

Math 171B
NUMERICAL OPTIMIZATION
Class Notes

© 1998. Philip E. Gill & Margaret H. Wright. All Rights Reserved.

Department of Mathematics
University of California, San Diego, La Jolla, CA 92093-0112.
September 1998

Contents

1	Background	7
1.1.	Introduction	7
1.2.	Eigenvalues and Singular Values	10
1.2.1	Eigenvalues and eigenvectors	10
1.2.2	The singular-value decomposition	13
1.3.	Norms	13
1.3.1	Vector norms	13
1.3.2	Matrix norms	15
1.4.	Some Properties of Sets	17
1.5.	Convergence of Sequences	18
1.5.1	Rates of convergence	20
1.6.	Properties of Univariate Functions	22
1.7.	Properties of Functions of Many Variables	24
1.7.1	Scalar-valued multivariate functions	26
1.7.2	The chain rule and its applications	27
1.7.3	Taylor-series for multivariate functions	28
1.8.	Results from Matrix Theory	28
1.9.	Properties of Convex Functions	29
2	Zero Finding	33
2.1.	Introduction	33
2.2.	Functions with one variable	34
2.2.1	The method of bisection	36
2.2.2	Newton's method for zero-finding	38
2.2.3	Secant and regula falsi methods	42
2.2.4	Safeguarded zero-finding algorithms	44
2.3.	Zero-finding for Functions of Many Variables	46
2.4.	Newton's Method	47
2.4.1	The Newton iteration	48
2.4.2	Convergence properties	49
2.5.	The Finite-Difference Newton Method	52
2.6.	Newton's Method with Backtracking	59
2.7.	Treatment of Singularity	63

3	Unconstrained Optimization	69
3.1.	Introduction	69
3.2.	Optimality Conditions	71
3.2.1	Functions of one variable	72
3.2.2	Functions of many variables	74
3.2.3	An alternative derivation	77
3.2.4	Properties of quadratic functions	78
3.3.	Minimizing a Function of One Variable	80
3.3.1	Interval reduction techniques	80
3.3.2	Polynomial interpolation	83
3.3.3	Safeguarded polynomial interpolation	85
3.4.	Model-Based Methods for Functions of Many Variables	85
3.4.1	Choice of model	85
3.4.2	The Quadratic Model	87
3.4.3	The Method of Steepest-Descent	87
3.4.4	Newton's Method	89
3.4.5	Local Convergence	90
3.5.	Line Search Methods	90
3.5.1	The backtracking line search	91
3.5.2	The Goldstein-Armijo line search	93
3.5.3	Modified Newton Methods	94
3.5.4	The spectral decomposition	95
3.5.5	Modified Newton as Steepest Descent	98
3.5.6	Convergence of Steepest Descent	100
3.5.7	Choosing the Line Search Parameter	104
3.6.	Trust-Region Methods	106
3.6.1	Computing the trust-region direction	108
4	Equality Constraints	115
4.1.	Introduction	115
4.2.	Optimality Conditions	117
4.2.1	Lagrange multipliers	119
4.2.2	The constraint qualification	120
4.2.3	The Karush-Kuhn-Tucker (KKT) conditions	123
4.2.4	The Lagrangian	123
4.3.	Quadratic Problems with Equality Constraints	128
4.3.1	Optimality conditions for EQP	128
4.3.2	Solving an equality constrained QP	131
4.4.	The Method of Multipliers	132
4.5.	Linear Equality Constraints	134
4.5.1	A local quadratic model	135
4.5.2	Newton-based methods	136
4.5.3	Discrete Newton methods	137
4.6.	Penalty Function Methods	137
4.6.1	The quadratic penalty function	138
4.6.2	Derivatives of the penalty function	141
4.6.3	The penalty trajectory	142

4.6.4	The “classical” penalty method	144
4.6.5	The primal-dual penalty method	148
4.6.6	The ℓ_1 penalty function	149
5	Inequality Constraints	151
5.1.	Introduction	151
5.2.	Optimality Conditions	152
5.2.1	First-order conditions	154
5.2.2	Second-order conditions	159
5.2.3	*Zero Lagrange multipliers	162
5.2.4	Optimality conditions in complementary slackness form	163
5.2.5	Convex problems	165
5.3.	General Nonlinear Constraints	166
5.4.	Penalty-Function Methods	168
5.5.	Barrier-Function Methods	168
5.5.1	The logarithmic barrier function	170
5.5.2	The barrier trajectory	171
5.5.3	Properties of the barrier Hessian	174

Chapter 1

Background

1.1. Introduction

Problems in all areas of mathematics, applied science, engineering, economics, medicine and statistics can be posed as *optimization problems*. An optimization problem begins with a set of independent variables or parameters, and often includes conditions or restrictions that define acceptable values of the variables. Such restrictions are known as the *constraints* of the problem. The other essential component of an optimization problem is a single measure of “goodness”, termed the *objective function*, which depends in some way on the variables. The solution of an optimization problem is a set of allowed values of the variables for which the objective function assumes its “optimal” value. In mathematical terms, this usually involves maximizing or minimizing.

Math 171B deals principally with *nonlinear* programming, which involves the minimization of a *nonlinear* function, possibly subject to *nonlinear* equality and inequality constraints.

In its most general form, the problem of interest is to find a point x^* such that $f(x^*)$ is the least value of $f(x)$ over a given set of acceptable values of x . This set, known as the *feasible region*, is usually defined in terms of a set of functional equalities and inequalities. For example, suppose that we wish to find the least value of the function $-x_1x_2^2$ for all nonnegative values of x_1 and x_2 that lie on the circle $x_1^2 + x_2^2 = 2$. This problem can be written formally as

$$\begin{array}{ll} \underset{x_1, x_2}{\text{minimize}} & -x_1x_2^2 \\ \text{subject to} & x_1^2 + x_2^2 = 2 \\ & x_1, x_2 \geq 0. \end{array} \quad (1.1.1)$$

Figure 1.1 shows the level curve of the function $f = -x_1x_2^2$ and the set of points that satisfies the constraint $x_1^2 + x_2^2 = 2$. This simple problem is an instance of the following general case:

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{\text{minimize}} & f(x) \\ \text{subject to} & c_i(x) = 0, \quad i = 1, 2, \dots, m_E, \\ & c_i(x) \geq 0, \quad i = m_E + 1, m_E + 2, \dots, m_E + m_I, \end{array}$$

where each $c_i(x)$ is a (possibly) nonlinear function of the n variables $x_1, x_2, \dots, x_n$. This problem has m_E equality constraints and m_I inequality constraints. The $\{c_i(x)\}$ are known as the *constraint functions* of the problem. For convenience, we will often consider the vector function $c(x)$ that has the constraint function $c_i(x)$ as its i th component. Throughout, it is assumed that f and c are twice-continuously differentiable—i.e., the second partial derivative of $c_i(x)$ with respect to each of the variables exists and is continuous.

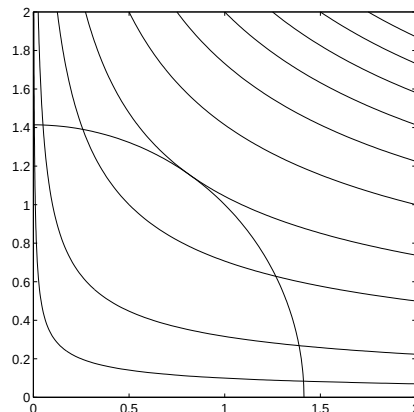


Figure 1.1: This graph depicts the level curves and constraint functions for the two-dimensional example (1.1.1). The feasible region consists of points lying on the circle $x_1^2 + x_2^2 = 2$ lying between the vertical and horizontal axes.

Before tackling this inequality constrained problem in all its generality, we consider first the so-called *unconstrained* problem, which occurs when there are no restrictions on x . This problem not only arises as a subproblem during the solution of the general case, but also occurs as a valid problem in its own right. Broadly speaking, an unconstrained minimizer x^* is a point at which $f(x)$ achieves its minimum value.

Throughout, we consider problems that are at least continuously differentiable, i.e., all first derivatives exist and are continuous. In some cases, we go further and assume that the second derivatives also exist and are continuous. The solution of the problem will always be denoted by x^* .

To motivate our interest in unconstrained problems, we consider a large class of unconstrained problems derived from applications in data fitting.

Consider a set of observed values $y_1, y_2, \dots, y_m$ of some process or experiment sampled at times $t_1, t_2, \dots, t_m$. We assume that the $\{y_i\}$ are values of some unknown function $y(t)$ that we would like to be able to define explicitly. One possibility is to approximate $y(t)$ by a simple function $\phi(t)$ defined in terms of a small number of parameters that can be adjusted to make $\phi(t) \approx y(t)$. In particular, we wish to identify the function ϕ that “best fits” the data values $\{y_i\}$ when evaluated at the data points $\{t_i\}$. One possibility is to use a simple polynomial form for ϕ . For example, we might estimate that the data can be

fitted by a quadratic polynomial of the form:

$$\phi(t) = x_1 + x_2 t + x_3 t^2,$$

where the values (x_1, x_2, x_3) are chosen so that $\phi(t)$ fits the data as closely as possible. In particular, we wish to choose (x_1, x_2, x_3) so that the magnitude of each of the error terms $y_i - (x_1 + x_2 t_i + x_3 t_i^2)$ is as small as possible. The most popular strategy is to minimize the sum of squares of these errors, which is the same as minimizing a function f such that

$$f(x) = \sum_{i=1}^m (y_i - (x_1 + x_2 t_i + x_3 t_i^2))^2,$$

where x is the vector with components x_1, x_2 and x_3 . This problem, known as the so-called *least-squares fitting problem*, is an unconstrained minimization problem since all values of the parameters x_1, x_2 , and x_3 are meaningful.

The data-fitting problem can be stated more generally as follows. Suppose that we wish to identify n parameters $x_1, x_2, \dots, x_n$. Consider the $m \times n$ matrix A and m -vector b such that

$$A = \begin{pmatrix} 1 & t_1 & t_1^2 \\ 1 & t_1 & t_1^2 \\ \vdots & \vdots & \vdots \\ 1 & t_m & t_m^2 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}.$$

The least-squares fitting problem is then equivalent to minimizing $\|b - Ax\|_2^2$, where $\|x\|_2$ denotes the two-norm $(\sum_{i=1}^m x_i^2)^{1/2}$. The unconstrained minimization of this function, which can be done by solving a single system of linear equations, will be discussed later.

In Figure 1.2 we give the results of fitting the quadratic polynomial $\phi(t) = x_1 + x_2 t + x_3 t^2$, to the data (t_i, y_i) given by $(1, 2), (2, 3), (3, 5), (4, 7)$. The left-hand figure gives the plot of the raw data, with $\{t_i\}$ on the horizontal axis and $\{y_i\}$ on the vertical axis. In the second figure depicts the quadratic polynomial that best fits the data in the least-squares sense. This polynomial is given by $\phi(t) = \frac{5}{4} + \frac{9}{20}t + \frac{1}{4}t^2$. We will show how to compute this polynomial later.

