Practical Methods of Optimization

Second Edition

R. Fletcher
Department of Mathematics
University of Dundee, Scotland, UK

A Wiley–Interscience Publication

JOHN WILEY & SONS
Chichester · New York · Brisbane · Toronto · Singapore

Reprinted August 1989

Library of Congress Cataloguing in Publication Data:

Fletcher, R. (Roger)
Practical methods of optimization.

"A Wiley-Interscience publication."
Bibliography: p.
Includes index.
1. Mathematical optimization.
I. Title.
QA402.5.F57 1987 515 87-8126
ISBN 0 471 91547 5

British Library Cataloguing in Publication Data:

Fletcher, R.
Practical methods of optimization.
2nd ed.
1. Mathematical optimization
I. Title
515 QA402.5

ISBN 0 471 91547 5

Typeset by Thomson Press (India) Ltd, New Delhi
Printed and bound in Great Britain by Courier International Ltd, Tiptree, Essex

Contents

Preface

The subject of optimization is a fascinating blend of heuristics and rigour, of theory and experiment. It can be studied as a branch of pure mathematics, yet has applications in almost every branch of science and technology. This book aims to present those aspects of optimization methods which are currently of foremost importance in solving real life problems. I strongly believe that it is not possible to do this without a background of practical experience into how methods behave, and I have tried to keep practicality as my central theme. Thus basic methods are described in conjunction with those heuristics which can be valuable in making the methods perform more reliably and efficiently. In fact I have gone so far as to present comparative numerical studies, to give the feel for what is possible, and to show the importance (and difficulty) of assessing such evidence. Yet one cannot exclude the role of theoretical studies in optimization, and the scientist will always be in a better position to use numerical techniques effectively if he understands some of the basic theoretical background. I have tried to present such theory as shows how methods are derived, or gives insight into how they perform, whilst avoiding theory for theory's sake.

Some people will approach this book looking for a suitable text for undergraduate and postgraduate classes. I have used this material (or a selection from it) at both levels, in introductory engineering courses, in Honours mathematics lectures, and in lecturing to M.Sc. and Ph.D. students. In an attempt to cater for this diversity, I have used a Jekyll and Hyde style in the book, in which the more straightforward material is presented in simple terms, whilst some of the more difficult theoretical material is nonetheless presented rigorously, but can be avoided if need be. I have also tried to present worked examples for most of the basic methods. One observation of my own which I pass on for what it is worth is that the students gain far more from a course if they can be provided with computer subroutines for a few of the standard methods, with which they can perform simple experiments for themselves, to see for example how badly the steepest descent method handles Rosenbrock's problem, and so on.

In addition to the worked examples, each chapter is terminated by a set of questions which aim to not only illustrate but also extend the material in the

text. Many of the questions I have used in tutorial classes or examination papers. The reader may find a calculator (and possibly a programmable calculator) helpful in some cases. A few of the questions are taken from the Dundee Numerical Analysis M.Sc. examination, and are open book questions in the nature of a one day mini research project.

The second edition of the book combines the material in Volumes 1 and 2 of the first edition. Thus unconstrained optimization is the subject of Part 1 and covers the basic theoretical background and standard techniques such as line search methods, Newton and quasi-Newton methods and conjugate direction methods. A feature not common in the literature is a comprehensive treatment of restricted step or trust region methods, which have very strong theoretical properties and are now preferred in a number of situations. The very important field of nonlinear equations and nonlinear least squares (for data fitting applications) is also treated thoroughly. Part 2 covers constrained optimization which overall has a greater degree of complexity on account of the presence of the constraints. I have covered the theory of constrained optimization in a general (albeit standard) way, looking at the effect of first and second order perturbations at the solution. Some books prefer to emphasize the part played by convex analysis and duality in optimization problems. I also describe these features (in what I hope is a straightforward way) but give them lesser priority on account of their lack of generality.

Most finite dimensional problems of a continuous nature have been included in the book but I have generally kept away from problems of a discrete or combinatorial nature since they have an entirely different character and the choice of method can be very specialized. In this case the nearest thing to a general purpose method is the branch and bound method, and since this is a transformation to a sequence of continuous problems of the type covered in this volume, I have included a straightforward description of the technique. A feature of this book which I think is lacking in the literature is a treatment of non-differentiable optimization which is reasonably comprehensive and covers both theoretical and practical aspects adequately. I hope that the final chapter meets this need. The subject of geometric programming is also included in the book because I think that it is potentially valuable, and again I hope that this treatment will turn out to be more straightforward and appealing than others in the literature. The subject of nonlinear programming is covered in some detail but there are difficulties in that this is a very active research area. To some extent therefore the presentation mirrors my assessment and prejudice as to how things will turn out, in the absence of a generally agreed point of view. However, I have also tried to present various alternative approaches and their merits and demerits. Linear constraint programming, on the other hand, is now well developed and here the difficulty is that there are two distinct points of view. One is the traditional approach in which algorithms are presented as generalizations of early linear programming methods which carry out pivoting in a tableau. The other is a more recent approach in terms of active set strategies:

I regard this as more intuitive and flexible and have therefore emphasized it, although both methods are presented and their relationship is explored.

