Preface

This book, like its companion volume *Nonlinear Optimization with Financial Applications*, is an outgrowth of undergraduate and postgraduate courses given at the University of Hertfordshire and the University of Bergamo. It deals with the theory behind numerical methods for nonlinear optimization and their application to a range of problems in science and engineering. The book is intended for final year undergraduate students in mathematics (or other subjects with a high mathematical or computational content) and exercises are provided at the end of most sections. The material should also be useful for postgraduate students and other researchers and practitioners who may be concerned with the development or use of optimization algorithms. It is assumed that readers have an understanding of the algebra of matrices and vectors and of the Taylor and mean value theorems in several variables. Prior experience of using computational techniques for solving systems of linear equations is also desirable, as is familiarity with the behaviour of iterative algorithms such as Newton's method for nonlinear equations in one variable. Most of the currently popular methods for continuous nonlinear optimization are described and given (at least) an intuitive justification. Relevant convergence results are also outlined and we provide proofs of these when it seems instructive to do so. This theoretical material is complemented by numerical illustrations which give a flavour of how the methods perform in practice.

The particular themes and emphases in this book have grown out of the author's experience at the Numerical Optimization Centre (NOC). This was established in 1968 and its staff (including Laurence Dixon, Ed Hersom, Joanna Gomulka, Sean McKeown and Zohair Maany) have made important contributions in fields as diverse as quasi-Newton methods, sequential quadratic programming, nonlinear least squares, global optimization, optimal control and automatic differentiation.

The computational results quoted in this book have been obtained using a Fortran90 module derived from the NOC's OPTIMA library. This software is not described in detail but interested readers can obtain it from an ftp site. Some of the student exercises can be attempted using OPTIMA but most can also be tackled in other ways, for example via the SOLVER tool in Microsoft Excel, the MATLAB toolbox of optimization procedures or the NAG libraries in C and Fortran.

I am indebted to many people for help in the writing of this book. Besides the NOC colleagues already mentioned, I would like to thank all the mathematics staff at the University of Hertfordshire for their support. I have also received encouragement and advice from Marida Bertocchi of the University of Bergamo, Alistair Forbes of the National Physical Laboratory, Berc Rustem of Imperial College and Ming Zuo of the University of Alberta. Any mistakes or omissions that remain are entirely my responsibility. My thanks are also due to John Martindale, Ann Kostant, Elizabeth Loew and their colleagues at Springer for encouragement and help with the preparation of the book. Finally, my deepest thanks go to my wife Nancy Mattson who, for a second time, has put up with the domestic side-effects of my preoccupation with authorship.

This book seeks to capture a view of the subject that I have acquired over a working lifetime's involvement with optimization and its applications. Optimization, by definition, is concerned with making things better. It is natural, therefore, that it should apply its own principles to itself and – in my experience, at least – this can generate a lively spirit of friendly rivalry between practitioners and algorithm developers. This spirit is worth celebrating in quasi-haiku form:

Optimization
means a quest for best answers
by the best methods.

Optimism means
believing both objectives
are achievable.

I hope readers will be stimulated by the challenge of finding more and more effective solutions to practical problems that become increasingly difficult.

Michael Bartholomew-Biggs
January, 2008

Chapter 1

Introducing Optimization

1.1. A tank design problem

In an optimization problem we seek values for certain *design* or *control variables* which minimize (or sometimes maximize) an *objective function.* A good example is the problem of finding the dimensions of a rectangular open-topped tank in order to obtain the smallest surface area which encloses a given volume, V^*. (The purpose of such a design might be to minimize heat loss through the sides.) We denote the height by x_1 and the lengths of the edges of the base by x_2 and x_3. The volume and surface area are then given by

$$V = x_1 x_2 x_3 \quad \text{and} \quad S = 2x_1x_2 + 2x_1x_3 + x_2x_3.$$

Hence the design problem can be posed as

$$\text{Minimize } \; S = 2x_1x_2 + 2x_1x_3 + x_2x_3 \quad \text{subject to } \; x_1x_2x_3 = V^*. \tag{1.1.1}$$

This is a three-variable optimization problem which includes an *equality constraint.* Methods for solving problems of this kind are discussed in Chapters 16–18; but an alternative *unconstrained* formulation can be obtained by eliminating one of the unknowns. Because $x_3 = V^* x_1^{-1} x_2^{-1}$ we can also seek the optimum tank dimensions by solving

$$\text{Minimize } \; S = 2x_1x_2 + 2V^*x_2^{-1} + V^*x_1^{-1}. \tag{1.1.2}$$

The solution of problems of this kind is discussed in Chapters 5–11.

