^2F(\lambda)}{d\lambda^2} = \mathbf{u}^T\mathbf{H}(\mathbf{x}(\lambda))\mathbf{u} \tag{1.18}$$

where $\mathbf{x}(\lambda) = \mathbf{x}' + \lambda\mathbf{u}$ and $F(\lambda) = f(\mathbf{x}(\lambda)) = f(\mathbf{x}' + \lambda\mathbf{u})$.

These results may be used to obtain Taylor's expansion for a multi-variable function. Consider again the single variable function $F(\lambda)$ defined on the line through $\mathbf{x}'$ in the direction $\mathbf{u}$ by $F(\lambda) = f(\mathbf{x}' + \lambda\mathbf{u})$. It is known that the Taylor expansion of $F(\lambda)$ about 0 is given by

$$F(\lambda) = F(0) + \lambda F'(0) + \tfrac{1}{2}\lambda^2 F''(0) + \dots \tag{1.19}$$

With $F(0) = f(\mathbf{x}')$, and substituting expressions (1.17) and (1.18) for respectively s$F'(\lambda)$ and $F''(\lambda)$ at $\lambda = 0$ into (1.19) gives

$$F(\lambda) = f(\mathbf{x}' + \lambda\mathbf{u}) = f(\mathbf{x}') + \boldsymbol{\nabla}^T f(\mathbf{x}')\lambda\mathbf{u} + \tfrac{1}{2}\lambda\mathbf{u}^T\mathbf{H}(\mathbf{x}')\lambda\mathbf{u} + \dots$$

Setting $\boldsymbol{\delta} = \lambda\mathbf{u}$ in the above gives the expansion:

$$f(\mathbf{x}' + \boldsymbol{\delta}) = f(\mathbf{x}') + \boldsymbol{\nabla}^T f(\mathbf{x}')\boldsymbol{\delta} + \tfrac{1}{2}\boldsymbol{\delta}^T\mathbf{H}(\mathbf{x}')\boldsymbol{\delta} + \dots \tag{1.20}$$

Since the above applies for any line (any $\mathbf{u}$) through $\mathbf{x}'$, it represents the general Taylor expansion for a multi-variable function about $\mathbf{x}'$. If $f(\mathbf{x})$ is fully continuously differentiable in the neighbourhood of $\mathbf{x}'$ it can be shown that the truncated second order Taylor expansion for a multi-variable function is given by

$$f(\mathbf{x}' + \boldsymbol{\delta}) = f(\mathbf{x}') + \boldsymbol{\nabla}^T f(\mathbf{x}')\boldsymbol{\delta} + \tfrac{1}{2}\boldsymbol{\delta}^T\mathbf{H}(\mathbf{x}' + \theta\boldsymbol{\delta})\boldsymbol{\delta} \tag{1.21}$$

for some $\theta \in [0, 1]$. This expression is important in the analysis of the behaviour of a multi-variable function at any given point $\mathbf{x}'$.

1.5.3 Necessary and sufficient conditions for a strong local minimum at $\mathbf{x}^*$

In particular, consider $\mathbf{x}' = \mathbf{x}^*$ a strong local minimizer. Then for any line (any $\mathbf{u}$) through $\mathbf{x}'$ the behaviour of $F(\lambda)$ in a neighbourhood of $\mathbf{x}^*$ is as shown in Figure 1.22, with minimum at at $\lambda = 0$.

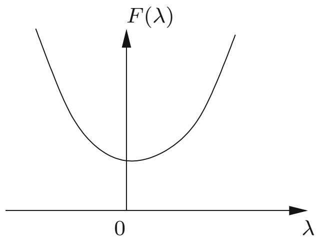

Figure 1.22: Behaviour of $F(\lambda)$ near $\lambda = 0$

Clearly, a *necessary first order condition* that must apply at $\mathbf{x}^*$ (corresponding to $\lambda = 0$) is that

$$\frac{dF(0)}{d\lambda} = \boldsymbol{\nabla}^T f(\mathbf{x}^*)\mathbf{u} = 0, \quad \text{for all } \mathbf{u} \neq \mathbf{0}. \tag{1.22}$$

It can easily be shown that this condition also implies that necessarily $\boldsymbol{\nabla} f(\mathbf{x}^*) = \mathbf{0}$.