This second edition has given me the opportunity to improve the presentation of some parts of the book and to introduce new developments and a certain amount of new material. In Part 1 the description of line searches is improved and some new results are included. The variational properties of the BFGS and DFP methods are now described in some detail. More simple proofs of the properties of trust region methods are given. Recent developments in hybrid methods for nonlinear least squares are described. A thorough treatment of the Dennis–Moré theorem characterizing superlinear convergence in nonlinear systems is given and its significance is discussed. In Part 2 the treatment of linear programming has been extended considerably and includes new methods for stable updating of LU factors and the reliable treatment of degeneracy. Also, important recent developments in polynomial time algorithms are described and discussed, including ellipsoid algorithms and Karmarkar's method. The treatment of quadratic programming now includes a description of range space and dual active set methods. For general linear constraint programming some new theorems are given, including convergence proofs for a trust region method. The chapter on nonlinear programming now includes an extra section giving a direct treatment of the L_1 exact penalty function not requiring any convex analysis. New developments in sequential quadratic programming (SQP) are described, particularly for the case that only the reduced Hessian matrix is used. A completely new section on network programming is given relating numerical linear algebraic and graph theoretic concepts and showing their application in various types of optimization problem. For non-smooth optimization, Osborne's concept of structure functionals is used to unify the treatment of regularity for second order conditions and to show the equivalence to nonlinear programming. It is also used to demonstrate the second order convergence of a non-smooth SQP algorithm. The Maratos effect and the use of second order corrections are described. Finally a new section giving optimality conditions for constrained composite non-smooth optimization is included. A considerable number of new exercises is also given.

It is a great pleasure to me to acknowledge those many people who have influenced my thinking and contributed to my often inadequate knowledge. Amongst many I must single out the assistance and encouragement given to me by Professor M. J. D. Powell, my former colleague at AERE Harwell, and one whose contributions to the subject are unsurpassed. I am also indebted to Professor A. R. Mitchell and other members of the University of Dundee for providing the stimulating and yet relaxed environment in which this book was prepared. I also wish to thank Professor D. S. Jones for his interest and encouragement in publishing the book, and Drs M. P. Jackson, G. A. Watson and R. S. Womersley for their constructive advice on the contents. I gratefully acknowledge those various people who have taken the trouble to write or otherwise inform me of errors, misconceptions, etc., in text. Whilst this new

edition has given me the opportunity to correct previous errors, it has also inevitably enabled me to introduce many new ones for which I apologise in advance. I am also grateful for the invaluable secretarial help that I have received over the years in preparing various drafts of this book.

Last, but foremost, I wish to dedicate this book to my parents and family as some small acknowledgement of their unfailing love and affection.

Dundee, December 1986 R. Fletcher

Table of Notation

$\mathbf{A}$	matrix (Jacobian matrix, matrix of constraint normals)
$\mathbf{I}$	unit matrix
$\mathbf{L}, \mathbf{U}$	lower or upper triangular matrices respectively
$\mathbf{P}, \mathbf{Q}$	permutation matrix or orthogonal matrix
$\mathbf{a}$	vector (usually a column vector)
$\mathbf{a}_i$ $i = 1, 2, \ldots$	set of vectors (columns of $\mathbf{A}$)
$\mathbf{e}_i$ $i = 1, 2, \ldots$	coordinate vectors (columns of $\mathbf{I}$)
$\mathbf{e}$	vector of ones $(1, 1, \ldots, 1)^T$
$\mathbf{A}^T, \mathbf{a}^T$	transpose
$\mathbb{R}^n$	n-dimensional space
$\mathbf{x}$	variables in an optimization problem
$\mathbf{x}^{(k)}$ $k = 1, 2, \ldots$	iterates in an iterative method
$\mathbf{x}^*$	local minimizer or local solution
$\mathbf{s}, \mathbf{s}^{(k)}$	search direction (on iteration k)
$\alpha, \alpha^{(k)}$	step length (on iteration k)
$\delta, \delta^{(k)}$	correction to $\mathbf{x}^{(k)}$
$f(\mathbf{x})$	objective function
∇	first derivative operator (elements $\partial/\partial x_i$)
$\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x})$	gradient vector
$f^*, \mathbf{g}^*, \ldots$	$f(\mathbf{x}^*), \mathbf{g}(\mathbf{x}^*), \ldots$
$f^{(k)}, \mathbf{g}^{(k)}, \ldots$	$f(\mathbf{x}^{(k)}), \mathbf{g}(\mathbf{x}^{(k)}), \ldots$
∇^2	second derivative operator (elements $\partial^2/\partial x_i \partial x_j$)
$\mathbf{G}(\mathbf{x}) = \nabla^2 f(\mathbf{x})$	Hessian matrix (second derivative matrix)
$\mathbb{C}^k$	set of k times continuously differentiable functions
$l(\mathbf{x})$	linear function (1.2.8)
$q(\mathbf{x})$	quadratic function (1.2.11)
$[a, b]$	closed interval
(a, b)	open interval
$\|\cdot\|$	norm of a vector or matrix
$\square$	end of proof
$\exists, \forall$	'there exists', 'for all'
$\Rightarrow, \Leftrightarrow$	'implies', 'equivalent to'
$O(.), o(.)$	'big O' and 'little o' notation (Hardy, 1960) (let $h \to 0$: then $a = O(h)$ iff $\exists$ a constant c such that $\|a\| \leqslant ch$, and $a = o(h)$ iff $a/h \to 0$)
$\subset$	set inclusion
$\in$	element in set