The main feature of the data-fitting problem considered so far is that the function $\phi(t)$ is *linear* in the parameters $x_1, x_2, \dots, x_n$. It is this that allows us to use the special methods of linear least-squares and linear programming described above. If $\phi(t)$ involves some nonlinear terms, the data fitting problem becomes a general (possibly nonsmooth) unconstrained minimization problem, and the degree of difficulty increases accordingly. For example, suppose we believe that for some values of x_1, x_2 , and x_3 , the following function best estimates the data:

$$\phi(t) = x_1 t + x_2 e^{x_3 t}.$$

In this case, it can be seen that $f(x)$ is the norm of a nonlinear function instead of a linear function. For example, in the least-squares case

$$f(x) = \sum_{i=1}^m (y_i - (x_1 t + x_2 e^{x_3 t}))^2.$$

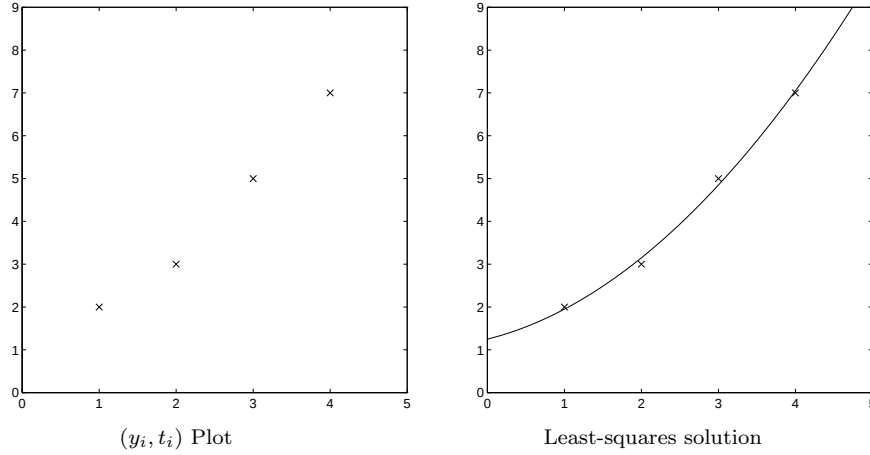


Figure 1.2: Data-fitting problem with least-squares best fit.

These problems, known as *nonlinear least-squares problems*, will be considered in more detail later. Nonlinear least-squares problems, or *nonlinear parameter optimization*, as it is often called, is one of the most common sources of unconstrained optimization problem.

1.2. Eigenvalues and Singular Values

1.2.1. Eigenvalues and eigenvectors

For any square matrix A , there is at least one number λ (which may be complex) and an associated nonzero vector u such that

$$(A - \lambda I)u = 0, \quad \text{or, equivalently,} \quad Au = \lambda u. \quad (1.2.1)$$

The first equation in (1.2.1) indicates that subtraction of λ from each diagonal element of A produces a singular matrix; the second relation shows that premultiplication of u by A does not alter the direction of u . A value λ for which (1.2.1) holds is called an *eigenvalue* of A , and the corresponding vector u is called an *eigenvector*. For example, when

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 3 & 1 \\ 0 & 3 \end{pmatrix}, \quad u = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \text{and} \quad v = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad (1.2.2)$$

we have

$$Au = \begin{pmatrix} 3 \\ 3 \end{pmatrix} = 3u \quad \text{and} \quad Bv = \begin{pmatrix} 3 \\ 0 \end{pmatrix} = 3v,$$

so that $\lambda = 3$ is an eigenvalue of both A and B , u is an eigenvector of A and v is an eigenvector of B .

Any $n \times n$ matrix A has n eigenvalues $\{\lambda_1, \dots, \lambda_n\}$ (not necessarily distinct), which are the roots of the n th degree polynomial equation

$$\det(A - \lambda I) = 0, \quad (1.2.3)$$

where “det” denotes the determinant. In example (1.2.2), the two eigenvalues of A are $\lambda_1 = 3$, $\lambda_2 = 1$, while B has a double eigenvalue ($\lambda_1 = \lambda_2 = 3$). We sometimes denote the set of eigenvalues of A by $\{\lambda_i(A)\}$.

Any nonzero multiple of an eigenvector is also an eigenvector, since $A(\gamma u) = \lambda(\gamma u)$ for any nonzero scalar γ ; thus, eigenvectors are often implicitly assumed to have Euclidean length one (see §1.3.1). A *general* matrix does not necessarily have n linearly independent eigenvectors. (However, see the discussion below of symmetric matrices.) For example, the matrix B of (1.2.2) has only a single eigenvector, of the form $(\gamma, 0)^T$. The eigenvalues and eigenvectors together comprise the *eigensystem* of a matrix.

A useful relation is that the sum of the diagonal elements of any square matrix A (called the *trace* of A) is equal to the sum of the eigenvalues:

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

In addition, the product of the eigenvalues is equal to the determinant of A :

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

Multiplying the relation (1.2.1) by any nonsingular matrix S , we have $SAu = S(\lambda u) = \lambda(Su)$. Since $S^{-1}S = I$, it follows that

$$SAu = SAS^{-1}Su = SAS^{-1}(Su) = \lambda(Su), \quad (1.2.4)$$

which shows that λ is an eigenvalue and Su is an eigenvector of SAS^{-1} . Two matrices are said to be *similar* if they have the *same eigenvalues*, and (1.2.4) verifies that SAS^{-1} is similar to A . (The matrix SAS^{-1} is called a *similarity transform* of A .) Similar matrices do not in general have the same eigenvectors.

If A is singular, there exists a nonzero vector x such that $Ax = 0$, which shows that a singular matrix has at least one zero eigenvalue. If A is nonsingular, all its eigenvalues are nonzero, and the eigenvalues of A^{-1} are the reciprocals of the eigenvalues of A .

It is sometimes convenient to have an explicit formula for the eigenvalues of a 2×2 matrix. (Since the eigenvalues are the roots of an n th degree polynomial, no such expression exists for the eigenvalues of a general matrix of order greater than four.) Using (1.2.3), the two eigenvalues of the matrix

$$T = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

are the roots of the quadratic equation $\lambda^2 - (a + d)\lambda + ad - bc = 0$, namely

$$\lambda_1, \lambda_2 = \frac{1}{2} \left(a + d \pm \sqrt{(a - d)^2 + 4bc} \right).$$

The eigensystems of *symmetric* matrices have two special properties. First, if A is symmetric, *all the eigenvalues of A are real*. Furthermore, a symmetric matrix always has n orthogonal eigenvectors that may be normalized to form an *orthonormal basis* for

$\mathbb{R}^n$. Let λ_i denote an eigenvalue of the symmetric matrix A , with v_i the corresponding normalized eigenvector, so that

$$Av_i = \lambda_i v_i, \quad i = 1, \dots, n. \quad (1.2.5)$$

Letting V be the matrix whose i th column is v_i , and Λ the diagonal matrix $\text{diag}(\lambda_1, \dots, \lambda_n)$, the equations in (1.2.5) may be written as $AV = V\Lambda$. Since V is orthogonal, multiplying on the right by V^T gives

$$A = V\Lambda V^T, \quad (1.2.6)$$

a form called the *spectral decomposition* of A . If an orthogonal matrix V and a diagonal matrix Λ can be found satisfying (1.2.6), the elements of Λ must be the eigenvalues of A , and the columns of V are the corresponding eigenvectors. We sometimes assume that the eigenvalues are ordered so that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$. The *spectral radius* of A , denoted by $\rho(A)$, is defined as

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

A square matrix A is said to be:

- positive definite if $u^T A u > 0$ for all $u \neq 0$;
- positive semidefinite if $u^T A u \geq 0$ for all u ;
- indefinite if $u^T A u > 0$ and $v^T A v < 0$ for some u, v .

Note that these definitions imply that a positive-definite matrix is also positive semidefinite (just as a positive number α satisfies $\alpha \geq 0$).

For any symmetric matrix, the positive-definiteness condition may be conveniently stated in terms of its eigenvalues. Let A be a symmetric matrix whose eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$, are ordered so that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$. Let $v_1, v_2, \dots, v_n$ denote the corresponding eigenvectors of A . It holds that

- (i) A is positive definite if and only if $\lambda_i > 0$ for all $1 \leq i \leq n$.
- (ii) If $\lambda_i \geq 0$ for $1 \leq i \leq n$ and there is at least one eigenvalue λ_j such that $\lambda_j = 0$, then A is positive semidefinite.
- (iii) An eigenvector v_j corresponding to a positive (negative) eigenvalue satisfies $v_j^T A v_j > 0$ ($v_j^T A v_j < 0$). An eigenvector corresponding to a zero eigenvalue is a null vector for A .
- (iv) The largest and smallest eigenvalues of A satisfy

$$\lambda_1 \geq \frac{x^T A x}{x^T x} \geq \lambda_n, \quad \text{for all } x \neq 0.$$

The *inertia* of a symmetric matrix A , denoted by $\text{In}(A)$, is defined as a triple of nonnegative integers

$$\text{In}(A) = (i_+(A), i_-(A), i_0(A)),$$

where $i_+(A)$ is the number of positive eigenvalues of A , $i_-(A)$ is the number of negative eigenvalues, and $i_0(A)$ is the number of zero eigenvalues. The inertia of A is equal to the inertia of B only if $i_+(A) = i_+(B)$, $i_-(A) = i_-(B)$ and $i_0(A) = i_0(B)$.

If C is any nonsingular matrix, *Sylvester's law of inertia* states that the inertia of C^TAC is the same as the inertia of A . (However, in general the eigenvalues of A and of C^TAC are different.)

1.2.2. The singular-value decomposition

Any $m \times n$ matrix A can be written as

$$A = USV^T, \quad (1.2.7)$$

where U is an $m \times m$ orthogonal matrix, V is an $n \times n$ orthogonal matrix, and S is an $m \times n$ diagonal matrix, $S = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_p)$, with $p = \min(m, n)$ and $\sigma_i \geq 0$, $i = 1, \dots, p$. The nonnegative numbers $\{\sigma_i\}$ are called the *singular values* of A , and (1.2.7) is called the *singular-value decomposition* (SVD). The singular value decomposition is arguably the most informative general representation of a matrix, and will be discussed in more detail in later chapters. The convention is usually adopted that $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p$, so that $\sigma_1(A)$ denotes the *largest* singular value of A .

If A has rank r and $r > 0$, then A has exactly r strictly positive singular values, so that $\sigma_r > 0$ and $\sigma_{r+1} = \dots = \sigma_p = 0$. If A has full rank, all its singular values are nonzero. The singular values of A are the square roots of the eigenvalues of A^TA (if $m \geq n$) or of AA^T (if $m < n$). If A is symmetric, its singular values are the absolute values of its eigenvalues.

1.3. Norms

To use terms like “smaller” or “larger” in our later work, we need to define how to measure the “size” of a vector or matrix. For scalars, the “natural” choice for the size of a number is its *absolute value*. The definition of *norms* provides a generalization of this association to vectors and matrices.

1.3.1. Vector norms

A *vector norm*, denoted by $\|\cdot\|$, satisfies the following three properties:

- (i) $\|x\| > 0$ for any nonzero vector x ;
- (ii) for any scalar γ , $\|\gamma x\| = |\gamma| \|x\|$;
- (iii) for any two vectors x and y , $\|x + y\| \leq \|x\| + \|y\|$ (the *triangle inequality*).

A norm is a continuous function of the elements of its argument. The definition of a norm extends in an obvious way to row vectors.

For a positive integer p , the p - or Hölder norm of an n -vector x (denoted by $\|x\|_p$) is defined as

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}, \quad (1.3.1)$$

and a finite limit exists as $p \rightarrow \infty$.