A *necessary second order condition* that must apply at $\mathbf{x}^*$ is that

$$\frac{d^2F(0)}{d\lambda^2} = \mathbf{u}^T\mathbf{H}(\mathbf{x}^*)\mathbf{u} > 0, \quad \text{for all } \mathbf{u} \neq \mathbf{0}. \tag{1.23}$$

Conditions (1.22) and (1.23) taken together are also *sufficient conditions* (i.e. those that imply) for $\mathbf{x}^*$ to be a strong local minimum if $f(\mathbf{x})$ is continuously differentiable in the vicinity of $\mathbf{x}^*$. This can easily be shown by substituting these conditions in the Taylor expansion (1.21).

Thus in summary, the *necessary and sufficient conditions* for $\mathbf{x}^*$ to be a strong local minimum are:

$$\begin{array}{l} \boldsymbol{\nabla} f(\mathbf{x}^*) = \mathbf{0} \\ \mathbf{H}(\mathbf{x}^*) \text{ positive-definite.} \end{array} \tag{1.24}$$

In the argument above it has implicitly been assumed that $\mathbf{x}^*$ is an unconstrained minimum *interior* to X. If $\mathbf{x}^*$ lies on the *boundary* of X (see Figure 1.23) then

$$\frac{dF(0)}{d\lambda} \geq 0, \text{ i.e. } \boldsymbol{\nabla}^T f(\mathbf{x}^*)\mathbf{u} \geq 0 \tag{1.25}$$

for all *allowable* directions $\mathbf{u}$, i.e. for directions such that $\mathbf{x}^* + \lambda\mathbf{u} \in X$ for arbitrary small $\lambda > 0$.

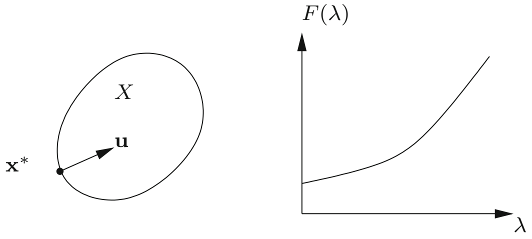

Figure 1.23: Behaviour of $F(\lambda)$ for all allowable directions of $\mathbf{u}$

Conditions (1.24) for an unconstrained strong local minimum play a very important role in the construction of practical algorithms for unconstrained optimization.

1.5.3.1 Application to the quadratic function

Consider the quadratic function:

$$f(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x} + c.$$

In this case the first order necessary condition for a minimum implies that

$$\boldsymbol{\nabla} f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b} = \mathbf{0}.$$

Therefore a candidate solution point is

$$\mathbf{x}^* = -\mathbf{A}^{-1}\mathbf{b}. \tag{1.26}$$

If the second order necessary condition also applies, i.e. if $\mathbf{A}$ is positive-definite, then $\mathbf{x}^*$ is a unique minimizer.

1.5.4 General indirect method for computing $\mathbf{x}^*$

The general indirect method for determining $\mathbf{x}^*$ is to solve the system of equations $\boldsymbol{\nabla} f(\mathbf{x}) = \mathbf{0}$ (corresponding to the first order necessary condition in (1.24)) by some numerical method, to yield all stationary points. An obvious method for doing this is Newton's method. Since in general the system will be non-linear, multiple stationary points are possible. These stationary points must then be further analysed in order to determine whether or not they are local minima.

1.5.4.1 Solution by Newton's method

Assume $\mathbf{x}^*$ is a local minimum and $\mathbf{x}^i$ an approximate solution, with associated unknown error $\boldsymbol{\delta}$ such that $\mathbf{x}^* = \mathbf{x}^i + \boldsymbol{\delta}$. Then by applying Taylor's theorem and the first order necessary condition for a minimum at $\mathbf{x}^*$ it follows that

$$\mathbf{0} = \boldsymbol{\nabla} f(\mathbf{x}^*) = \boldsymbol{\nabla} f(\mathbf{x}^i + \boldsymbol{\delta}) = \boldsymbol{\nabla} f(\mathbf{x}^i) + \mathbf{H}(\mathbf{x}^i)\boldsymbol{\delta} + \mathrm{O}\|\boldsymbol{\delta}\|^2.$$