$\varnothing$	empty set
$\triangleq$	equal by definition
$\mathscr{L}$	Lagrangian function
σ, ρ, τ	fixed parameters in algorithms
$\mathbf{r}$	residual vector, artificial variables vector
$\mathbf{c}_i(\mathbf{x})$ $i = 1, 2, \ldots$	constraint functions
E, I	set of equality and inequality constraints respectively (I is set of integer variables in Chapter 13)
$\mathscr{A}$	set of active constraints
$\mathbf{c}(\mathbf{x})$	vector of constraint functions (usually for equality constraints only)
$\mathbf{a}_i(\mathbf{x}) = \nabla \mathbf{c}_i(\mathbf{x})$	constraint gradient vector (normal vector)
x_i^+, x_i^-	$\max(x, 0)$ and $\max(-x, 0)$ respectively
$\mathbf{A}_B$	basis matrix in linear programming
B, N	sets of basic and nonbasic variables respectively
$\hat{\mathbf{A}}, \hat{\mathbf{c}}, \hat{\mathbf{b}}$	tableau matrix, reduced costs, basic variable values
$\mathbf{x}_c, \mathbf{x}_+$	current and next point (Section 8.7)
λ_i $i = 1, 2, \ldots$	Lagrange multipliers
$\boldsymbol{\lambda}$	vector of Lagrange multipliers (usually for equality constraints only)
$\mathbf{I}^*$	$\mathscr{A}^* \cap \mathbf{I}$ (set of active inequality constraints at $\mathbf{x}^*$)
$\mathscr{F}, F, \mathscr{G}, G$	feasible direction sets for optimality conditions
K	convex set
$\mathbf{Y}, \mathbf{Z}$	left inverse and null space matrices in generalized elimination
$:=$	assignment operator
$\mathrm{q}^{(k)}(\boldsymbol{\delta})$	model quadratic function obtained by Taylor series expansion about $\mathbf{x}^{(k)}$
$\mathbf{l}^{(k)}(\boldsymbol{\delta})$	linear model about $\mathbf{x}^{(k)}$ to vector of constraint functions
$\lvert\mathscr{A}\rvert$	number of elements in active set
v, σ	weighting parameters in penalty functions
$\dot{\mathbf{x}}$	$\mathrm{d}x/\mathrm{d}\theta$ for some trajectory $\mathbf{x}(\theta)$
∇_x, ∇_λ	partial derivative operators with respect to x, λ respectively
$\blacktriangledown = \begin{pmatrix} \nabla_x \\ \nabla_\lambda \end{pmatrix}$	combined vector of partial derivatives
$\mathbf{W} = \nabla_x^2 \mathscr{L}(\mathbf{x}, \boldsymbol{\lambda})$	Hessian of Lagrangian function
$\mathbf{W}^*, \mathbf{W}^{(k)}, \ldots$	$\mathbf{W}(\mathbf{x}^*, \boldsymbol{\lambda}^*), \mathbf{W}(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)}), \ldots$
$\mathbf{M}$	approximation to reduced Hessian matrix ($\mathbf{Z}^T\mathbf{G}\mathbf{Z}$ in Chapter 11, or $\mathbf{Z}^T\mathbf{W}\mathbf{Z}$ in Chapter 12)
$\phi(\mathbf{x})$	penalty function (possibly with additional parameters in argument list)
$\psi^{(k)}(\boldsymbol{\delta})$	model approximating function about $\mathbf{x}^{(k)}$ to $\phi(\mathbf{x}^{(k)} + \boldsymbol{\delta})$
$[\mathrm{x}]$	greatest integer not larger than x
G, T	graph, tree
$h(\mathbf{c})$	convex non-smooth function
$\partial h(\mathbf{c})$	subdifferential (set of all subgradients at $\mathbf{c}$)
$h^*, h^{(k)}, \ldots$	$h(\mathbf{c}^*), h(\mathbf{c}^{(k)}), \ldots$ (may be $h(\mathbf{c}(\mathbf{x}^*))$ etc. when $\mathbf{c} = \mathbf{c}(\mathbf{x})$)
$\mathbf{D}^*$	basis vectors for subdifferential (14.2.30)
$\partial h - \boldsymbol{\lambda}$	the set $\{\mathbf{u}: \mathbf{u} = \boldsymbol{\gamma} - \boldsymbol{\lambda}, \boldsymbol{\gamma} \in \partial h\}$
$\partial h \backslash \boldsymbol{\lambda}$	the set $\{\mathbf{u}: \mathbf{u} \in \partial h, \ \mathbf{u} \neq \boldsymbol{\lambda}\}$

PART 1

UNCONSTRAINED OPTIMIZATION

Chapter 1

Introduction

1.1 HISTORY AND APPLICATIONS

Optimization might be defined as the science of determining the 'best' solutions to certain mathematically defined problems, which are often models of physical reality. It involves the study of optimality criteria for problems, the determination of algorithmic methods of solution, the study of the structure of such methods, and computer experimentation with methods both under trial conditions and on real life problems. There is an extremely diverse range of practical applications. Yet the subject can be studied (not here) as a branch of pure mathematics.