The optimal tank dimensions can be found by solving either (1.1.1) or (1.1.2). However an important factor has been omitted from both of them. If any two of the x_i have negative values then the constraint

M. Bartholomew-Biggs, *Nonlinear Optimization with Engineering Applications*,
DOI: 10.1007/978-0-387-78723-7_1, © Springer Science+Business Media, LLC 2008

on volume can still be satisfied but the surface area may be negative. Because a negative value for S is necessarily less than a positive one, a solution with, say, $x_1 < 0$ and $x_2 < 0$ might seem "better" than a solution with all the x_i positive. Of course, negative dimensions have no practical meaning and so the problem formulation should explicitly exclude them. We can do this by adding *inequality constraints*, as in

$$\text{Minimize } \; 2x_1x_2 + 2x_1x_3 + x_2x_3 \; \text{ s.t. } \; x_1x_2x_3 = V^*, \;\; x_i \geq 0, \;\; i = 1, 2, 3. \tag{1.1.3}$$

or

$$\text{Minimize } \; 2x_1x_2 + 2V^*x_2^{-1} + V^*x_1^{-1} \;\; \text{s.t.} \;\; x_i \geq 0, \; i = 1, 2. \tag{1.1.4}$$

(The abbreviation "s.t." is often used instead of "subject to".) Methods for dealing with problems such as (1.1.3) and (1.1.4) are considered in Chapters 20–23.

In this chapter and the next we restrict ourselves to unconstrained problems involving only one variable. We can obtain such a problem from the tank design example by adding an extra requirement that the base must be square; that is, $x_2 = x_3$. Now the expressions for volume and surface area become

$$V = x_1x_2^2 \;\; \text{and} \;\; S = 4x_1x_2 + x_2^2.$$

Using the constraint on V to eliminate x_1, we get S in terms of x_2 only; that is,

$$S = 4V^*x_2^{-1} + x_2^2. \tag{1.1.5}$$

Figure 1.1 shows S as a function of x_2 when $V^* = 5$. In this case the minimum occurs when $x_2 \approx 2.2$.

Figure 1.1 illustrates the well-known fact that, at the minimum of a *differentiable* function, the slope – that is, the first derivative – is zero. Hence, for this rather simple problem, we can obtain the minimum surface area by solving

$$\frac{dS}{dx_2} = -4V^*x_2^{-2} + 2x_2 = 0$$

which gives $x_2 = (2V^*)^{1/3}$. Hence, when $V^* = 5$, the optimum square base has edges of length 2.1544.

Not all optimization problems are as easy as the minimization of (1.1.5). Some objective functions are hard to differentiate; and, even when the first derivative has been found, the equation obtained by setting it to zero may be difficult to solve. This book describes some of the computational methods used by engineers and scientists to deal with

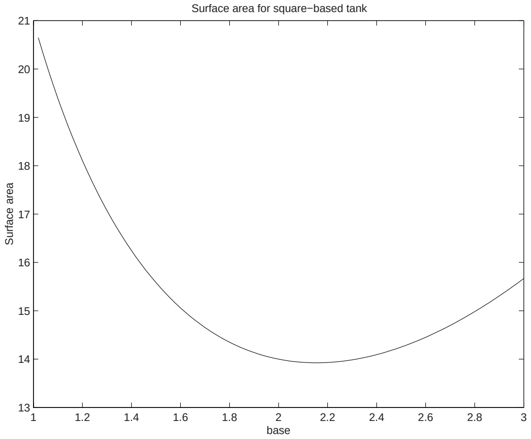

Figure 1.1. Tank surface area as a function of x_2.

optimization problems which do not have an analytical solution. Such problems occur in many situations, for example, finding a formula which gives the closest match to some experimental data, choosing the shortest route which avoids a number of obstacles, or devising a maintenance schedule which gives the least operating cost. Such case studies are used later in the book as a basis for the practical comparison of different optimization methods.