The three most common vector norms correspond to $p = 1, 2$ and ∞ in (1.3.1). When x is an n -vector, these norms are given by

- $\|x\|_1 = \sum_{i=1}^n |x_i|$ (the ℓ_1 or one-norm);
- $\|x\|_2 = \sqrt{x^T x}$ (the ℓ_2 , Euclidean or two-norm, also called the *length* or *Euclidean length* of x);
- $\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$ (the ℓ_∞ , max or infinity-norm).

For example, if

$$x = \begin{pmatrix} 1 \\ -2 \end{pmatrix}, \quad \text{then} \quad \|x\|_1 = 3, \quad \|x\|_2 = \sqrt{5} \quad \text{and} \quad \|x\|_\infty = 2. \quad (1.3.2)$$

The standard formula for the cosine of the angle θ between two nonzero vectors x and y is

$$\cos \theta = \frac{x^T y}{\|x\|_2 \|y\|_2}. \quad (1.3.3)$$

When x and y are *orthogonal*, i.e., $x^T y = 0$, relation (1.3.3) confirms the usual interpretation that x and y intersect at right angles. Since the cosine is bounded in magnitude by unity, (1.3.3) also gives the following useful *Cauchy-Schwarz inequality*:

$$-\|x\|_2 \|y\|_2 \leq x^T y \leq \|x\|_2 \|y\|_2. \quad (1.3.4)$$

For any nonzero vector x and norm $\|\cdot\|$, the *normalized* vector u such that u has the same direction as x and $\|u\| = 1$ is given by

$$u = \frac{1}{\|x\|} x = \frac{x}{\|x\|}. \quad (1.3.5)$$

(The final expression in (1.3.5) illustrates the compact notation x/α to mean $(1/\alpha)x$, where x is a vector and α is a scalar.)

Given any norm $\|\cdot\|$ we can define the function

$$\nu(x) = \max_{\substack{v \in \mathbb{R}^n \\ v \neq 0}} \frac{|x^T v|}{\|v\|}. \quad (1.3.6)$$

It can be shown that: (i) $\nu(x) \geq 0$ for all x , with $\nu(x) = 0$ if and only if $x = 0$; (ii) $\nu(\alpha x) = |\alpha| \nu(x)$ for any scalar α ; and (iii) $\nu(x + y) \leq \nu(x) + \nu(y)$ for all vectors x and y . It follows that $\nu(x)$ defines a vector norm, which is known as the vector norm *dual to* $\|\cdot\|$.

The dual norm of the two-norm is itself the two-norm. This follows from the Cauchy-Schwartz inequality

$$-\|v\|_2 \|x\|_2 \leq v^T x \leq \|v\|_2 \|x\|_2,$$

where both inequalities are exact for the vector $v^* = \pm x$. It follows that $v^* = \pm x$ solves the problem associated with $\nu(x)$, with

$$\nu(x) = \max_{\substack{v \in \mathbb{R}^n \\ v \neq 0}} \frac{|x^T v|}{\|v\|_2} = \frac{|x^T v^*|}{\|v^*\|_2} = \frac{|x^T x|}{\|x\|_2} = \|x\|_2.$$

The two-norm is said to be *self-dual*.

It can be shown that the dual of the Hölder norm $\|\cdot\|_p$ is $\|\cdot\|_q$, where $1/p + 1/q = 1$. This result implies that the dual of the dual of the Hölder norm is the original norm. It can be shown that $\lim_{p \rightarrow \infty} \|x\|_p = \max_{1 \leq i \leq n} |x_i| = \|x\|_\infty$, which implies that the one- and infinity-norms are dual to each other.

We often need the vector v^* that minimizes the ratio (1.3.6) defining $\nu(x)$. Consider the dual norm for the one-norm, which is the infinity norm. In this case, $v^* = \pm e_i$, where e_i is the i th coordinate vector with i defined such that $|x_i| = \max_{1 \leq j \leq n} |x_j|$. Then

$$\nu(x) = \max_{\substack{v \in \mathbb{R}^n \\ v \neq 0}} \frac{|x^T v|}{\|v\|_1} = \frac{|x^T v^*|}{\|v^*\|_1} = \frac{|x^T e_i|}{\|e_i\|_1} = |x_i| = \max_{1 \leq j \leq n} |x_j| = \|x\|_\infty.$$

Similarly, for the dual of the infinity norm, we have $v^* = \pm \eta$ where η is the vector

$$\eta_j = \begin{cases} +1 & \text{if } x_j \geq 0; \\ -1 & \text{if } x_j < 0. \end{cases}$$

Then $x^T \eta = \sum_{j=1}^n |x_j|$, with $\|\eta\|_\infty = 1$ and

$$\nu(x) = \frac{|x^T v^*|}{\|v^*\|_\infty} = \frac{|x^T \eta|}{\|\eta\|_\infty} = \|x\|_1.$$

Next we consider the so-called *ellipsoidal norm* $\|x\|_C = (x^T C x)^{1/2}$, where C is a positive-definite symmetric matrix. Since C is positive definite, there exists a nonsingular matrix M such that $C = M^T M$. As M is nonsingular, we have $x^T v = x^T M^{-1} M v = (M^{-T} x)^T (M v)$. The Cauchy-Schwartz inequality can be used to obtain the sequence of inequalities

$$\begin{aligned} |x^T v| &\leq \|M^{-T} x\|_2 \|M v\|_2 \\ &= (x^T M^{-1} M^{-T} x)^{1/2} (v^T M^T M v)^{1/2} \\ &= (x^T C^{-1} x)^{1/2} (v^T C v)^{1/2} \\ &= \|x\|_{C^{-1}} \|v\|_C. \end{aligned}$$

This establishes the bound $|x^T v|/\|v\|_C \leq \|x\|_{C^{-1}}$, which is achieved for the vector v^* such that $v^* = \pm C^{-1} x$. It follows that the dual of the ellipsoidal norm $\|x\|_C$ is another ellipsoidal norm $\|x\|_{C^{-1}}$ based on C^{-1} .

1.3.2. Matrix norms

A *matrix norm*, denoted as in the vector case by $\|\cdot\|$, satisfies the same three properties as a vector norm:

- (i) $\|A\| > 0$ for any nonzero matrix A ;
- (ii) $\|\gamma A\| = |\gamma| \|A\|$ for any scalar γ ;
- (iii) $\|A + B\| \leq \|A\| + \|B\|$.

It is also useful for a matrix norm to satisfy a fourth property, called *consistency*:

$$(iv) \|AB\| \leq \|A\| \|B\|.$$

Given any vector norm, an associated matrix norm can be defined as follows. Intuitively, the matrix A should have a “large” norm if its application to a nonzero vector x can produce a vector Ax whose norm is large relative to $\|x\|$. For any matrix A and vector norm $\|\cdot\|$, the *subordinate* or *induced* matrix norm $\|A\|$ is defined as

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}. \quad (1.3.7)$$

(Note that the norms on the right-hand side of this expression are *vector* norms.) Using (1.3.5), an equivalent (sometimes more convenient) definition of $\|A\|$ is:

$$\|A\| = \max_{\|u\|=1} \|Au\|.$$

A lower bound on $\|A\|$ can be obtained by computing the ratio of $\|Ax\|$ and $\|x\|$ for any specific nonzero vector x :

$$\|A\| \geq \frac{\|Ax\|}{\|x\|}. \quad (1.3.8)$$

Equality in (1.3.8) must be *achieved* for at least one nonzero vector x , for which $\|A\| \|x\| = \|Ax\|$.

The matrix norms subordinate to the one-, two- and infinity-norms for vectors are:

- $\|A\|_1 = \max_j \|a_j\|_1$ (the maximum absolute column sum);
- $\|A\|_2 = \sigma_1(A)$ (the largest singular value; see §1.2.2);
- $\|A\|_\infty = \max_i \|a'_i\|_1$ (the maximum absolute row sum).

A vector norm $\|\cdot\|$ and a matrix norm $\|\cdot\|'$ are said to be *compatible* or *consistent* if

$$\|Ax\| \leq \|A\|' \|x\|$$

for every A and x . A vector norm and its subordinate matrix norm are always compatible (see (1.3.8)).

An important matrix norm that is not subordinate to a vector norm is the *Frobenius* norm, defined for an $m \times n$ matrix A as

$$\|A\|_F = \left(\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2 \right)^{1/2},$$

which can be shown to satisfy

$$\|A\|_F^2 = \text{trace}(A^T A).$$

The Frobenius norm is essentially the Euclidean norm of its matrix argument treated as a “vector” with mn components. The Frobenius norm is *compatible* with the Euclidean vector norm, i.e., for any A and x ,

$$\|Ax\|_2 \leq \|A\|_F \|x\|_2.$$

Although different norms usually vary in value for a fixed vector or matrix (see (1.3.2)), they can nonetheless be regarded as “equivalent” in the following sense. For any two vector norms, say $\|\cdot\|$ and $\|\cdot\|'$, there exist positive constants c_l and c_u , depending only on the dimension of the vector, such that, for any vector x ,

$$c_l\|x\| \leq \|x\|' \leq c_u\|x\|. \quad (1.3.9)$$

A similar result also holds for any two matrix norms.

Relation (1.3.9) allows us to choose the most convenient norm in certain situations when the exact value of the norm is not crucial. For example, if a theorem states that the two-norm of a certain vector is bounded, we know from (1.3.9) that *any* norm of the vector is bounded. Similarly, the one-, infinity- or Frobenius norms may be preferred in matrix calculations because they are less expensive to compute than the two-norm.

1.4. Some Properties of Sets

In this class we will always be finding minimizers or zeros of functions that are defined on a finite-dimensional normed space. A normed vector space is essentially a set of points D with some method for measuring the distance between any two points x and y in D . Most often, we will consider the simplest case where D is the set of n vectors $(x_1, x_2, \dots, x_n)^T$, and the distance between two n -vectors x and y is the so-called Euclidean distance $(\sum_{i=1}^n (x_i - y_i)^2)^{1/2}$, which is just the two-norm of the vector $x - y$.

In general, the method of measuring the distance between any two points x and y in D is known as the “metric” associated with D .

We need to define the most common properties associated with subsets of D .

Definition 1.4.1. *Given a positive δ and a point $x_0 \in D$, we define $\mathcal{B}(x, \delta)$ to be the set of all $x \in D$ whose distance to x_0 is less than δ , i.e.,*

$$\mathcal{B}(x_0, \delta) = \{x \mid \|x - x_0\| < \delta\}.$$

We call $\mathcal{B}(x_0, \delta)$ the open ball of radius δ centered at x_0 .

In $\mathbb{R}^1$, the open ball of radius δ is the open interval $(x_0 - \delta, x_0 + \delta)$.

If we say that a certain property holds “in an open neighborhood” of x , we mean that there exists some $\delta > 0$ such that the property holds in an open ball $\mathcal{B}$ centered at x ; in this context, the *existence of the neighborhood* is important, but the actual value of δ is not.

Definition 1.4.2. *(Interior of a set.) Given a set S , a point x is an interior point of S if $x \in S$ and there exists an open ball centered at x that is entirely contained in S . The interior of S , denoted by $\text{int}(S)$, is the set of all interior points of S . ■*

An important property of an open or closed ball is that, given any two points $x, y \in \mathcal{B}$, every point z on a line joining x and y must also lie in $\mathcal{B}$. Sets with this property are called *convex sets*. A convex set satisfies the following definition.