Before 1940 relatively little was known about methods for numerical optimization of functions of many variables. There had been some least squares calculations carried out, and steepest descent type methods had been applied in some physics problems. The Newton method in many variables was known, and more sophisticated methods were being attempted such as the self-consistent field method for variational problems in theoretical chemistry. Nonetheless anything of any complexity demanded armies of assistants operating desk calculating machines. There is no doubt therefore that the advent of the computer was paramount in the development of optimization methods and indeed in the whole of numerical analysis. The 1940s and 1950s saw the introduction and development of the very important branch of the subject known as linear programming. (The term 'programming' by the way is synonymous with 'optimization' and was originally used to mean optimization in the sense of optimal planning.) All these methods however had a fairly restricted range of application, and again in the post-war period the development of 'hill-climbing' methods took place—methods of wide applicability which did not rely on any special structure in the problem. The latter methods were at first very crude and inefficient, but the subject was again revolutionized in 1959 with the publication of a report by W. C. Davidon which led to the introduction of variable metric methods. My friend and colleague M. J. D. Powell describes a

meeting he attended in 1961 in which the speakers were telling of the difficulty of minimizing functions of ten variables, whereas he had just programmed a method based on Davidon's ideas which had solved problems of 100 variables in a short time. Since that time the development of the subject has proceeded apace and has included methods for a wide variety of problems. This book describes these developments in what is hoped will be a systematic and comprehensive way.

The applicability of optimization methods is widespread, reaching into almost every activity in which numerical information is processed (Science, Engineering, Mathematics, Economics, Commerce, etc.). To provide a comprehensive account of all these applications would therefore be unrealistic, but a selection might include:

(a) chemical reactor design;
(b) aero-engine or aero-frame design;
(c) structural design—buildings, bridges, etc.;
(d) commerce—resource allocation, scheduling, blending;

and applications to other branches of numerical analysis:

(e) data fitting;
(f) variational principles in p.d.e.s;
(g) nonlinear equations in o.d.e.s; and
(h) penalty functions.

More such applications can be found in the proceedings of a conference on 'Optimization in Action' (Dixon, 1976), and many more of course in the specialized technical literature. However to give some idea of what is involved consider the optimum design of a distillation column, which can be modelled in an idealized way as in Figure 1.1.1. The aim of such a column is to separate out a more volatile component from a mixture of components in the input

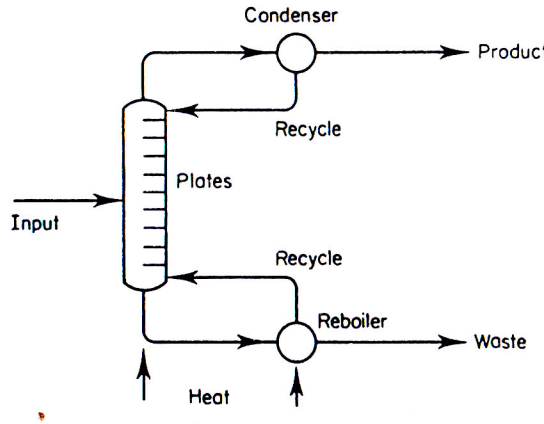

Figure 1.1.1 A model distillation column

stream. An *objective function* to be optimized might therefore be the quantity of the product or the profit from operating the system. The variables would be the rate of flow in the input, the heat rates applied and on each plate the liquid and vapour compositions of each component, and the temperature and vapour pressure. The variables are subject to restrictions or interrelations of many kinds, which are referred to as *constraints*. For instance compositions and flows must be non-negative ($x_i \geqslant 0$) and temperatures must not exceed certain upper bounds ($T_i \leqslant T_{\max}$). Relationships such as the unit sum of percentage compositions must be included explicitly ($\sum_i x_i = 1$). More complicated constraints state how components interact physically, for instance vapour and liquid compositions are related by $v_i = l_i \phi(T_i)$, where $\phi(T_i)$ is a given but highly nonlinear function of temperature. A more difficult situation arises if the number of plates in the column is allowed to vary, and this is an example of an *integer variable* which can take on only integer values.

This book however is not concerned with applications, except insofar as they indicate the different types of optimization problem which arise. It is possible to categorize these into a relatively small number of standard problems and to state algorithms for each one. The user's task is to discover into what category his problem fits, and then to call up the appropriate optimization subroutine on a computer. This subroutine will specify to the user how the problem data is to be presented, for example nonlinear functions usually have to be programmed in a user-written subroutine in a certain standard format. It is also as well to remember that in practice the solution of an optimization problem is not the only information that the user might need. He will often be interested in the *sensitivity* of the solution to changes in the parameters, especially so if the mathematical model is not a close approximation to reality, or if he cannot build his design to the same accuracy as the solution. He may indeed be interested in the variation of the solution obtained by varying some parameters over wide ranges, and it is often possible to provide this information without re-solving the problem numerous times.

This book therefore is concerned with some of the various standard optimization problems which can arise. In fact the material is divided into Part 1 and Part 2. Part 1 is devoted to the subject of *unconstrained optimization*, in which the optimum value is sought of an objective function of many variables, without any constraints. This problem is important in its own right and also as a major tool in solving some constrained problems. Also many of the ideas carry over into constrained optimization. The special case of sums of squares functions, which arise in data fitting problems, is also considered. This also includes the solution of sets of simultaneous nonlinear equations, which is an important problem in its own right, but which is often solved by optimization methods. Part 2 is devoted to *constrained optimization* in which the additional complication arises of the various types of constraint referred to above. An overview of constrained optimization is given in Section 7.