Exercises

1. What happens to the surface area (1.1.5) as $x_2 \to 0$? What is the minimum value of S if x_2 lies in the range $-1 < x_2 < 0$?

2. If $x_2 = x_3$, reformulate (1.1.2) as an unconstrained minimization problem involving x_1 only. Using the value $V^* = 5$, plot a graph of the objective function in the range $1 \leq x_1 \leq 3$. Hence deduce the minimum surface area. What happens to the surface area as $x_1 \to 0$?

3. Formulate the problem of finding the maximum volume that can be enclosed by a rectangular open tank with a fixed surface area and then estimate a solution when the base of the tank is square and the fixed surface area is 8. (Note that maximizing a function $F(x)$ is equivalent to minimizing $-F(x)$.)

1.2. Least squares data-fitting

Suppose that a laboratory experiment produces a record of measured temperatures, θ, (°C) against time t (minutes), as in Table 1.1. Suppose also that we believe the underlying relationship between θ and t is linear, of the form $\theta = at$, for some unknown coefficient a. The data points

Measurement i	1	2	3	4
Time t_i	1.0	2.0	3.0	4.0
Temperature θ_i	2.3	5.1	7.2	9.5

Table 1.1. Experimental data for temperature versus time.

do not, in fact, lie on a straight line (perhaps because of experimental errors). Hence, out of all the straight lines which pass near the data points, we wish to find the one which gives the best approximation, in the sense that the discrepancies between the data and the straight line *model* are as small as possible.

Figure 1.2 shows the errors $at_i - \theta_i$ as vertical lines PP'.

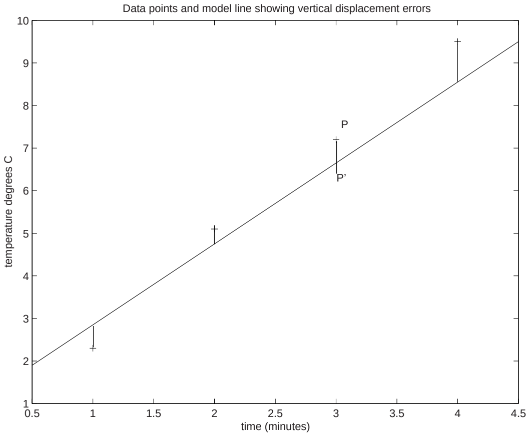

Figure 1.2. Data points and model line showing vertical errors.

A common way to find the best approximation is to choose a to minimize the sum of squares of these vertical errors; that is, we want a to solve the problem

$$\text{Minimize} \quad F(a) = \sum_{i=1}^{4} (at_i - \theta_i)^2. \tag{1.2.1}$$

At a minimum of F, the first derivative $F'(a)$ is zero. Hence the optimum value of a satisfies

$$\frac{dF}{da} = 2\sum_{i=1}^{4} (at_i - \theta_i)t_i = 0. \tag{1.2.2}$$

This leads to

$$a\sum_{i=1}^{4} t_i^2 = \sum_{i=1}^{4} \theta_i t_i.$$

Substituting for t_i and θ_i from Table 1.1 we get $30a = 72.1$ and so $a \approx 2.4033$.

This simple problem is an example of the *least squares* approach to approximating a set of data points by a model function. The approach can be extended (as shown in later chapters) to models with more than one unknown coefficient.

The data-fitting problem we have just solved is extremely easy because F is a quadratic function of the variable a. This means that equation (1.2.2) is linear and yields a unique answer. We now show that some optimization problems are not so straightforward by considering another way to minimize discrepancies between the data and the model. Rather than dealing with just the vertical error at a data point, we take account of the *total displacement* given by the perpendicular distance of $(t_i,\ \theta_i)$ from the line $\theta = at$ as shown in Figure 1.3.