Definition 1.4.3. *(Convex set.) A set S is convex if, for all $x, y \in S$,*

$$z = \theta y + (1 - \theta)x \in S, \quad \text{for all } \theta \in [0, 1]. \quad \blacksquare$$

Another interpretation of this definition comes from observing that the point may be written in the form $z = x + \theta(y - x)$ with $0 \leq \theta \leq 1$. This implies that a set is convex if all steps θ ($0 \leq \theta \leq 1$) from x along a vector in the direction of y must give a point that also lies in S .

Definition 1.4.4. Let S be a subset of a normed space D . A point $x \in D$ is called a limit point of S if there is a sequence $\{x_k\}$ of points of S that converges to x . The set of all limit points of S is called the closure of S .

An important property of a limit point x is that every open ball centered at x contains at least one point of S .

In many cases, the set S will consist of the members of a sequence of *approximate solutions* generated by an algorithm. This sequence is usually written as $\{x_k\}$.

The closure of the open ball $\mathcal{B}(x, \delta)$ is the set

$$\bar{\mathcal{B}}(x, \delta) = \{x \mid \|x - y\| \leq \delta\},$$

which is known as the *closed ball* of radius δ centered at x .

Definition 1.4.5. (*Closed set.*) A set S is said to be closed if $S = \bar{S}$, i.e., a set is closed if it contains all its limit points. ■

Definition 1.4.6. (*Boundary of a set.*) Given a set S , a point x is a boundary point of S if every neighborhood of x contains at least one point in S and at least one point not in S . The boundary of S is the collection of all boundary points of S . ■

It is straightforward to show that a closed set contains all its boundary points. The boundary of the closed ball $\bar{\mathcal{B}}(x, \delta)$ is the set of points y that lie on the “sphere” $\|y - x\| = \delta$.

Definition 1.4.7. Let S and D be two sets such that $S \subset D \subset \mathbb{R}^n$. The set S is said to be open (closed) relative to D if $S = D \cap N$, where N is some open (closed) set in $\mathbb{R}^n$.

An open set D in $\mathbb{R}^n$ is always open relative to $\mathbb{R}^n$. If $S \subset D$, and if D is open, then the set S is open relative to D since $S = S \cap D$.

Definition 1.4.8. (*Bounded set.*) A set S is bounded if there exists a positive number M such that

$$\|x - y\| \leq M, \quad \text{for all } x, y \in S. \quad \blacksquare$$

1.5. Convergence of Sequences

In studying a sequence of approximations to a solution, we typically consider two features: whether the elements of the sequence converge to the desired solution; and, if so, the “speed” of convergence. Rigorous definitions are required in order to make meaningful statements about these issues. Unless specified otherwise, every sequence is assumed to be infinite; however, non-finiteness may sometimes be stated for emphasis.

Definition 1.5.1. (*Convergence of a sequence.*) The sequence $\{x_k\}$, $k = 0, \dots$, converges to the value x^* if, for any $\epsilon > 0$, there exists an index K such that for all $k \geq K$,

$$\|x_k - x^*\| < \epsilon. \quad \blacksquare \tag{1.5.1}$$

When a sequence converges, the (necessarily unique) value x^* is called the *limit* of the sequence, since relation (1.5.1) is equivalent to stating that $\lim_{k \rightarrow \infty} x_k = x^*$. If we say simply that a sequence *converges*, we mean that some limit (whose value is unspecified and may be irrelevant) exists.

Definition 1.5.2. (*Subsequence*). Given a sequence $\{x_k\}$, a subsequence is a sequence formed by omitting some of the elements of the original sequence, without altering the order of the remaining elements. ■

A subsequence is often defined by specifying a set of *indices* $\{k_i\}$, $i = 0, 1, \dots$, of the selected elements, so that the subsequence is $\{x_{k_i}\}$. For example, if $\{x_k\}$ is the sequence of integers $x_k = k$, $k = 0, 1, 2, \dots$, then the *odd* integers $1, 3, \dots$, form a subsequence whose set of indices is $k_i = 2i + 1$.

Definition 1.5.3. (*Limit point*). A point x^* is a limit point of the sequence $\{x_k\}$ if there exists a subsequence that converges to x^* . ■

To illustrate the ideas behind these crucial definitions, consider three sequences. First, the sequence of integers $x_k = k$ does not converge and has no limit point. Second, the sequence $x_k = 2 + 2^{-k}$ converges to $x^* = 2$, as we now show. Given any $\epsilon > 0$ (no matter how small), an integer K exists such that $2^k > 1/\epsilon$ for all $k \geq K$. Hence, $2^{-k} < \epsilon$ for all $k \geq K$, so that

$$x_k - x^* = x_k - 2 = 2 + 2^{-k} - 2 = 2^{-k} < \epsilon,$$

and (1.5.1) is satisfied. Finally, the sequence

$$x_k = \begin{cases} 2 + 2^{-k} & k \text{ even;} \\ \frac{1}{2} & k \text{ odd,} \end{cases} \quad (1.5.2)$$

does not converge, but has two limit points (2 and $\frac{1}{2}$) since the subsequences corresponding to even and odd k converge to these values.

We next define the property of boundedness, and then give three related theorems.

Definition 1.5.4. (*Boundedness of a sequence*). The scalar sequence $\{x_k\}$ is bounded below if there exists a constant m such that $x_k \geq m$ for all k , and bounded above if there exists a constant M such that $x_k \leq M$ for all k . If a sequence is bounded above and below, the sequence is bounded. If no constant M exists such that $x_k \leq M$ for all k , $\{x_k\}$ is unbounded above, with an analogous definition for “unbounded below”. ■

For example, the sequence $x_k = k$, $k = 0, 1, \dots$, is bounded below by zero and unbounded above, whereas the sequence (1.5.2) is bounded.

Theorem 1.5.1. (*Existence of a limit point*). Every bounded scalar sequence has at least one limit point. ■

Theorem 1.5.2. (*Existence of a greatest lower or least upper bound*). If S is a non-empty set of real numbers that is bounded below, then S has a greatest lower bound. If S is a non-empty set of real numbers that is bounded above, then S has a least upper bound. ■

Sequences whose elements become consistently larger or smaller are the subject of the following definition and theorem.

Definition 1.5.5. (*Monotonicity of a sequence*). The scalar sequence $\{x_k\}$ is decreasing if $x_{k+1} < x_k$ for all k , and is non-increasing if $x_{k+1} \leq x_k$ for all k . Similarly, $\{x_k\}$ is increasing if $x_{k+1} > x_k$ for all k , and non-decreasing if $x_{k+1} \geq x_k$ for all k . A sequence that is either non-increasing or non-decreasing is said to be monotonic. ■

Theorem 1.5.3. (*Convergence of a monotonic sequence*). If $\{x_k\}$ is a non-increasing scalar sequence that is bounded below, then $\{x_k\}$ converges to its greatest lower bound. Similarly, a non-decreasing sequence that is bounded above converges to its least upper bound. ■

1.5.1. Rates of convergence

A vital concept in the analysis of iterative algorithms is the *order* or *rate* of convergence of the sequence of iterates. (We use the terms “order of convergence” and “rate of convergence” interchangeably.) Broadly speaking, the order of convergence indicates *how fast* convergence occurs; the larger the order, the faster the convergence. Since our concern is almost entirely with sequences of a somewhat restricted nature, we shall not state the most general definitions of the rate of convergence.

Definition 1.5.6. (*Lower bound on the order of convergence*). Let $\{x_k\}$ be a sequence that converges to x^* , and let $r \geq 1$. We say that $\{x_k\}$ converges to x^* with order at least r if there exist a constant $\beta_r \geq 0$ and a positive integer K such that, for all $k \geq K$,

$$\|x_{k+1} - x^*\| \leq \beta_r \|x_k - x^*\|^r. \quad \blacksquare \quad (1.5.3)$$

The following definition is a special case of the widely used concept of *q-order*.

Definition 1.5.7. (*Order of convergence; the asymptotic error constant*). Let $\{x_k\}$ be a sequence that converges to x^* , and let $r \geq 1$. If the sequence

$$q_k = \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^r} \quad (1.5.4)$$

is well defined and converges to a finite nonzero limit γ_r , then $\{x_k\}$ converges to x^* with order r , and γ_r is the asymptotic error constant. ■

When $r = 1$, the order of convergence is said to be *linear*, and it must hold that $\gamma_r \leq 1$. When $r = 2$ or 3 , the rate of convergence is *quadratic* or *cubic*, respectively.

When Definition 1.5.7 applies, r is unique. Let $\bar{r}$ be a number purporting to be the rate of convergence. A “rule of thumb” is that the limit of the sequence $\{q_k\}$ (1.5.4) is zero if $\bar{r} < r$, and that the values of q_k are unbounded above if $\bar{r} > r$. For example, consider the sequence $x_k = 2^{-k}$, whose limit is $x^* = 0$. Since $x_k \neq x^*$ for any k , the sequence $\{q_k\}$ of (1.5.4) is well defined, and

$$q_k = \frac{x_{k+1}}{x_k^r} = \frac{2^{-k-1}}{2^{-rk}} = 2^{rk-k-1}. \quad (1.5.5)$$

If $r = 1$, $q_k = \frac{1}{2}$ for all k . On the other hand, suppose that $r > 1$. Writing $r = 1 + \delta$, where $\delta > 0$,

$$q_k = 2^{\delta k - 1},$$

and q_k becomes arbitrarily large as k increases. We conclude that 2^{-k} converges to zero with order one and asymptotic error constant $\frac{1}{2}$.

A higher rate of convergence implies (ultimately) faster convergence because of the following relationship between consecutive powers of sufficiently small numbers.

Property 1.5.1. *Given any constant $M > 0$, there exists $\bar{\epsilon} > 0$ such that*

$$M\epsilon^{m+1} < \epsilon^m \quad \text{for all } 0 < \epsilon < \bar{\epsilon}. \quad (1.5.6)$$

It is straightforward to see that (1.5.6) will hold if $\bar{\epsilon} = 1/M$.

Convergence that is “faster than linear” is given a special name, and may be characterized without knowing the precise value of r .

Definition 1.5.8. (*q-superlinear convergence*). *Let $\{x_k\}$ be a sequence that converges to x^* . If there exists a sequence $\{\beta_k\}$ that converges to zero and an index K such that*

$$\|x_{k+1} - x^*\| \leq \beta_k \|x_k - x^*\|$$

for all $k > K$, then $\{x_k\}$ converges q-superlinearly to x^* . ■

If a value of r satisfying Definition 1.5.7 exists, the corresponding sequence converges superlinearly when $r > 1$. However, it is important to realize that superlinearly convergent sequences exist for which there is no value of r satisfying Definition 1.5.7. For example, consider the sequence

$$x_k = 2^{-2^{2^k}}, \quad \text{whose first three elements are } \frac{1}{4}, \frac{1}{16}, \frac{1}{65536}.$$

Although Definition 1.5.8 applies and $\{x_k\}$ therefore converges superlinearly to zero, the associated sequence $\{q_k\}$ in Definition 1.5.7 converges to zero for any fixed value of r , and hence the order of convergence is not defined.

It is sometimes convenient to analyze the error in an approximation using the following definition.