In this book a selection has had to be made amongst the extensive literature about optimization methods. I have been concerned to present *practical* methods

(and associated theory) which have been implemented and for which a body of satisfactory numerical experience exists. I am equally concerned about reliability of algorithms and whether there is proof or good reason to think that convergence to a solution will occur at a reasonably rapid rate. However, I shall also be trying to point out which new ideas in the subject I feel are significant and which might lead to future developments. Many people may read this book seeking a particular algorithm which best solves their specific problem. Such advice is not easy to give, especially in that the decision is not as clear-cut as it may seem. There are many special cases which should be taken into account, for instance the relative ease of computing the function and its derivatives. Similarly, considerations of how best to pose the problem in the first instance are relevant to the choice of method. Finally, and of most importance, the decision is subject to the availability of computer subroutines or packages which implement the methods. However some program libraries now give a decision tree in the documentation to help the user choose his method. Whilst these are valuable, they should only be used as a rough guide, and never as a substitute for common sense or the advice of a specialist in optimization techniques.

1.2 MATHEMATICAL BACKGROUND

The book relies heavily on the concepts and techniques of matrix algebra and numerical linear algebra, which are not set out here (see Broyden, 1975, for example), although brief explanations are given in passing in certain cases. A *vector* is represented by a lower case bold letter (e.g. $\mathbf{a}$) and usually refers to a column vector. A *matrix* is referred to by a bold upper case letter (e.g. $\mathbf{B}$). That is

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}; \qquad \mathbf{B} = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1s} \\ B_{21} & B_{22} & \cdots & B_{2s} \\ \vdots & \vdots & & \vdots \\ B_{r1} & B_{r2} & \cdots & B_{rs} \end{bmatrix}.$$

Sometimes b_{ij} is used for elements of $\mathbf{B}$ in place of B_{ij}. Transposition is referred to by superscript T so that $\mathbf{a}^T$ is a row vector and $\mathbf{a}^T\mathbf{z}$ for instance is the *scalar product* $\mathbf{a}^T\mathbf{z} = \mathbf{z}^T\mathbf{a} = \sum_i a_i z_i$.

The ideas of vector spaces are also used, although often only in a simple minded way. A *point* $\mathbf{x}$ in n-dimensional space ($\mathbb{R}^n$) is the vector $(x_1, x_2, \ldots, x_n)^T$, where x_1 is the component in the first coordinate direction, and so on. Most of the methods to be described are *iterative methods* which generate a *sequence* of points, $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, $\mathbf{x}^{(3)}$,... say, or $\{\mathbf{x}^{(k)}\}$ (the superscripts denoting iteration number), hopefully converging to a fixed point $\mathbf{x}^*$ which is the solution of the problem (see Figure 1.2.2). The idea of a *line* is important, and is the set of points

$$\mathbf{x}(=\mathbf{x}(\alpha)) = \mathbf{x}' + \alpha\mathbf{s} \tag{1.2.1}$$

for all α (sometimes for all $\alpha \geqslant 0$; this is strictly a half-line), in which $\mathbf{x}'$ is a fixed

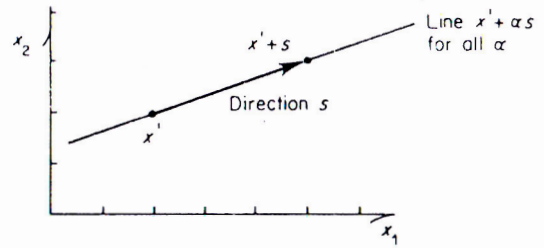

Figure 1.2.1 A line in two dimensions

point along the line (corresponding to $\alpha = 0$), and $\mathbf{s}$ is the *direction* of the line. For instance in Figure 1.2.1 $\mathbf{x}'$ is the point $\begin{pmatrix} 2 \\ 2 \end{pmatrix}$ and $\mathbf{s}$ the direction $\begin{pmatrix} 3 \\ 1 \end{pmatrix}$. The vector $\mathbf{s}$ is indicated by the arrow. Sometimes it is convenient to *normalize* $\mathbf{s}$ so that for instance $\mathbf{s}^T\mathbf{s} = \sum_i s_i^2 = 1$; this does not change the line, but only the value of α associated with any point.

The calculus of any *function* of many variables, $f(\mathbf{x})$ say, is clearly important. Some pictorial intuition for two variable problems is often gained by drawing *contours* (surfaces along which $f(\mathbf{x})$ is constant). A well-known test function for optimization methods is Rosenbrock's function