If $\mathbf{x}^i$ is a good approximation then $\boldsymbol{\delta} \doteq \boldsymbol{\Delta}$, the solution of the linear system $\mathbf{H}(\mathbf{x}^i)\boldsymbol{\Delta} + \boldsymbol{\nabla} f(\mathbf{x}^i) = \mathbf{0}$, obtained by ignoring the second order term in $\boldsymbol{\delta}$ above. A better approximation is therefore expected to be $\mathbf{x}^{i+1} = \mathbf{x}^i + \boldsymbol{\Delta}$ which leads to the Newton iterative scheme: Given an initial approximation $\mathbf{x}^0$, compute

$$\mathbf{x}^{i+1} = \mathbf{x}^i - \mathbf{H}^{-1}(\mathbf{x}^i)\boldsymbol{\nabla} f(\mathbf{x}^i) \tag{1.27}$$

for $i = 0,\ 1,\ 2,\ \ldots$ Hopefully $\lim\limits_{i\to\infty} \mathbf{x}^i = \mathbf{x}^*$.

1.5.4.2 Example of Newton's method applied to a quadratic problem

Consider the unconstrained problem:

$$\text{minimize } f(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x} + c.$$

In this case the first iteration in (1.27) yields

$$\mathbf{x}^1 = \mathbf{x}^0 - \mathbf{A}^{-1}(\mathbf{A}\mathbf{x}^0 + \mathbf{b}) = \mathbf{x}^0 - \mathbf{x}^0 - \mathbf{A}^{-1}\mathbf{b} = -\mathbf{A}^{-1}\mathbf{b}$$

i.e. $\mathbf{x}^1 = \mathbf{x}^* = -\mathbf{A}^{-1}\mathbf{b}$ in a single step (see (1.26)). This is to be expected since in this case no approximation is involved and thus $\boldsymbol{\Delta} = \boldsymbol{\delta}$.

1.5.4.3 Difficulties with Newton's method

Unfortunately, in spite of the attractive features of the Newton method, such as being quadratically convergent near the solution, the basic Newton method as described above does not always perform satisfactorily. The main difficulties are:

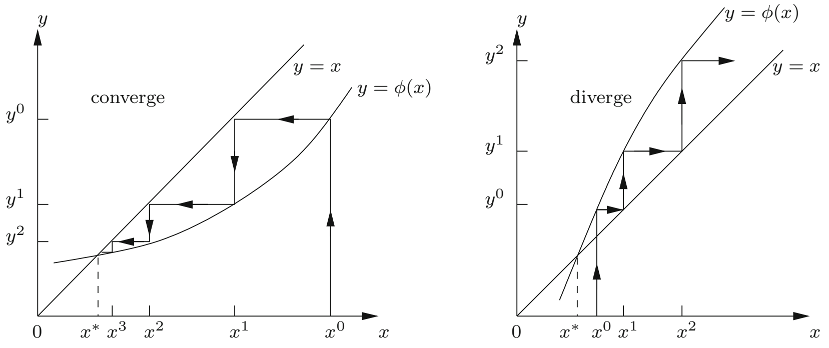

Figure 1.24: Graphical representation of Newton's iterative scheme for a single variable

(i) the method is not always convergent, even if $\mathbf{x}^0$ is close to $\mathbf{x}^*$, and

(ii) the method requires the computation of the Hessian matrix at each iteration.

The first of these difficulties may be illustrated by considering Newton's method applied to the one-dimensional problem: solve $f'(x) = 0$. In this case the iterative scheme is

$$x^{i+1} = x^i - \frac{f'(x^i)}{f''(x^i)} = \phi(x^i), \text{ for } i = 0,\ 1,\ 2,\ \ldots \tag{1.28}$$

and the solution corresponds to the fixed point x^* where $x^* = \phi(x^*)$. Unfortunately in some cases, unless x^0 is chosen to be exactly equal to x^*, convergence will not necessarily occur. In fact, convergence is dependent on the nature of the fixed point function $\phi(x)$ in the vicinity of x^*, as shown for two different ϕ functions in Figure 1.24. With reference to the graphs Newton's method is: $y^i = \phi(x^i),\ x^{i+1} = y^i$ for $i = 0,\ 1,\ 2,\ \ldots$. Clearly in the one case where $|\phi'(x)| < 1$ convergence occurs, but in the other case where $|\phi'(x)| > 1$ the scheme diverges.