Definition 1.5.9. (*Order notation*). *Let f be an arbitrary function of a positive variable h ; let $p \geq 0$ and $\epsilon < 1$. Suppose that there exists a finite constant M , independent of h , such that for all $h < \epsilon$, it holds that*

$$|f(h)| \leq Mh^p; \quad \text{then we write } f = O(h^p),$$

and say “ f is of order h^p ” (or “ f is big oh of h^p ”). If

$$\lim_{h \rightarrow 0} \frac{f(h)}{h^p} = 0, \quad \text{then we write } f = o(h^p),$$

and say “ f is little oh of h^p ”. Finally, if there exists a nonzero constant L such that

$$\lim_{h \rightarrow 0} \frac{f(h)}{h^p} = L, \quad \text{then we write } f \sim Lh^p. \quad \blacksquare$$

When $p = 0$, the above relations are written as $O(1)$ and $o(1)$, since $h^0 = 1$ for any nonzero h . If $f = O(1)$, $|f|$ is bounded. If $f = o(1)$, then $\lim_{h \rightarrow 0} f(h) = 0$. Broadly speaking, $a = o(b)$ means that a is going to zero faster than b , and $a = O(b)$ means that a and b are going to zero at the same “speed”.

1.6. Properties of Univariate Functions

The “fastest” methods for zero-finding and optimization rely on properties such as continuity and differentiability that allow us to deduce information about a function f at a “new” point from its behavior at “old” points.

In characterizing intervals on the real line, we use the following standard notation: “ $(a$ ” indicates that a is the left endpoint of an interval that is *open* on the left, i.e., the interval includes points that lie to the right of a , *not including* a ; whereas “ $[a$ ” means that the interval is *closed* on the left, i.e., includes a as well as points to the right of a . The notations “ $b)$ ” and “ $]b$ ” have a similar meaning for the right endpoints. We sometimes denote intervals by the letter $\mathcal{I}$. If $\mathcal{I}$ is the interval $[a, b]$, $|\mathcal{I}|$ means the *length* of $\mathcal{I}$, namely $b - a$.

If an interval exists with endpoints x_1 and x_2 , where $x_1 \neq x_2$, the statement “ $x \in \text{int}(x_1, x_2)$ ” means that x lies in (x_1, x_2) if $x_1 < x_2$ and in (x_2, x_1) otherwise.

The following notation conveys the continuity properties of a function and its derivatives.

Definition 1.6.1. (The notation C^k). Let f be a function and k a nonnegative integer. The statement that $f \in C^k$ (in an interval) means that f and its first k derivatives are continuous (in the interval). ■

To illustrate, $f_1(x) = |x|$ is continuous everywhere, but its first derivative is discontinuous at $x = 0$, so that $f_1 \in C^0$. In contrast, $f_2(x) = x^2$ is in C^∞ , since f_2 itself and all its derivatives are continuous everywhere. If $f \in C^m$, f is also in C^s for $0 \leq s < m$. If $f \in C^1$, f is said to be *continuously differentiable*; if $f \in C^2$, f is *twice-continuously differentiable*.

The following theorems will be used throughout the remainder of the book, and are grouped here for reference.

Theorem 1.6.1. (The Intermediate Value Theorem). Let f be a continuous function in $[a, b]$. If $f(a) \neq f(b)$, then f assumes every value between $f(a)$ and $f(b)$ in $[a, b]$. ■

The Intermediate Value Theorem does not state that f assumes *only* values between $f(a)$ and $f(b)$, but rather that f assumes at least this range of values.

Theorem 1.6.2. (The Extreme Value Theorem). Let f be a continuous function in $[a, b]$. Then

- (i) f is bounded in $[a, b]$; and
- (ii) f attains its maximum and minimum in $[a, b]$, i.e., there exists $\xi \in [a, b]$ such that $f(\xi) \geq f(x)$ for all $x \in [a, b]$, and there exists $\eta \in [a, b]$ such that $f(\eta) \leq f(x)$ for all $x \in [a, b]$. ■

It follows from this result and the Intermediate-Value Theorem that a continuous function f assumes every value between its maximum and minimum in $[a, b]$.

Theorem 1.6.3. (The Mean-Value Theorem of Differential Calculus). Let f be a function that is continuous in $[a, b]$ and differentiable in (a, b) . Then there exists at least one point $\xi \in (a, b)$ such that

$$f'(\xi) = \frac{f(b) - f(a)}{b - a}. \quad \blacksquare$$

This theorem may be stated in terms of an integral as follows.

Theorem 1.6.4. (*The Integral Mean Value Theorem*). Let f be a function that is continuous in $[a, b]$ and differentiable in (a, b) . Then

$$f(b) = f(a) + \int_a^b f'(z) dz. \blacksquare$$

Theorem 1.6.5. (*The Nonzero Derivative Theorem*). Let f be a function that is continuously differentiable in $[a, b]$, and assume that $f'(\xi) \neq 0$ for some point $\xi \in (a, b)$. Then

- (i) if $f'(\xi) > 0$, there exists δ satisfying $0 < \delta < \min(\xi - a, b - \xi)$ such that $f(x) < f(\xi) < f(z)$ for all $\xi - \delta < x < \xi$ and $\xi < z < \xi + \delta$;
- (ii) if $f'(\xi) < 0$, there exists δ satisfying $0 < \delta < \min(\xi - a, b - \xi)$ such that $f(x) > f(\xi) > f(z)$ for all $\xi - \delta < x < \xi$ and $\xi < z < \xi + \delta$. $\blacksquare$

Theorem 1.6.6. (*Taylor's theorem with remainder*). Let $h > 0$, and assume that $f \in C^p$ in $[x, x + h]$. Then there exists θ satisfying $0 \leq \theta \leq 1$ such that

$$\begin{aligned} f(x + h) = f(x) + hf'(x) + \frac{1}{2}h^2f''(x) + \cdots \\ + \frac{1}{(p-1)!}h^{p-1}f^{(p-1)}(x) + R_p, \end{aligned} \quad (1.6.1)$$

where the remainder term R_p is given by

$$R_p = \frac{1}{p!}h^p f^{(p)}(x + \theta h) \quad (1.6.2)$$

and $f^{(k)}$ denotes the k th derivative of f . $\blacksquare$

It is often convenient to use an integral form of the remainder term.

Theorem 1.6.7. (*Taylor's theorem with integral remainder*). Let $h > 0$, and assume that $f \in C^p$ in $[x, x + h]$. Then

$$\begin{aligned} f(x + h) = f(x) + hf'(x) + \frac{1}{2}h^2f''(x) + \cdots \\ + \frac{1}{(p-1)!}h^{p-1}f^{(p-1)}(x) + R_p, \end{aligned} \quad (1.6.3)$$

where the remainder term R_p is given by

$$R_p = \frac{1}{(p-1)!}h^p \int_0^1 (1 - \xi)^{p-1} f^{(p)}(x + \xi h) d\xi \quad (1.6.4)$$

and $f^{(k)}$ denotes the k th derivative of f . $\blacksquare$

The Taylor-series expansion (1.6.1) is most commonly written with a linear or quadratic remainder term, i.e.,

$$f(x+h) = f(x) + hf'(x + \theta_1 h) \quad (1.6.5)$$

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(x + \theta_2 h) \quad (1.6.6)$$

where $0 < \theta_1 < 1$ and $0 < \theta_2 < 1$.

Equivalent first- and second-order expansions with an integral remainder term are

$$f(x+h) = f(x) + h \int_0^1 f'(x + \xi h) d\xi, \quad (1.6.7)$$

$$\text{and } f(x+h) = f(x) + hf'(x) + h^2 \int_0^1 (1-\xi) f''(x + \xi h) d\xi. \quad (1.6.8)$$

Alternative forms for the error terms of these Taylor-series expansions involve the integral of the *difference* between the next higher derivative. In the first-order case, we include an additional term $hf'(x)$ on the right-hand side of (1.6.7) to obtain

$$f(x+h) = f(x) + hf'(x) + h \int_0^1 (f'(x + \xi h) - f'(x)) d\xi. \quad (1.6.9)$$

Note that this form expresses the remainder term in terms of the “sum” of the variation of the first derivative as ξ varies between 0 and 1. Hence, if $f'(x)$ varies dramatically, we should expect the remainder term to be large.

A similar “difference” form for the remainder in the second-order expansion can be derived by including an additional term $\frac{1}{2}h^2 f''(x)$ on the right-hand side of (1.6.8), giving

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(x) + h^2 \int_0^1 (1-\xi)(f''(x + \xi h) - f''(x)) d\xi.$$

Definition 1.6.2. (*Multiplicity of a zero*). Let $f \in C^k$ and $f(x^*) = 0$. Then x^* is a zero (root) of f of multiplicity k if $f^{(i)}(x^*) = 0$, $i = 1, \dots, k-1$, and $f^{(k)}(x^*) \neq 0$, where $f^{(i)}$ denotes the i th derivative of f . ■

For example, if $f(x) = x^2 - 1$, $x^* = 1$ is a zero of multiplicity one, since $f'(x^*) = 2x^* = 2 \neq 0$. In contrast, if $f(x) = (x-1)^2$, then $x^* = 1$ is a zero of multiplicity two, since $f'(1) = 0$ but $f''(1) = 2 \neq 0$. A zero of multiplicity one is called a *simple* zero.

1.7. Properties of Functions of Many Variables

Let $F_i(x)$ denote a scalar-valued function of n variables $x_1, x_2, \dots, x_n$. Let $F(x)$ denote the m -vector whose i th component is $F_i(x)$, i.e.,

$$F(x) = \begin{pmatrix} F_1(x) \\ F_2(x) \\ \vdots \\ F_m(x) \end{pmatrix}.$$

A scalar-valued function F of a single variable x is *differentiable* if there exists a number $F'(x)$ such that

$$\lim_{h \rightarrow 0} \frac{1}{h} \{F(x+h) - F(x) - F'(x)h\} = 0.$$

If $F'(x)$ exists, it is unique and is known as the *first derivative* of $F(x)$.

This definition can be extended to define the derivative of a general vector-valued function. In the definition used here, the quantity h is a *vector* whose norm goes to zero.

Definition 1.7.1. (*Derivative of a vector-valued function.*) A vector-valued function $F : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ is differentiable at a point $x \in \text{int}(D)$ if there exists an $m \times n$ matrix $F'(x)$ such that

$$\lim_{h \rightarrow 0} \frac{1}{\|h\|} \{F(x+h) - F(x) - F'(x)h\} = 0.$$

If $F'(x)$ exists, it is unique and is known as the first derivative of $F(x)$. ■

An explicit matrix representation of $F'(x)$ may be obtained in terms of the first partial derivatives of F . If the partial derivative $\partial F_i / \partial x_j$ of the i th element $F_i(x)$ with respect to the j th variable x_j exists, then

$$\lim_{\delta \rightarrow 0} \frac{1}{\delta} \left\{ F_i(x + \delta e_j) - F_i(x) - \frac{\partial F_i}{\partial x_j} \delta \right\} = 0.$$

By choosing the vector h of Definition 1.7.1 as δe_j , where e_j is the j th column of the identity matrix, it is evident that the ij th element of $F'(x)$ is the ij th element of the following $m \times n$ matrix:

$$F'(x) = \begin{pmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \cdots & \frac{\partial F_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial F_m}{\partial x_1} & \frac{\partial F_m}{\partial x_2} & \cdots & \frac{\partial F_m}{\partial x_n} \end{pmatrix}.$$

This matrix of partial derivatives is known as the *Jacobian* and is often denoted by $J(x)$. We shall use the notation $F'(x)$ and $J(x)$ interchangeably.