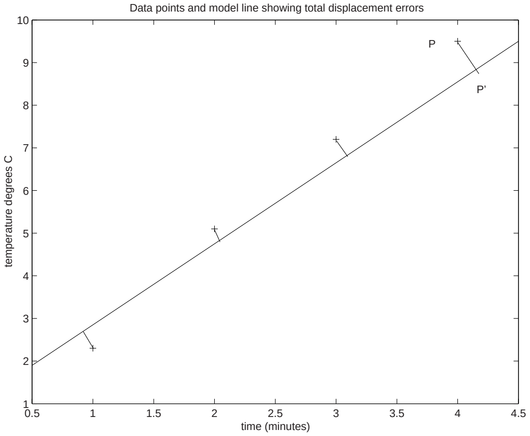

Figure 1.3. Data points and model line showing total displacement errors.

We can determine the perpendicular distance between point and line as follows. A typical point, P', on the line $\theta = at$ has coordinates (t, at) and the slope of the line joining P' to the data point P with coordinates (t_i, θ_i) is

$$m = \frac{at - \theta_i}{t - t_i}.$$

We want to find the value of t which makes $m = -a^{-1}$ because the line PP' will then be perpendicular to $\theta = at$ and P' will be the model point which is closest to the data point P. The value of t at the *footpoint* P' is found by solving

$$-a^{-1}(t - t_i) = at - \theta_i.$$

Hence P' is defined by

$$t = \tau_i = \frac{a^{-1}t_i + \theta_i}{a + a^{-1}} = \frac{t_i + a\theta_i}{a^2 + 1}. \tag{1.2.3}$$

The total displacement PP' is then $\sqrt{(t_i - \tau_i)^2 + (\theta_i - a\tau_i)^2}$ and to get the optimum straight line $\theta = at$ we must find a by solving

$$\text{Minimize } \hat{F}(a) = \sum_{i=1}^{8} \phi_i^2 \tag{1.2.4}$$

$$\text{where} \quad \phi_i = (t_i - \tau_i) \quad \text{and} \quad \phi_{i+4} = (\theta_i - a\tau_i) \quad \text{for} \quad i = 1, \ldots, 4. \tag{1.2.5}$$

Of course, $\tau_1, \ldots, \tau_4$ and $\phi_1, \ldots, \phi_8$ are functions of a. If we substitute the known values of t_i and θ_i we see that (1.2.4) is a more complicated expression than the corresponding function (1.2.1) in the vertical least-squares problem. From (1.2.3) we get

$$\tau_1 = \frac{1 + 2.3a}{a^2 + 1}; \quad \tau_2 = \frac{2 + 5.1a}{a^2 + 1}; \quad \tau_3 = \frac{3 + 7.2a}{a^2 + 1}; \quad \tau_4 = \frac{4 + 9.5a}{a^2 + 1}.$$

Hence

$$\phi_1 = \left(1 - \frac{1 + 2.3a}{a^2 + 1}\right), \quad \phi_5 = \left(2.3 - \frac{a + 2.3a^2}{a^2 + 1}\right)$$

with similar expressions for the remaining ϕ_i.

It is now clear that the function (1.2.4) is not quadratic and its first derivative is not linear. Hence, forming and solving the equation $\hat{F}'(a) = 0$ is more difficult than for the vertical least-squares problem. In practice, we would normally minimize a function such as $\hat{F}(a)$ by using iterative methods of the kind described in the next chapter. More information about total least squares and the footpoint problem is given in [27].

We can, of course, estimate the minimum of $\hat{F}(a)$ by plotting a graph, as shown in Figure 1.4. In this case, the best straight line approximation in the *total* least squares sense is very similar to the approximation based on vertical least squares with slope $a \approx 2.4$.

Exercises

1. Using vertical displacements, find the straight line $y = mx$ to give a least-squares approximation to the data points $(3, 7)$, $(4, 8)$, $(6, 11)$.

2. Show that the footpoint P' could have been found by putting $\tau_i = t_f$ where $t = t_f$ solves the problem

$$\text{Minimize } (t_i - t)^2 + (\theta_i - at)^2.$$

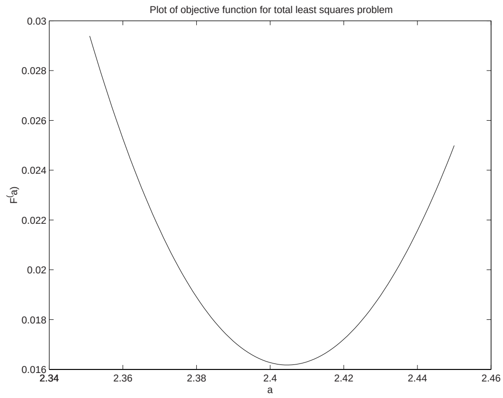

Figure 1.4. Plot of $\hat{F}(a)$ the total least squares error function.