In more dimensions the situation may be even more complicated. In addition, for a large number of variables, difficulty (ii) mentioned above becomes serious in that the computation of the Hessian matrix represents a major task. If the Hessian is not available in analytical form, use can be made of automatic differentiation techniques to compute it, or it can be estimated by means of finite differences. It should also be noted that in computing the Newton step in (1.27) a $n \times n$ linear system must be solved. This represents further computational effort. Therefore in practice the simple basic Newton method is not recommended. To avoid the convergence difficulty use is made of a modified Newton method, in which a more direct search procedure is employed in the direction of the Newton step, so as to ensure descent to the minimum $\mathbf{x}^*$. The difficulty associated with the computation of the Hessian is addressed in practice through the systematic update, from iteration to iteration, of an approximation of the Hessian matrix. These improvements to the basic Newton method are dealt with in greater detail in the next chapter.

1.6 Test functions

The efficiency of an algorithm is studied using standard functions with standard starting points $\mathbf{x}^0$. The total number of functions evaluations required to find the minimizer $\mathbf{x}^*$ is usually taken as a measure of the efficiency of the algorithm.

1.6.1 Unconstrained

Some classical unconstrained minimization test functions from (Rao 1996) are listed below.

1. Rosenbrock's parabolic valley:

$$f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2;\ \mathbf{x}^0 = \begin{bmatrix} -1.2 \\ 1.0 \end{bmatrix}\ \mathbf{x}^* = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

2. Quadratic function:

$$f(\mathbf{x}) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2;\ \mathbf{x}^0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}\ \mathbf{x}^* = \begin{bmatrix} 1 \\ 3 \end{bmatrix}.$$

3. Powell's quartic function:

$$\begin{aligned} f(\mathbf{x}) &= (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4; \\ \mathbf{x}^0 &= [3, -1, 0, 1]^T;\ \ \mathbf{x}^* = [0, 0, 0, 0]^T. \end{aligned}$$

4. Fletcher and Powell's helical valley:

$$\begin{aligned} f(\mathbf{x}) &= 100\left((x_3 - 10\theta(x_1,x_2))^2 \right. \\ &\quad \left. + \left(\sqrt{x_1^2 + x_2^2} - 1\right)^2\right) + x_3^2; \\ \text{where } 2\pi\theta(x_1,x_2) &= \begin{cases} \arctan \dfrac{x_2}{x_1} \text{ if } x_1 > 0 \\ \pi + \arctan \dfrac{x_2}{x_1} \text{ if } x_1 < 0 \end{cases} \\ \mathbf{x}^0 &= [-1,0,0]^T; \quad \mathbf{x}^* = [1,0,0]^T. \end{aligned}$$

5. A non-linear function of three variables:

$$\begin{aligned} f(\mathbf{x}) &= -\frac{1}{1+(x_1-x_2)^2} - \sin\left(\frac{1}{2}\pi x_2 x_3\right) - \exp\left(-\left(\frac{x_1+x_3}{x_2} - 2\right)^2\right); \\ \mathbf{x}^0 &= [0,1,2]^T; \quad \mathbf{x}^* = [1,1,1]^T. \end{aligned}$$

6. Freudenstein and Roth function:

$$\begin{aligned} f(\mathbf{x}) &= (-13 + x_1 + ((5 - x_2)x_2 - 2)x_2)^2 \\ &\quad +(-29 + x_1 + ((x_2 + 1)x_2 - 14)x_2)^2; \\ \mathbf{x}^0 &= [0.5,-2]^T; \quad \mathbf{x}^* = [5,4]^T; \ \mathbf{x}^*_{\text{local}} = [11.41\ldots,-0.8968\ldots]^T. \end{aligned}$$

7. Powell's badly scaled function:

$$\begin{aligned} f(\mathbf{x}) &= (10\,000 x_1 x_2 - 1)^2 + (\exp(-x_1) + \exp(-x_2) - 1.0001)^2; \\ \mathbf{x}^0 &= [0,1]^T; \quad \mathbf{x}^* = [1.098\ldots \times 10^{-5}, 9.106\ldots]^T. \end{aligned}$$