As might be expected, the Jacobian of a linear function is constant, as in the one-variable case. In particular, if $F(x) = Ax - b$, with A an $m \times n$ matrix with constant elements, then $F'(x)$ is the $m \times n$ matrix A . In general, however, the elements of the Jacobian are themselves functions of x . For example, if $x \in \mathbb{R}^3$ and $F(x)$ is the two-vector

$$F(x) = \begin{pmatrix} F_1(x) \\ F_2(x) \end{pmatrix} = \begin{pmatrix} x_1^2 + x_2^2 + x_3^2 - 1 \\ e^{x_1} - x_2 - \cos x_3 \end{pmatrix},$$

then $F'(x)$ is the 2×3 matrix

$$J(x) = \begin{pmatrix} 2x_1 & 2x_2 & 2x_3 \\ e^{x_1} & -1 & \sin x_3 \end{pmatrix}.$$

1.7.1. Scalar-valued multivariate functions

A scalar-valued function F (such as the objective function in an optimization problem) is an example of a 1×1 vector-valued function. Applying the definition of the first derivative, we see that the first derivative is a *row vector* of first partial derivatives:

$$f'(x) = \left(\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \cdots \quad \frac{\partial f}{\partial x_n} \right).$$

One of the simplest examples of a scalar-valued function is the *affine function* $f(x) = a^T x - \beta = \sum_{j=1}^n a_j x_j - \beta$, where a be an n -vector and β be a scalar. From the above discussion, we may deduce that the derivative of an affine function is

$$f'(x) = (a_1 \quad a_2 \quad \cdots \quad a_n).$$

A *quadratic function* has the form $f(x) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n h_{ij} x_i x_j$, where the n^2 components h_{ij} are constant. If H is the matrix whose (i, j) th element is h_{ij} , then F may be written in the matrix-vector form $f(x) = \frac{1}{2} x^T H x$. The derivative of this function is given by $f'(x) = \frac{1}{2} x^T (H + H^T)$.

For a given constant σ , the set of points $\mathcal{S}$ such that $\mathcal{S} = \{x \mid f(x) = \sigma\}$ defines a *level surface* or *level curve* in $\mathbb{R}^n$. The column vector whose components are the first partial derivatives of F evaluated at x is normal to this surface at x . This column vector is known as the *gradient vector*, and sometimes denoted by $\nabla f(x)$, or just $g(x)$.

Definition 1.7.2. If $f(x)$ is a scalar-valued function, its gradient is the transpose of the derivative $f'(x)$. Thus, the gradient is the column vector of partial derivatives:

$$\nabla f(x) \triangleq f'(x)^T = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}.$$

■

The gradients of the affine function $f(x) = a^T x - \beta$ and quadratic function $\frac{1}{2} x^T H x$, are $\nabla f(x) = a$ and $\nabla f(x) = \frac{1}{2} (H + H^T)x$, respectively.

Definition 1.7.3. If $f(x)$ is a scalar-valued function, its Hessian $\nabla^2 f(x)$ is the Jacobian of the gradient vector $\nabla f(x)$. Thus, the Hessian is the $n \times n$ matrix of mixed second-order partial derivatives of f :

$$\nabla^2 f(x) \triangleq (\nabla f(x))' = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_2 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_2} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \frac{\partial^2 f}{\partial x_2 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}.$$

■

If the first- and second-order partial derivatives $\partial f/\partial x_i$, $\partial f/\partial x_j$ and $\partial^2 f/\partial x_i \partial x_j$ exist and are continuous, then $\partial^2 f/\partial x_i \partial x_j = \partial^2 f/\partial x_j \partial x_i$ and the Hessian is symmetric.

The affine function $f(x) = a^T x - \beta$ and quadratic function $f(x) = \frac{1}{2} x^T H x$, have Hessians $\nabla^2 f(x) = 0$, and $\nabla^2 f(x) = \frac{1}{2}(H + H^T)$, respectively. If H is symmetric, i.e., $h_{ij} = h_{ji}$, then $\nabla^2 f(x) = H$. For a general nonlinear function, the components of the Hessian are functions of x . For example, the function $f(x) = x_1 x_2^2 + 3x_1^2 x_2$ has gradient and Hessian

$$\nabla f(x) = \begin{pmatrix} x_2^2 + 6x_1 x_2 \\ 2x_1 x_2 + 3x_1^2 \end{pmatrix} \quad \text{and} \quad \nabla^2 f(x) = \begin{pmatrix} 6x_2 & 2x_2 + 6x_1 \\ 2x_2 + 6x_1 & 2x_1 \end{pmatrix}.$$

Note that the Hessian is symmetric, as is the usual case.

1.7.2. The chain rule and its applications

One of the most useful results in multivariate optimization is the *chain rule*. The chain rule is a mechanism that allows the simple evaluation of complicated derivatives.

Result 1.7.1. *Let F and G be vector-valued functions such that F maps vectors in $\mathbb{R}^p$ to vectors in $\mathbb{R}^q$ and G maps vectors in $\mathbb{R}^q$ to vectors in $\mathbb{R}^s$. Consider the composite function*

$$H(x) = G(F(x)),$$

that maps vectors in $\mathbb{R}^p$ to vectors in $\mathbb{R}^s$. If F and G are differentiable, then H is differentiable and the derivative of $H(x)$ is given by $H'(x) = G'(F(x))F'(x)$. ■

Essentially, the chain rule says that the Jacobian of H is the product of the Jacobians of G and F . However, it is important to recognize that the *order* of this product is crucial since both G and F are generally rectangular and may only be multiplied together in the order stated.

One of the most important applications of the chain rule concerns the calculation of the derivatives of a scalar-valued function along a direction p . Given any point $\bar{x} \in \mathbb{R}^n$, we shall be interested in the behavior of f as we move along a direction p that starts at $\bar{x}$. Let the *scalar-valued* function $\psi(\alpha)$ denote the restriction of $f(x)$ along the direction p , i.e.,

$$\psi(\alpha) \triangleq f(\bar{x} + \alpha p), \quad (1.7.1)$$

where $\bar{x}$ and p are held constant and the scalar α is allowed to vary. The function $\psi(\alpha)$ represents the univariate variation of f as we move along a linear path starting at $\bar{x}$. The chain rule allows us to define the derivatives of ψ with respect to α , i.e., the scalar quantities

$$\psi'(\alpha), \quad \psi''(\alpha), \quad \dots \quad \text{etc.}$$

To apply the chain rule it is necessary to notice that ψ is a composite map of the form $G(F(x))$ in which $F: \mathbb{R}^1 \rightarrow \mathbb{R}^n$ maps α into $\bar{x} + \alpha p$, and $G: \mathbb{R}^n \rightarrow \mathbb{R}^1$ maps $\bar{x} + \alpha p$ to $f(\bar{x} + \alpha p)$. It follows that the derivative of ψ is

$$\psi'(\alpha) = f'(\bar{x} + \alpha p) \frac{d}{d\alpha} (\bar{x} + \alpha p) = \nabla f(\bar{x} + \alpha p)^T p. \quad (1.7.2)$$

Note that the order of this product is crucial. The quantity $\nabla f(\bar{x} + \alpha p)^T p$ is known as the *directional derivative* of f along p at $\bar{x} + \alpha p$ (often we just say “directional derivative” when the position and direction are obvious). The directional derivative gives the slope of f along the direction p .

For the second derivative, we use the chain rule to obtain the derivative of $\nabla f(\bar{x} + \alpha p)^T p = p^T \nabla f(\bar{x} + \alpha p)$ with respect to α . Using the fact that the derivative of the gradient is the Hessian, we obtain

$$\psi''(\alpha) = p^T \frac{d}{d\alpha} \nabla f(\bar{x} + \alpha p) \frac{d}{d\alpha} (\bar{x} + \alpha p) = p^T \nabla^2 f(\bar{x} + \alpha p) p. \quad (1.7.3)$$

The quantity $p^T \nabla^2 f(\bar{x} + \alpha p) p$ is known as the *curvature* of f along p at $\bar{x} + \alpha p$ (or just the “curvature” for short).

1.7.3. Taylor-series for multivariate functions

Given these expressions for $\psi'(\alpha)$ and $\psi''(\alpha)$, the first- and second-order Taylor-series expansions (1.6.6) and (1.6.6) for the univariate function $\psi(\alpha)$ may be written as $\psi(\alpha) = \psi(0) + \alpha \psi'(\theta_1 \alpha)$, and $\psi(\alpha) = \psi(0) + \alpha \psi'(0) + \frac{1}{2} \alpha^2 \psi''(\theta_2 \alpha)$, where θ_1 and θ_2 are scalars such that $0 < \theta_1 < 1$ and $0 < \theta_2 < 1$. If these expansions are written in terms of f , we obtain

$$\begin{aligned} f(\bar{x} + \alpha p) &= f(\bar{x}) + \alpha \nabla f(\bar{x} + \theta_1 \alpha)^T p, \\ f(\bar{x} + \alpha p) &= f(\bar{x}) + \alpha \nabla f(\bar{x})^T p + \frac{1}{2} \alpha^2 p^T \nabla^2 f(\bar{x} + \theta_2 \alpha p) p. \end{aligned}$$

The equivalent first- and second-order expansions with an integral remainder term are

$$\begin{aligned} f(\bar{x} + \alpha p) &= f(\bar{x}) + \alpha \int_0^1 \nabla f(\bar{x} + \xi \alpha)^T p d\xi, \\ f(\bar{x} + \alpha p) &= f(\bar{x}) + \alpha \nabla f(\bar{x})^T p + \alpha^2 \int_0^1 p^T \nabla^2 f(\bar{x} + \xi \alpha p) p (1 - \xi) d\xi. \end{aligned}$$

1.8. Results from Matrix Theory

Lemma 1.8.1. *Let $\|\cdot\|$ be any norm on $\mathbb{R}^{n \times n}$ such that $\|AB\| \leq \|A\| \|B\|$ and $\|I\| = 1$. If $\|E\| < 1$ then $(I + E)^{-1}$ exists and*

$$\|(I + E)^{-1}\| \leq \frac{1}{1 - \|E\|}. \quad (1.8.1)$$

If A is nonsingular and $\|A^{-1}(B - A)\| \leq 1$, then B is nonsingular and

$$\|B^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}(B - A)\|}. \quad (1.8.2)$$

Proof. Let λ be any eigenvalue of E with associated normalized eigenvector v . Then $Ev = \lambda v$ and $(I + E)v = (1 + \lambda)v$, so that $1 + \lambda$ is an eigenvalue of $I + E$. We must show that $1 + \lambda \neq 0$. Taking norms in $Ev = \lambda v$ gives

$$|\lambda| = \|\lambda v\| \leq \|E\| \|v\| = \|E\| < 1.$$

If $\|E\| < 1$, we must have $1 + \lambda \neq 0$ and $I + E$ is nonsingular.

For the second part, define $C = (I + E)^{-1}$. Taking norms in the matrix identity $(I + E)C = I$ gives

$$\begin{aligned} 1 = \|I\| &= \|(I + E)C\| = \|C + CE\| \geq \|C\| - \|CE\| \\ &\geq \|C\| - \|C\| \|E\| = \|C\|(1 - \|E\|). \end{aligned}$$

This implies $\|(I + E)^{-1}\| \leq 1/(1 - \|E\|)$, as required.

For the final part, we define $E = A^{-1}(B - A)$, so that

$$B = A(I + E).$$

By assumption, $\|E\| < 1$, and $(I + E)$ is nonsingular. As A is also nonsingular, so is B , and

$$B^{-1} = (I + E)^{-1}A^{-1}.$$

Taking norms and using (1.8.1) gives (1.8.2) ■

Corollary 1.8.1. (*The Banach Lemma.*) Let A be nonsingular, with $\|A\| \leq \mu$. If $\|A - C\| \leq \delta$ and $\mu\delta < 1$, then C is nonsingular, and

$$\|C^{-1}\| \leq \frac{\mu}{1 - \mu\delta}.$$

1.9. Properties of Convex Functions

When finding minimizers or zeros of functions we often need to be able to predict the behavior of a function from its behavior at a few sample points. In the general case, we have to use Taylor-series expansions. Such predictions are made much simpler when we deal with the class of so-called *convex* functions.