3. Use the data in the worked example to find expressions for $\phi_2, \ldots, \phi_4$ and $\phi_6, \ldots, \phi_8$; hence complete the expression for $\hat{F}(a)$ in (1.2.4) and obtain the expression for $\hat{F}'(a)$. Plot a graph to estimate the solution of $\hat{F}'(a) = 0$.

4. Consider the data in Table 1.1 and suppose θ_4 is changed to 14.2. Calculate a model line $\theta = at$ using both vertical and total least squares. Comment on the difference between the two solutions.

1.3. A routing problem

Suppose a robot vehicle starts at the origin and is required to proceed to a point P, as shown in Figure 1.5. It must move initially along the x-axis and then turn towards P at some point Q. The circle represents a "no-go" area which the vehicle must avoid. The point Q is to be chosen to minimize a combination of the total distance travelled and the length of the route that lies within the circle.

If the line from Q to P cuts the circle at R and S then we can define the optimum route as the one which minimizes

$$F = \text{distance OQ} + \text{distance QP} + \rho(\text{distance RS})$$

where ρ is a positive constant. This form of function penalizes the portion of the route inside the no-go region. If ρ is large we expect little or none of the optimum route to pass through the circle. On the other hand, as $\rho \to 0$, the optimum route will come closer to the straight line OP.

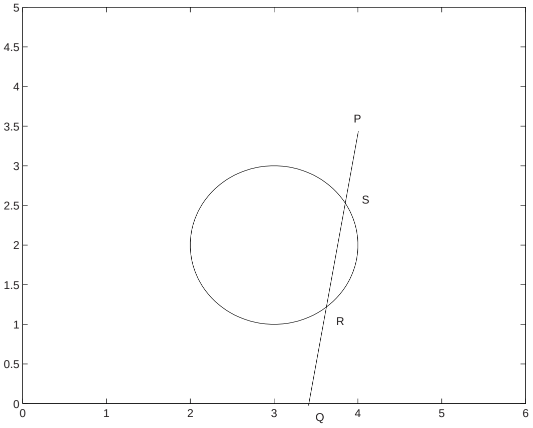

Figure 1.5. A routing problem.

If P has coordinates $(x_p,\ y_p)$ and if x is the (unknown) distance OQ then the total length of the route is

$$d(x) = x + \sqrt{(x - x_p)^2 + y_p^2}. \tag{1.3.1}$$

(We are assuming that P and the circle are in the positive quadrant and hence that x is positive.) We now need to determine the points of intersection (if any) of line QP and the circle. We assume that P is outside the no-go area and also that the circle does not cut the x-axis; it follows that the line segment QP will either cut the circle twice or not at all. The coordinates of any point on the line between Q and P are

$$(x + \lambda(x_p - x),\ \lambda y_p) \quad \text{where} \quad 0 \le \lambda \le 1.$$

If the no-go area has centre $(x_c,\ y_c)$ and radius r then points of intersection with QP occur when λ satisfies

$$(x + \lambda(x_p - x) - x_c)^2 + (\lambda y_p - y_c)^2 - r^2 = 0.$$

This simplifies to $\alpha\lambda^2 + \beta\lambda + \gamma = 0$, where the coefficients are given by

$$\alpha = (x - x_p)^2 + y_p^2, \quad \beta = 2[(x_p - x)(x - x_c) - y_p y_c] \tag{1.3.2}$$

$$\text{and} \quad \gamma = (x - x_c)^2 + y_c^2 - r^2. \tag{1.3.3}$$

We let $\delta = \beta^2 - 4\alpha\gamma$. If $\delta \le 0$ there are no points of intersection with the circle and so the distance RS is zero. On the other hand, if $\delta > 0$ the intersection points are given by

$$\lambda_1 = \frac{-\beta + \sqrt{\delta}}{2\alpha}, \quad \lambda_2 = \frac{-\beta - \sqrt{\delta}}{2\alpha}. \tag{1.3.4}$$