8. Brown's badly scaled function:

$$\begin{aligned} f(\mathbf{x}) &= (x_1 - 10^6)^2 + (x_2 - 2\times 10^{-6})^2 + (x_1 x_2 - 2)^2; \\ \mathbf{x}^0 &= [1,1]^T; \quad \mathbf{x}^* = [10^6, 2\times 10^{-6}]^T. \end{aligned}$$

9. Beale's function:

$$\begin{aligned} f(\mathbf{x}) &= (1.5 - x_1(1 - x_2))^2 + (2.25 - x_1(1 - x_2^2))^2 \\ &\quad +(2.625 - x_1(1 - x_2^3))^2; \\ \mathbf{x}^0 &= [1,1]^T; \quad \mathbf{x}^* = [3,0.5]^T. \end{aligned}$$

10. Wood's function:

$$\begin{aligned} f(\mathbf{x}) &= 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 \\ &\quad +10(x_2 + x_4 - 2)^2 + 0.1(x_2 - x_4)^2 \\ \mathbf{x}^0 &= [-3, -1, -3, -1]^T; \quad \mathbf{x}^* = [1, 1, 1, 1]^T. \end{aligned}$$

1.6.2 Constrained

Some classical constrained minimization test problems from Hock and Schittkowski (1981) are listed below.

1. Hock & Schittkowski Problem 1:

$f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$
such that
$x_2 \geq -1.5$

$$\mathbf{x}^0 = \begin{bmatrix} -2 \\ 1 \end{bmatrix} \ \mathbf{x}^* = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \ \lambda^* = 0$$

2. Hock & Schittkowski Problem 2:

$f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$
such that
$x_2 \geq 1.5$

$$\mathbf{x}^0 = \begin{bmatrix} -2 \\ 1 \end{bmatrix} \ \mathbf{x}^* = \begin{bmatrix} 1.2243707487363527 \\ 1.5000000000000000 \end{bmatrix} \ \lambda^* = 200$$

3. Hock & Schittkowski Problem 6:

$f(\mathbf{x}) = (1 - x_1)^2$
such that
$10(x_2 - x_1^2) = 0$

$$\mathbf{x}^0 = \begin{bmatrix} -1.2 \\ 1 \end{bmatrix} \ \mathbf{x}^* = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \ \lambda^* = 0.4$$

4. Hock & Schittkowski Problem 7:

$$f(\mathbf{x}) = \ln(1 + x_1^2) - x_2$$
such that
$$(1 + x_1^2)^2 + x_2^2 - 4 = 0$$

$$\mathbf{x}^0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \; \mathbf{x}^* = \begin{bmatrix} 0 \\ \sqrt{3} \end{bmatrix} \; \lambda^* = 3.15$$

5. Hock & Schittkowski Problem 10:

$$f(\mathbf{x}) = x_1 - x_2$$
such that
$$-3x_1^2 + 2x_1x_2 - x_2^2 + 1 \geq 0$$

$$\mathbf{x}^0 = \begin{bmatrix} -10 \\ 10 \end{bmatrix} \; \mathbf{x}^* = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \; \lambda^* = 1.0$$

6. Hock & Schittkowski Problem 18:

$$f(\mathbf{x}) = 0.01x_1^2 + x_2^2$$
such that
$$x_1x_2 - 25 \geq 0$$
$$x_1^2 + x_2^2 - 25 \geq 0$$
$$2 \leq x_1 \leq 50$$
$$2 \leq x_2 \leq 50$$

$$\mathbf{x}^0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \; \mathbf{x}^* = \begin{bmatrix} \sqrt{250} \\ \sqrt{2.5} \end{bmatrix} \; \lambda^* = 0.079$$

7. Hock & Schittkowski Problem 27:

$$f(\mathbf{x}) = 0.01(x_1 - 1)^2 + (x_2 - x_1^2)^2$$
such that
$$x_1 + x_3^2 + 1 = 0$$
$$\mathbf{x}^0 = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} \; \mathbf{x}^* = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \; \lambda^* = 2$$

8. Hock & Schittkowski Problem 42:

$$f(\mathbf{x}) = (x_1-1)^2 + (x_2-2)^2 + (x_3-3)^2 + (x_4-4)^2$$
such that
$$x_1 - 2 = 0$$
$$x_3^2 + x_4^2 - 2 = 0$$
$$\mathbf{x}^0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{x}^* = \begin{bmatrix} 2 \\ 2 \\ 0.6\sqrt{2} \\ 0.8\sqrt{2} \end{bmatrix} \quad \lambda^*_{max} = 7.07,\ \lambda^*_{min} = 3.54$$