Definition 1.9.1. (*Convexity.*) A function is convex on an interval $[a, b]$ if

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y), \quad (1.9.1)$$

for all x, y in $[a, b]$, and $0 < \theta < 1$. ■

In geometric terms, this definition implies that if f is convex, the value of f at any point $z \in \text{int}(x, y)$ must lie below the chord joining $f(x)$ and $f(y)$.

A function is *strictly convex* if (1.9.1) holds with strict inequality.

Theorem 1.9.1. Assume that $f \in C^1$ in an interval $[a, b]$. The function f is convex on $[a, b]$ if and only if $f'(x)(y - x) \leq f(y) - f(x)$ for all x and y in $[a, b]$.

Proof. First we show that if $f(y) - f(x) \geq f'(x)(y - x)$ for all x and y in $[a, b]$, then f must be convex.

Let $0 \leq \theta \leq 1$ and define $z = \theta x + (1 - \theta)y$. Clearly, z lies in the interval $[a, b]$. By assumption, we have

$$f(x) - f(z) \geq f'(z)(x - z) \quad \text{and} \quad f(y) - f(z) \geq f'(z)(y - z). \quad (1.9.2)$$

Adding θ times the first inequality to $(1 - \theta)$ times the second inequality, we obtain

$$\begin{aligned}\theta f(x) + (1 - \theta)f(y) - f(z) &= \theta(f(x) - f(z)) + (1 - \theta)(f(y) - f(z)) \\ &\geq \theta f'(z)(x - z) + (1 - \theta)f'(z)(y - z).\end{aligned}$$

The inequalities (1.9.2) give

$$\theta f(x) + (1 - \theta)f(y) - f(z) \geq f'(z)(\theta(x - z) + (1 - \theta)(y - z)) = 0,$$

since $z = \theta x + (1 - \theta)y$. We have shown that

$$\theta f(x) + (1 - \theta)f(y) \geq f(z) = f(\theta x + (1 - \theta)y)$$

so that f must be convex, as required.

Conversely, assume that f is convex in $[a, b]$. We must show that $f(y) - f(x) \geq f'(x)(y - x)$ for all x and y in $[a, b]$. If h is any number in $[0, 1]$, $x + h(y - x)$ will lie in $\text{int}(x, y)$. Moreover, since $x + h(y - x) = x(1 - h) + hy$, we have

$$f(x + h(y - x)) = f(x(1 - h) + hy) \leq (1 - h)f(x) + hf(y),$$

from the definition of the convexity of f . Rearrangement of this equation gives

$$f(x + h(y - x)) - f(x) \leq h(f(y) - f(x)),$$

so that dividing both sides of this inequality by h leads to

$$\frac{1}{h} [f(x + h(y - x)) - f(x)] \leq f(y) - f(x).$$

Since $\lim_{h \rightarrow 0} (1/h) [f(x + h(y - x)) - f(x)] = f'(x)(y - x)$, we have

$$f'(x)(y - x) \leq f(y) - f(x),$$

as required. ■

Theorem 1.9.2. Assume that $f \in C^1$ in an interval $[a, b]$. The function f is convex on $[a, b]$ if and only if $(f'(y) - f'(x))(y - x) \geq 0$ for all x and y in $[a, b]$.

Proof. If f is convex, we may use Theorem 1.9.1 to obtain the inequalities

$$f(y) - f(x) \geq f'(x)(y - x) \quad \text{and} \quad f(x) - f(y) \geq f'(y)(x - y).$$

Adding these inequalities together gives $(f'(y) - f'(x))(y - x) \geq 0$, as required.

Conversely, suppose that

$$(f'(y) - f'(x))(y - x) \geq 0 \quad \text{for all } x \text{ and } y \text{ in } [a, b].$$

The Mean-Value Theorem (Theorem 1.6.3) gives

$$f(y) - f(x) = f'(u)(y - x), \tag{1.9.3}$$

where $u = x + h(y - x)$ with $0 < h < 1$.

Consider the inequality

$$(f'(u) - f'(x))(y - x) = \frac{1}{h} ((f'(u) - f'(x))(u - x)) \geq 0,$$

by assumption.

Continuing with equation (1.9.3) we obtain

$$\begin{aligned} f(y) - f(x) &= f'(u)(y - x) = ((f'(u) - f'(x))(y - x)) + f'(x)(y - x) \\ &\geq f'(x)(y - x), \end{aligned}$$

which implies that f is convex from Theorem 1.9.1. ■

Theorem 1.9.3. *Assume that $f \in C^2$ on an interval $[a, b]$. The function f is convex on $[a, b]$ if and only if $f''(x)$ is nonnegative for all x in $[a, b]$.*

Proof. Assume that f is convex. The Taylor-series expansion of f at x gives

$$f(y) = f(x) + f'(x)(y - x) + \frac{1}{2}f''(\xi)(y - x)^2,$$

where $\xi \in \text{int}(x, y)$. Rearranging this equation gives

$$f(y) - f(x) - f'(x)(y - x) = \frac{1}{2}f''(\xi)(y - x)^2.$$

If f is convex, the left-hand side of this equation is nonnegative for all $x, y \in I$ from Theorem 1.9.1.

Conversely, the definition of the second derivative of f gives

$$\begin{aligned} f''(x) &= \lim_{h \rightarrow 0} \frac{1}{h} (f'(x + h) - f'(x)) \\ &= \lim_{h \rightarrow 0} \frac{1}{h^2} ((f'(x + h) - f'(x))h) \\ &\geq 0, \end{aligned}$$

for all h . ■

Chapter 2

Zero Finding

2.1. Introduction

Suppose that it is necessary to find a value x for which the nonlinear equation $e^x = 2$ is satisfied. This problem is equivalent to that of finding a value x for which the nonlinear function $F(x) = e^x - 2$ is zero. For this reason, finding a *root* or *solution* of a nonlinear equation is always equivalent to finding a *zero* of a related nonlinear function.

It is essential to recognize from the outset that “numerical” techniques for zero-finding differ in flavor from what might be termed “exact” or “analytic” techniques. To illustrate the latter, consider finding the solution of $e^x = 2$. Using elementary calculus, the exact (unique) solution is a “one-liner”: $x = \ln 2$. In this setting, a solution is found by exploiting knowledge of the *specific mathematical form* of the problem. Although the solution is a number, the number is typically represented in *abstract form* as a formal mathematical expression.

In contrast, the methods to be discussed in this chapter are designed to find a zero of a *generic* nonlinear function F for which the functional form is *unknown*, so that F is treated as a “black box” provided by the problem-solver that simply produces the value of F for any given x . (In some applications, values of the function $F(x)$ are available only as the result of a long calculation for which an explicit formula is not available or inconvenient.) Some knowledge is usually assumed of general properties of F (such as differentiability), but nothing more. Although a *computed solution* is a number (as in the exact case), it is expressed here in *concrete form* as a representable version of the solution. For the example above, the computed solution of $e^x - 2 = 0$ on a machine with six decimal digits is $x = .693147$.

Given a general function $F(x)$, the *zero-finding* problem is:

$$\text{find } x^* \text{ such that } F(x^*) = 0.$$

A point x^* where $F(x^*) = 0$ will be called either a *zero* or a *root* of F . Zero-finding algorithms usually involve *infinite* sequences of iterates $\{x_k\}$ with the property that x_k converges to x^* , but $x_k \neq x^*$ for any k . Consequently, although x_k becomes a better and better approximation to x^* , we may never actually “find” x^* itself!

2.2. Functions with one variable

Problems with one variable occur in many settings. Consider the everyday situation in which a person borrows money and wishes to make equal monthly payments so that the entire loan plus interest will be fully paid at the end of a specified time. The “right” monthly payment is the solution of a nonlinear equation in one variable (see below).

One-dimensional problems also occur as (sometimes unnoticed) *subproblems* within physical and numerical calculations involving higher dimensions. For example, high-energy physicists may need to determine when a typical particle reaches a critical velocity (a one-dimensional nonlinear equation). Finally, the study of problems involving just one variable suggests approaches that can often be generalized to solve problems with many variables. Concentrating first on the one-variable case helps make it easier to understand the underlying ideas and to “picture” the various strategies that are available.

In some applications, values of the function $F(x)$ are available only as the result of a long calculation for which an explicit formula is not available or inconvenient. The loan-repayment problem mentioned above is an example. Consider finding the monthly repayment x for an N -year loan of P dollars when the interest rate is $I\%$, compounded annually. The amount P_1 owing at the end of the first year is the balance of the loan at the beginning of the year, plus the yearly interest on the outstanding balance, minus the twelve monthly payments. We may write P_1 in terms of P and the unknown payment x as

$$P_1 = P + \frac{PI}{100} - 12x = P \left(1 + \frac{I}{100} \right) - 12x.$$

Similarly, if P_i is the amount owed at the end of year i , then

$$\begin{aligned} P_i &= P_{i-1} \left(1 + \frac{I}{100} \right) - 12x \\ &= \left(P_{i-2} \left(1 + \frac{I}{100} \right) - 12x \right) \left(1 + \frac{I}{100} \right) - 12x, \quad \text{etc.} \end{aligned}$$

If the monthly repayment is to pay off the loan at the end of the N th year, we must have

$$P_N = P_N(x) = P_{N-1} \left(1 + \frac{I}{100} \right) - 12x = 0.$$

Note that the evaluation of the function $f \equiv P_N$ can be accomplished using the recurrence relation

$$P_0 = P, \quad P_i = P_{i-1} \left(1 + \frac{I}{100} \right) - 12x, \quad i = 1, 2, \dots, N.$$

This particular form of the problem has an analytic solution, which can be found by summing the geometric series for P_N and solving explicitly for x . However, it is easy to devise other forms of the problem for which a solution cannot be written in closed form. For example, suppose that the lender offers a loan in which the interest rate is a function of the outstanding loan and interest. In this case, the interest at the end of the i th year is of the form $I_i \equiv I_i(P_i)$, and the problem generally must be solved numerically.

The numerical solution techniques under consideration are *iterative*, in the following sense. (For simplicity, only one-step methods are considered in this discussion.) If x^* is

the exact solution, a sequence $\{x_k\}$ is generated by a numerical method such that each successive iterate is a new (and, ideally, improved) estimate of x^* . An initial iterate x_0 is required, and production of x_{k+1} usually involves evaluation of F at x_k , and sometimes auxiliary calculations. Although in theory the sequence is infinite, in practice only a finite number of iterates are computed. It should not be surprising that we seek methods whose sequences of iterates are guaranteed to reach a “good” approximation to the desired solution.

Beyond assurance of eventual convergence, an important practical issue is the *efficiency* of a method—broadly speaking, the amount of work or time required to obtain a satisfactory solution. For an iterative method, it would be useful to know in advance how many iterations will be performed before termination. Unfortunately, disappointingly few general statements can be made about the total number of iterations for most of the “fast” iterative methods (in contrast to “interval reduction” methods; see §2.2.1). The most common kind of analysis considers instead *how fast* the *later* elements of the iterative sequence converge (see §1.5.1 concerning rates of convergence).