The distance RS is then given by $|\lambda_1 - \lambda_2| \times$ (distance QP). Hence the optimum route is obtained by minimizing

$$F(x) = d(x) + \rho v(x) \qquad (1.3.5)$$

where $d(x)$ is given by (1.3.1) and

$$v(x) = |\lambda_1 - \lambda_2|\sqrt{(x - x_p)^2 + y_p^2}. \qquad (1.3.6)$$

Note that α, β and γ are functions of x because of (1.3.2) and (1.3.3). Hence (1.3.4) implies that λ_1 and λ_2 also depend on x. It is possible – but not trivial – to differentiate $F(x)$ but it will not be possible to find an analytical solution to the equation $F'(x) = 0$.

If we take the target point $(x_p,\ y_p)$ as (5,4) and define the no-go region by $x_c = y_c = 2,\ r = 1$ then we can plot the function (1.3.5), as shown in Figure 1.6. It is clear that the optimum value of x is about 1.62. The solution path leaves the x-axis at a point Q such that PQ is a tangent to the circular boundary of the no-go region.

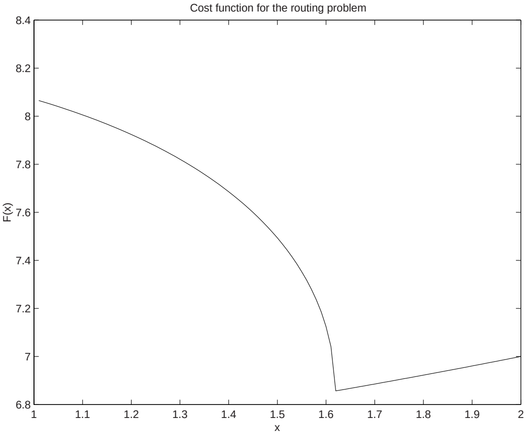

Figure 1.6. Nonsmooth cost function for the routing problem.

Figure 1.6 shows that the minimum corresponds to a “kink” in $F(x)$; that is, the slope of (1.3.5) is not zero at the optimum but instead has a discontinuity. This is due to the presence of the square root in (1.3.6). The function (1.3.5) is said to be *nonsmooth.*

Most of the optimization methods described in this book are intended for use with smooth (i.e., continuously differentiable) functions. We can formulate the routing problem in terms of a function which is smooth if we choose to minimize

$$\Phi(x) = d(x) + \rho v(x)^3 \qquad (1.3.7)$$

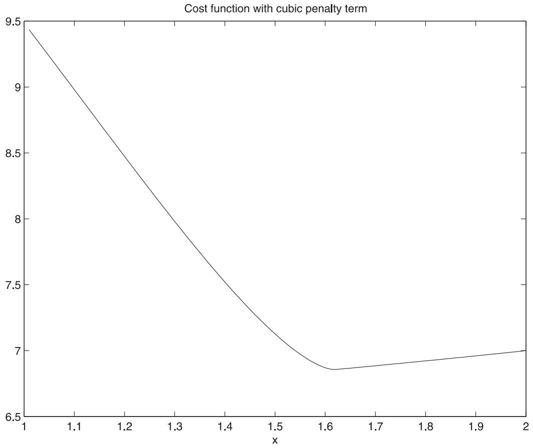

Figure 1.7. Smooth cost function for the routing problem.

whose graph is given in Figure 1.7 (for $\rho = 1$). The minimum of (1.3.7) occurs at approximately the same place as that of the nonsmooth function (1.3.5). The use of functions such as (1.3.7) in some real-life routing problems is described in [65, 9].

Exercises

1. Calculate expressions for the first derivatives of (1.3.1) and (1.3.6).

2. Using the sample data $x_p = 5,\ y_p = 4,\ x_c = y_c = 2,\ r = 1$, plot graphs to determine the minima of (1.3.5) and (1.3.7) when $\rho = 0.5, 0.05$ and 0.005. Comment on any differences you observe.

3. Use the data $x_p = 4,\ y_p = 8,\ x_c = 4,\ y_c = 2,\ r = 2$ to plot graphs of (1.3.5) and (1.3.7) with $\rho = 1$ in the range $0 \le x \le 10$. Comment on what you observe.