9. Hock & Schittkowski Problem 66:

$$f(\mathbf{x}) = 0.2x_3 - 0.8x_1$$
such that
$$x_2 - \exp(x_1) \geq 0$$
$$x_3 - \exp(x_2) \geq 0$$
$$0 \leq x_1 \leq 100$$
$$0 \leq x_2 \leq 100$$
$$0 \leq x_3 \leq 10$$
$$\mathbf{x}^0 = \begin{bmatrix} 0 \\ 1.05 \\ 2.9 \end{bmatrix} \quad \mathbf{x}^* = \begin{bmatrix} 0.1841264879 \\ 1.202167873 \\ 3.327322322 \end{bmatrix} \quad \lambda^*_{max} = 0.096,\ \lambda^*_{min} = 0.096$$

10. Hock & Schittkowski Problem 104:

$$f(\mathbf{x}) = 0.4x_1^{0.67}x_7^{-0.67} + 0.4x_2^{0.67}x_8^{-0.67} + 10 - x_1 - x_2$$
such that
$$1 - 0.0588x_5x_7 - 0.1x_1 \geq 0$$
$$1 - 0.0588x_6x_8 - 0.1x_1 - 0.1x_2 \geq 0$$
$$1 - 4x_3x_5^{-1} - 2x_3^{-0.71}x_5^{-1} - 0.0588x_3^{-1.3}x_7 \geq 0$$
$$1 - 4x_4x_6^{-1} - 2x_4^{-0.71}x_6^{-1} - 0.0588x_4^{-1.3}x_8 \geq 0$$
$$1 \leq f(\mathbf{x}) \leq 4.2$$
$$1 \leq x_i \leq 10, \quad i = 1, \ldots, 8$$

$$\mathbf{x}^0 = [\ 6,\ 3,\ 0.4,\ 0.2,\ 6,\ 6,\ 1,\ 0.5\ ]^{\mathrm{T}}$$
$$\mathbf{x}^* = [6.465114,\ 2.232709,\ 0.6673975,\ 0.5957564,\ 5.932676,\ 5.527235,\ 1.013322,\ 0.4006682]^{\mathrm{T}}$$
$$\lambda^*_{max} = 1.87,\ \lambda^*_{min} = 0.043$$

1.7 Exercises

1.7.1 Sketch the graphical solution to the following problem:

$$\min_{\mathbf{x}} f(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 2)^2$$
$$\text{such that } x_1 + 2x_2 = 4;\ x_1 \geq 0;\ x_2 \geq 0.$$

In particular indicate the feasible region:

$$F = \{(x_1, x_2) \big| x_1 + 2x_2 \ = \ 4; x_1 \ \geq \ 0; x_2 \ \geq \ 0\}$$

and the solution point $\mathbf{x}^*$.

1.7.2 Show that x^2 is a convex function.

1.7.3 Show that the sum of convex functions is also convex.

1.7.4 Determine the gradient vector and Hessian matrix of the Rosenbrock function given in Section 1.6.1.

1.7.5 Write the quadratic function $f(\mathbf{x}) = x_1^2 + 2x_1x_2 + 3x_2^2$ in the standard matrix-vector notation. Is $f(\mathbf{x})$ positive-definite?

1.7.6 Write each of the following objective functions in standard form:

$$f(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x} + c.$$

(i) $f(\mathbf{x}) = x_1^2 + 2x_1x_2 + 4x_1x_3 + 3x_2^2 + 2x_2x_3 + 5x_3^2 + 4x_1 - 2x_2 + 3x_3$.

(ii) $f(\mathbf{x}) = 5x_1^2 + 12x_1x_2 - 16x_1x_3 + 10x_2^2 - 26x_2x_3 + 17x_3^2 - 2x_1 - 4x_2 - 6x_3$.

(iii) $f(\mathbf{x}) = x_1^2 - 4x_1x_2 + 6x_1x_3 + 5x_2^2 - 10x_2x_3 + 8x_3^2$.