$$f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \tag{1.2.2}$$

the contours for which are shown in Figure 1.2.2. Some other contours are

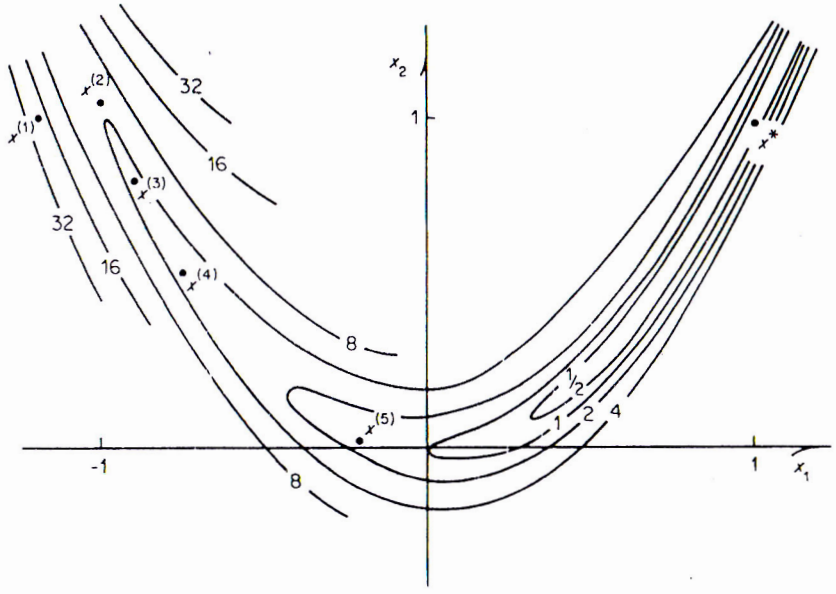

Figure 1.2.2 Contours for Rosenbrock's function, equation (1.2.2)

illustrated in Figure 6.2.2 in Chapter 6. In general it will be assumed that the problem functions which arise are *smooth*, that is continuous and continuously (Fréchet) differentiable ($\mathbb{C}^1$). Therefore for a function $f(\mathbf{x})$ at any point $\mathbf{x}$ there is a *vector of first partial derivatives*, or *gradient vector*

$$\begin{pmatrix} \partial f/\partial x_1 \\ \partial f/\partial x_2 \\ \vdots \\ \partial f/\partial x_n \end{pmatrix}_{\mathbf{x}} = \nabla f(\mathbf{x}) \tag{1.2.3}$$

where ∇ denotes the gradient operator $(\partial/\partial x_1, \ldots, \partial/\partial x_n)^{\mathrm{T}}$. If $f(\mathbf{x})$ is twice continuously differentiable ($\mathbb{C}^2$) then there exists a *matrix of second partial derivatives* or *Hessian matrix*, written $\nabla^2 f(\mathbf{x})$, for which the i, jth element is $\partial^2 f/(\partial x_i \partial x_j)$. This matrix is square and symmetric. Since any column (the jth, say) is $\nabla(\partial f/\partial x_j)$, the matrix can strictly be written as $\nabla(\nabla f^{\mathrm{T}})$. For example, in (1.2.2)

$$\begin{aligned} \nabla f(\mathbf{x}) &= \begin{pmatrix} -400x_1(x_2 - x_1^2) - 2(1 - x_1) \\ 200(x_2 - x_1^2) \end{pmatrix} \\ \nabla^2 f(\mathbf{x}) &= \begin{bmatrix} 1200x_1^2 - 400x_2 + 2 & -400x_1 \\ -400x_1 & 200 \end{bmatrix} \end{aligned} \tag{1.2.4}$$

and this illustrates that ∇f and $\nabla^2 f$ will in general depend upon $\mathbf{x}$, and vary from point to point. Thus at $\mathbf{x}' = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\nabla f(\mathbf{x}') = \begin{pmatrix} -2 \\ 0 \end{pmatrix}$ and $\nabla^2 f(\mathbf{x}') = \begin{bmatrix} 2 & 0 \\ 0 & 200 \end{bmatrix}$ by substitution into (1.2.4).

These expressions can be used to determine the derivatives of f along any line $\mathbf{x}(\alpha)$ in (1.2.1). By the chain rule

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} = \sum_i \frac{\mathrm{d}}{\mathrm{d}\alpha} x_i(\alpha) \frac{\partial}{\partial x_i} = \sum_i s_i \frac{\partial}{\partial x_i} = \mathbf{s}^{\mathrm{T}} \nabla \tag{1.2.5}$$

so the *slope* of $f(=f(\mathbf{x}(\alpha)))$ along the line at any point $\mathbf{x}(\alpha)$ is

$$\frac{\mathrm{d}f}{\mathrm{d}\alpha} = \mathbf{s}^{\mathrm{T}} \nabla f = \nabla f^{\mathrm{T}} \mathbf{s}. \tag{1.2.6}$$

Likewise the *curvature* along the line is

$$\frac{\mathrm{d}^2 f}{\mathrm{d}\alpha^2} = \frac{\mathrm{d}}{\mathrm{d}\alpha} \frac{\mathrm{d}f}{\mathrm{d}\alpha} = \mathbf{s}^{\mathrm{T}} \nabla (\nabla f^{\mathrm{T}} \mathbf{s}) = \mathbf{s}^{\mathrm{T}} \nabla^2 f \mathbf{s} \tag{1.2.7}$$

where ∇f and $\nabla^2 f$ are evaluated at $\mathbf{x}(\alpha)$. Note that, writing $\mathbf{G} = \nabla^2 f$, then $\mathbf{Gs}$ is the vector for which $(\mathbf{Gs})_i = \sum_j G_{ij} s_j$, and $\mathbf{s}^{\mathrm{T}}\mathbf{Gs}$ is the scalar product of $\mathbf{s}$ and $\mathbf{Gs}$. For example, for (1.2.2) at $\mathbf{x}' = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, the slope along the line generated by

$\mathbf{s} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ (the x_1-axis in Figure 1.2.2) is $\mathbf{s}^{\mathrm{T}} \nabla f = -2$ and the curvature is $\mathbf{s}^{\mathrm{T}}\mathbf{Gs} = 2$ $\left(\text{since } \mathbf{Gs} = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \right)$.

These definitions of slope and curvature depend on the size of $\mathbf{s}$, and this ambiguity can be resolved by requiring that $\|\mathbf{s}\| = 1$. (Note: the *norm* $\|\mathbf{s}\|$ is just a measure of the size of $\mathbf{s}$; one common norm is the L_2 norm $\|\mathbf{s}\|_2 = \sqrt{(\mathbf{s}^{\mathrm{T}}\mathbf{s})}$.) Denoting $\nabla f(\mathbf{x}')$ by $\mathbf{g}'$, then $\pm \mathbf{g}'/\|\mathbf{g}'\|_2$ are the directions of greatest and least slope at $\mathbf{x}'$, over all directions for which $\|\mathbf{s}\|_2 = 1$, and are orthogonal to the contour and tangent plane of $f(\mathbf{x})$ at $\mathbf{x}'$ (see Figure 1.2.3 and Question 1.4).

Special cases of many variable functions include the general *linear function* which can be written

$$l(\mathbf{x}) = \sum_{i=1}^{n} a_i x_i + b = \mathbf{a}^{\mathrm{T}}\mathbf{x} + b \tag{1.2.8}$$

where $\mathbf{a}$ and b are constant. (Strictly this should be described as an *affine* function on account of the existence of the constant b. However, the use of *linear* to describe a function whose graph is a line (or a hyperplane) is common in optimization. I do not intent to depart from this usage, but apologise to the erudite reader.) If the coordinate vector

$$\mathbf{e}_i = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \leftarrow i\text{th position} \tag{1.2.9}$$

is defined, then the identity $\nabla x_i = \mathbf{e}_i$ gives

$$\nabla \mathbf{x}^{\mathrm{T}} = \nabla(x_1, x_2, \ldots, x_n) = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n] = \mathbf{I} \tag{1.2.10}$$

since the vectors $\mathbf{e}_i$ are the columns of the *unit matrix* $\mathbf{I}$. Thus for (1.2.8), $\nabla l = \mathbf{a}$

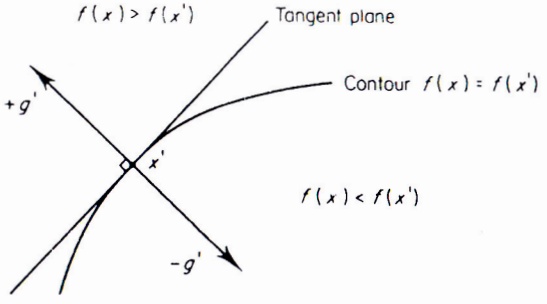

Figure 1.2.3 Properties of the gradient vector

is a constant vector, and $\nabla^2 l = \mathbf{0}$ is the *zero matrix*. A general *quadratic function* can be written

$$q(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T\mathbf{G}\mathbf{x} + \mathbf{b}^T\mathbf{x} + c \tag{1.2.11}$$

where $\mathbf{G}, \mathbf{b}$, and c are constant and $\mathbf{G}$ is symmetric, or as

$$q(\mathbf{x}) = \tfrac{1}{2}(\mathbf{x} - \mathbf{x}')^T\mathbf{G}(\mathbf{x} - \mathbf{x}') + c' \tag{1.2.12}$$

where $\mathbf{G}\mathbf{x}' = -\mathbf{b}$ and $c' = c - \tfrac{1}{2}\mathbf{x}'^T\mathbf{G}\mathbf{x}'$. From the rule for differentiating a product, it can be verified that

$$\nabla(\mathbf{u}^T\mathbf{v}) = (\nabla\mathbf{u}^T)\mathbf{v} + (\nabla\mathbf{v}^T)\mathbf{u} \tag{1.2.13}$$

if $\mathbf{u}$ and $\mathbf{v}$ depend upon $\mathbf{x}$. It therefore follows from (1.2.11) (using $\mathbf{u} = \mathbf{x}$, $\mathbf{v} = \mathbf{G}\mathbf{x}$) that

$$\nabla q(\mathbf{x}) = \tfrac{1}{2}(\mathbf{G} + \mathbf{G}^T)\mathbf{x} + \mathbf{b} = \mathbf{G}\mathbf{x} + \mathbf{b} \tag{1.2.14}$$

using the symmetry of $\mathbf{G}$. Likewise $\nabla^2 q = \mathbf{G}$ can be established. Thus $q(\mathbf{x})$ has a constant Hessian matrix $\mathbf{G}$ and its gradient is a linear function of $\mathbf{x}$. A consequence of (1.2.14) is that if $\mathbf{x}'$ and $\mathbf{x}''$ are two given points and if $\mathbf{g}' = \nabla q(\mathbf{x}')$ and $\mathbf{g}'' = \nabla q(\mathbf{x}'')$ then

$$\mathbf{g}'' - \mathbf{g}' = \mathbf{G}(\mathbf{x}'' - \mathbf{x}') \tag{1.2.15}$$

that is the Hessian matrix maps differences in position into differences in gradient. This result is used widely.