1.7.7 Determine the definiteness of the following quadratic form:

$$f(\mathbf{x}) = x_1^2 - 4x_1x_2 + 6x_1x_3 + 5x_2^2 - 10x_2x_3 + 8x_3^2. \tag{1.29}$$

1.7.8 Approximate the Rosenbrock function given in Section 1.6.1 using a first order Taylor series expansion around $\mathbf{x}^0$. Compute the accuracy of the approximation at $\mathbf{x} = \mathbf{x}^0 + \Delta\mathbf{x}$, with $\Delta\mathbf{x} = [0,\ 1.0]^{\mathrm{T}}$.

1.7.9 Approximate the Rosenbrock function given in Section 1.6.1 using a second order Taylor series expansion around $\mathbf{x}^0$. Compute the accuracy of the approximation at $\mathbf{x} = \mathbf{x}^0 + \Delta\mathbf{x}$, with $\Delta\mathbf{x} = [0,\ 1.0]^{\mathrm{T}}$.

1.7.10 Compute the directional derivatives for the Rosenbrock function given in Section 1.6.1 at $\mathbf{x}^0$ along the following three directions

$$\begin{aligned}\mathbf{u}^1 &= [1,\ 0],\\ \mathbf{u}^2 &= [0,\ 1],\\ \mathbf{u}^3 &= [\frac{1}{\sqrt{2}},\ \frac{1}{\sqrt{2}}].\end{aligned}$$

Compare the first two computed directional derivatives to the components of the gradient vector. What conclusions can you draw.

1.7.11 Clearly state which of the directions computed in Exercise 1.7.10 are descent directions, i.e. directions along which the function will decrease for small positive steps along the direction.

1.7.12 Propose a descent direction of unit length that would result in the largest directional derivative magnitude at $\mathbf{x}^0$.

1.7.13 Compute the eigenvalues and eigenvectors for the computed $\mathbf{A}$ matrices in Exercise 1.7.6.

1.7.14 Determine whether the $\mathbf{A}$ matrices computed in Exercise 1.7.6 are positive-definite, negative-definite or indefinite.

1.7.15 Compare the associated eigenvalue for each computed eigenvector $\mathbf{u}^i$ in Exercise 1.7.13 against the second derivative of the univariate function $f(\lambda) = (\mathbf{x^0} + \lambda\mathbf{u^i})^{\mathrm{T}}\mathbf{A}(\mathbf{x^0} + \lambda\mathbf{u^i})$ and draw concrete conclusions.

1.7.16 Consider the following constrained optimization problem

$\min_{\mathbf{x}} f(\mathbf{x}) = x_1^4 - 6x_1^3 + 9x_1^2 + x_2^4 - 0.5x_2^3 + 0.0625x_2^2,$
such that
$x_1 \geq 0,$
$x_2 \leq 0.$

Utilizing only transformation of variables reformulate the constrained minimization problem as an unconstrained minimization problem.

1.7.17 Given the function $f(\mathbf{x})$ and the non-linear variable scaling $\mathbf{z}(\mathbf{x}) = \mathbf{G}(\mathbf{x})$ that transforms the domain, $\mathbf{x} \in \mathcal{R}^n$, to the domain, $\mathbf{z} \in \mathcal{R}^n$, with inverse relation $\mathbf{x}(\mathbf{z}) = \mathbf{G}^{-1}(\mathbf{z})$ transforming $\mathbf{z}$ back to the $\mathbf{x}$ domain. By substituting $\mathbf{x}(\mathbf{z})$ into $f(\mathbf{x})$ we obtain $f(\mathbf{z})$. Utilize the chain rule to derive the expression for computing $\nabla_{\mathbf{z}} f(\mathbf{z})$.

1.7.18 Perform the first five Newton steps in solving the *Rosenbrock function* listed in Section 1.6.1.

1.7.19 Perform the first five Newton steps in solving the *Quadratic function* listed in Section 1.6.1.

1.7.20 Perform the first five Newton steps in solving the *Freudenstein and Roth function* listed in Section 1.6.1.

1.7.21 Perform the first five Newton steps in solving the *Powell's badly scaled function* listed in Section 1.6.1.

1.7.22 Perform the first five Newton steps in solving the *Brown's badly scaled function* listed in Section 1.6.1.

1.7.23 Perform the first five Newton steps in solving the *Beale's function* listed in Section 1.6.1.