An indispensable technique for handling more general smooth functions of many variables is the *Taylor series*. For functions of one variable the infinite series is

$$f(\alpha) = f(0) + \alpha f'(0) + \tfrac{1}{2}\alpha^2 f''(0) + \cdots \tag{1.2.16}$$

although the series may be truncated after the term in α^p, replacing $f^{(p)}(0)$ by $f^{(p)}(\xi)$ where $\xi \in [0, \alpha]$. An integral form of the remainder can also be used. Now let $f(\alpha) = f(\mathbf{x}(\alpha))$ be the value of a function of many variables along the line $\mathbf{x}(\alpha)$ (see (1.2.1)). Then using (1.2.6) and (1.2.7) in (1.2.16)

$$f(\mathbf{x}' + \alpha\mathbf{s}) = f(\mathbf{x}') + \alpha\mathbf{s}^T\nabla f(\mathbf{x}') + \tfrac{1}{2}\alpha^2\mathbf{s}^T[\nabla^2 f(\mathbf{x}')]\mathbf{s} + \cdots \tag{1.2.17}$$

or by writing $\mathbf{h} = \alpha\mathbf{s}$

$$f(\mathbf{x}' + \mathbf{h}) = f(\mathbf{x}') + \mathbf{h}^T\nabla f(\mathbf{x}') + \tfrac{1}{2}\mathbf{h}^T[\nabla^2 f(\mathbf{x}')]\mathbf{h} + \cdots \tag{1.2.18}$$

These are two forms of the many variable Taylor series. Furthermore, consider applying (1.2.18) to the function $\partial f(\mathbf{x})/\partial x_i$. Since $\nabla(\partial f(\mathbf{x})/\partial x_i)$ is the ith column of the Hessian matrix $\nabla^2 f$, it follows that

$$\nabla f(\mathbf{x}' + \mathbf{h}) = \nabla f(\mathbf{x}') + [\nabla^2 f(\mathbf{x}')]\mathbf{h} + \cdots \tag{1.2.19}$$

which is a Taylor series expansion for the gradient of f. Neglecting the higher terms in the limit $\mathbf{h} \to \mathbf{0}$, then this reduces to (1.2.15) showing that a general

function behaves like a quadratic function in a sufficiently small neighbourhood of $\mathbf{x}'$.

It is hoped that a grasp of simple mathematical concepts such as these will enable the reader to follow most of the developments in the book. In certain places more complicated mathematics is used without detailed explanation. This is usually in an attempt to establish important results rigorously; however they often can be skipped over without losing the thread of the explanation. A summary of the notations used in the book is given immediately following the Preface.

QUESTIONS FOR CHAPTER 1

1.1. Obtain expressions for the gradient vector and Hessian matrix for the functions of n variables:
(i) $\mathbf{a}^T\mathbf{x}$: $\mathbf{a}$ constant;
(ii) $\mathbf{x}^T\mathbf{A}\mathbf{x}$: $\mathbf{A}$ unsymmetric and constant;
(iii) $\tfrac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x}$: $\mathbf{A}$ symmetric, $\mathbf{A}$, $\mathbf{b}$ constant;
(iv) $\mathbf{f}^T\mathbf{f}$: $\mathbf{f}$ is an m-vector depending on $\mathbf{x}$ and $\nabla\mathbf{f}^T$ is denoted by $\mathbf{A}$ which is not constant.

1.2. Write down the Taylor expansion for the gradient $\mathbf{g}(\mathbf{x}' + \boldsymbol{\delta})$ about $\mathbf{x}'$, neglecting terms of order $\|\boldsymbol{\delta}\|^2$. Hence show that if $f(\mathbf{x})$ is a quadratic function with Hessian $\mathbf{G}$, then $\boldsymbol{\gamma} = \mathbf{G}\boldsymbol{\delta}$, where $\boldsymbol{\delta}$ is the difference between any two points and $\boldsymbol{\gamma}$ is the corresponding difference in gradients.

1.3. Write down the Taylor expansion for the m-vector $\mathbf{f}(\mathbf{x})$ about $\mathbf{x}'$, where $\nabla\mathbf{f}^T$ is denoted by $\mathbf{A}$.

1.4. At a point $\mathbf{x}'$ for which $\mathbf{g}' \neq \mathbf{0}$, show that the direction vector $\mathbf{s} = \mathbf{g}'/\|\mathbf{g}'\|_2$ has the greatest slope, over all vectors for which $\mathbf{s}^T\mathbf{s} = 1$. (The *steepest ascent* vector.)

1.5. At a point $\mathbf{x}'$ for which $\mathbf{g}' \neq \mathbf{0}$, show that the direction vectors $\pm\mathbf{g}'$ are orthogonal to the contour and the tangent plane surface at $\mathbf{x}'$.

1.6. If $\mathbf{x}(\alpha)$ is any twice differentiable arc, if $f(\mathbf{x}(\alpha))$ is regarded as $f(\alpha)$, and if $d\mathbf{x}(\alpha_0)/d\alpha = \mathbf{s}$ and $d^2\mathbf{x}(\alpha_0)/d\alpha^2 = \mathbf{t}$, use the chain rule to obtain expressions for $df(\alpha_0)/d\alpha$ and $d^2 f(\alpha_0)/d\alpha^2$ in terms of $\mathbf{s}, \mathbf{t}$ and the derivatives of $f(\mathbf{x})$ evaluated at $\mathbf{x}(\alpha_0)$.

(Some other questions which partly refer to the material of Section 1.2 are given at the end of Chapter 2.)