Because F may contain arbitrary nonlinearities, standard techniques for measuring work are unlikely to be meaningful. In linear algebraic applications, the work associated with certain factorizations depends in a known way on the matrix dimensions, and the number of “flops” required to compute the solution serves as an estimate of efficiency. However, the effort involved in computing an unknown nonlinear function cannot be estimated in advance. Even when F is “known”, the amount of work needed to compute it may depend on x —for example, when F involves solving a differential equation whose behavior changes radically for different parameter values. Since all zero-finding algorithms need to evaluate F , a frequent “after the fact” measure of efficiency is the number of evaluations of F needed to satisfy a specified set of termination criteria. Evaluation of other quantities (such as the first derivative F') may also be counted, although the proper weight to be assigned depends on the problem. (In some cases, F' is essentially a “free” byproduct of the calculation of F ; in others, computation of F' may increase the required time by several orders of magnitude.)

A complicated (and often overlooked) issue in designing an iterative method is the selection of *termination criteria*. An ideal termination criterion is one whose satisfaction ensures that the current iterate is a “sufficiently good” approximation to the desired solution. In the “pseudo-code” to be given for various algorithms, the phrase “test for convergence” indicates the invocation of an appropriate termination criteria. In a practical algorithm, the iterative sequence *must terminate* after a *finite* number of iterations. In the one-variable case, termination criteria generally involve tests of one or both of the following forms:

$$|F(x_k)| \leq \tau_F; \quad (2.2.1)$$

$$|x_k - x^*| \leq \tau_x, \quad (2.2.2)$$

where the “ F -tolerance” τ_F indicates that $|F|$ is “small enough”, and the “ x -tolerance” τ_x indicates that x_k is “close enough” to the desired solution.

Since x^* is unknown, condition (2.2.2) cannot be tested directly. Fortunately, in many instances an *interval of uncertainty* $[x_l, x_u]$ is known such that x^* lies somewhere in the interval, i.e., $x_l \leq x^* \leq x_u$. For example, suppose that F is continuous and we have an interval $[a, b]$ for which $F(a) \neq 0$, $F(b) \neq 0$ and $F(a)$ and $F(b)$ are of *opposite sign*. The

Intermediate Value Theorem (Theorem 1.6.1) immediately implies that there must exist at least one point in (a, b) where F vanishes, and $[a, b]$ constitutes an interval of uncertainty.

In this spirit, the best zero-finding methods compute not only a sequence of iterates $\{x_k\}$, but also a sequence of *intervals* (of uncertainty) $\{\mathcal{I}_k\}$, where $\mathcal{I}_k$ contains x_k as well as x^* . If each successive interval is strictly smaller in length, so that $|\mathcal{I}_{k+1}| < |\mathcal{I}_k|$ for all k , then $\{|\mathcal{I}_k|\}$ is a monotonically decreasing sequence. If $\mathcal{I}_{k+1}$ is constructed appropriately from $\mathcal{I}_k$, the length of the interval of uncertainty converges to zero. Consequently, condition (2.2.2) will eventually hold for some k , since satisfaction of (2.2.2) is implied by the *verifiable* condition

$$|\mathcal{I}_k| \leq \tau_x. \quad (2.2.3)$$

(For brevity, a statement that an interval of uncertainty “decreases” should be taken to mean that its *length* decreases.) Methods that produce a sequence of decreasing intervals of uncertainty are called *interval reduction methods*.

2.2.1. The method of bisection

The *method of bisection* is a classical technique for producing a sequence $\{\mathcal{I}_k\}$ of intervals of uncertainty with strictly decreasing lengths. The bisection algorithm starts with an initial interval of uncertainty $\mathcal{I}_0 = [a_0, b_0]$ such that F is of *opposite sign* at the endpoints. (As mentioned above, $\mathcal{I}_0$ is thus known to contain a zero of F .) Each “new” interval is constructed as follows. Let $\mathcal{I}_k$ denote the current interval $[a_k, b_k]$, which must have the property that $F(a_k)F(b_k) < 0$. The value of F is calculated at $m_k = \frac{1}{2}(a_k + b_k)$, the *midpoint* of $\mathcal{I}_k$. (In fact, only the *sign* of F at m_k needs to be determined.) If $F(m_k) = 0$, m_k is a zero and the algorithm terminates. If $F(m_k) \neq 0$, a new interval is produced by replacing either the left or right endpoint by m_k , thereby cutting the length of the interval in half. (It is customary to designate m_k as the new “iterate” x_{k+1} .) If $F(a_k)F(m_k) < 0$, the new interval $\mathcal{I}_{k+1}$ is $[a_k, m_k]$, and m_k replaces the right endpoint; otherwise, $\mathcal{I}_{k+1}$ is $[m_k, b_k]$, and the “old” left endpoint is discarded. Since exactly one of these conditions must hold, a bisection step is always well defined.

The pseudocode in Algorithm 2.2.1 represents a typical step of the bisection method, where “ a ” and “ b ” refer to *current* endpoints of the interval of uncertainty. We assume that $F(a_0) \neq 0$, $F(b_0) \neq 0$ and $\text{sign}(F(a_0)) \neq \text{sign}(F(b_0))$. The notation $\leftarrow$ is used as in a programming language to mean that the quantity on the right is evaluated first and then replaces the variable on the left.

Algorithm 2.2.1. The bisection algorithm.

```

 $a \leftarrow a_0; \quad b \leftarrow b_0;$ 
while  $b - a > \tau_x$  do
     $m \leftarrow (a + b)/2;$  evaluate  $F(m);$ 
    if  $\text{sign}(F(m)) = \text{sign}(F(a))$  then
         $a \leftarrow m$  else  $b \leftarrow m;$ 
end while

```

Each step of bisection requires one evaluation of F , and reduces the length of the interval of uncertainty by a factor of precisely two (no more, no less). Bisection can be viewed as an “optimal” algorithm for the class of continuous functions that change sign in

$[a, b]$, in the sense that *in the worst case* it yields the smallest interval of uncertainty for a specified number of function evaluations. Evaluating F at any point besides the midpoint *might* reduce the interval by a factor of more than two, but is not guaranteed to do so. Hence it is always possible to construct a continuous function whose zero will be found more quickly by bisection than by any other algorithm.

If $F(m_k) \neq 0$ for any k , the convergence test (2.2.3) will be satisfied for the smallest integer K such that

$$b_K - a_K \leq \tau_x, \quad \text{i.e.,} \quad \frac{b_0 - a_0}{2^K} \leq \tau_x,$$

which occurs when

$$K \approx \log_2 \left(\frac{b_0 - a_0}{\tau_x} \right). \quad (2.2.4)$$

Unless an exact zero is encountered by chance, (2.2.4) shows that the “efficiency” of bisection—i.e., the number of function evaluations required to satisfy (2.2.3)—does not vary with F , but depends only on the size of the initial interval. The length of the interval of uncertainty converges to zero linearly, with asymptotic error constant $\frac{1}{2}$ (see Definition 1.5.7).

Despite its reliability and guaranteed convergence, the bisection method has certain deficiencies. An important practical caveat is that the “true” zero x^* will be contained in the final interval produced by the bisection method only if the *computed* values of F always have the *correct sign*, so that care must be taken to evaluate the sign of F accurately.

Ironically, the most serious “defect” of bisection method when applied to many practical problems is precisely the quality of invariance that provides ironclad reliability for the entire class of continuous functions. Since the speed of convergence is independent of the behavior of F , it cannot improve when F is “well behaved”. The unvarying choice of the midpoint for the next evaluation of F is likely to be extremely inefficient if F is smooth and the values of F at the endpoints have widely differing magnitudes. For example, if $F(a) = -10^{50}$ and $F(b) = 10^{-8}$, x^* seems “more likely” to be close to b than to a . (Of course, this judgment depends completely on an assumption of “good behavior” by F .)

Another disadvantage is that bisection is not guaranteed to find a zero x^* at which F does not change sign. It follows from the Nonzero Derivative Theorem (Theorem 1.6.5) that any such zero must be a multiple root. (See Definition 1.6.2.) For example, when $F(x) = (x - 1)^2$, F is never negative, but has a double root at $x^* = 1$.

Furthermore, bisection must be initiated with an interval $[a, b]$ of somewhat specialized nature (satisfying $F(a)F(b) < 0$). In some circumstances, an initial interval of uncertainty may be known that does not possess this property, which means that additional function evaluations must be expended to begin the bisection algorithm.

Although “pure” bisection is rarely used in sophisticated zero-finding methods, it remains the quintessential interval reduction method. The principle of recursively “splitting” a large problem into a small number of more tractable subproblems can be applied in many contexts whose relationship to zero-finding is not immediately obvious. Bisection can be interpreted as the simplest form of powerful “divide and conquer” paradigms in sorting, searching and combinatorial problems. For example, the well known technique of *binary search* locates a specified element in a sorted list by comparing the desired element with the “middle” element, thereby eliminating half the list with a single comparison. The simplicity and reliability of the bisection method will always provide an important standard of comparison for all other methods.

2.2.2. Newton's method for zero-finding

Despite the importance of the bisection method, one can argue convincingly that *Newton's method* is even more fundamental. Newton's method occurs in many flavors and disguises, explicitly and implicitly, throughout mathematics. At the most basic level, Newton's method illustrates an important technique in algorithm design:

**REPLACE A HARD PROBLEM
BY A SEQUENCE OF (RELATED) EASY PROBLEMS.**

In the context of minimization, the original problem itself—finding a minimizer of a general nonlinear function F —is the “hard” problem. A simpler related problem is constructed in two stages. Given a current iterate x_k , F is *modeled* or *approximated* by a function (denoted by $L_k(x)$) that depends on x_k and whose minimizer can be found “easily”. Second, the minimizer of $L_k(x)$ (denoted by x_L) is computed and taken as an estimate of a minimizer of F . A (highly simplified) generic model-based algorithm has the structure shown in Algorithm 2.2.2, in which the above process is repeated at each new iterate.

Algorithm 2.2.2. Generic model-based algorithm.

```

 $k \leftarrow 0$ ;
repeat
    test for convergence and exit if converged;
    construct  $L_k(x)$ ;   compute  $x_L$ ;
     $x_{k+1} \leftarrow x_L$ ;    $k \leftarrow k + 1$ ;
end repeat

```

The success of such an approach depends completely on the “appropriateness” of the chosen model function in the context of minimization. It makes sense to construct a model function L that represents F only if F has general properties that can be incorporated into L . For example, we often assume that F is *continuously differentiable*, and choose a “simple” model function L with this same property.

Let x_k be a point at which $F(x_k) \neq 0$ (otherwise $F(x_k) = 0$ and there is nothing to do). We seek a model function L_k that approximates F and is likely to have a zero closer to x^* than x_k . An obvious candidate for L_k is a function whose *value* and *first derivative* at x_k match those of F at x_k , i.e., $L_k(x_k) = F(x_k)$ and $L'_k(x_k) = F'(x_k)$. These conditions are satisfied by an infinite number of smooth model functions. Among all these, however, the most straightforward and obvious must surely be the *tangent line* to F at x_k —and this choice is the foundation of Newton's method.

Once x_k has been specified, the corresponding values of F and F' are known. The tangent line at x_k is the *linear function*

$$L_k(x) = F(x_k) + F'(x_k)(x - x_k). \quad (2.2.5)$$

If $F'(x_k) \neq 0$, the unique zero x_L of this straight line satisfies

$$F(x_k) + F'(x_k)(x_L - x_k) = 0, \quad (2.2.6)$$

which gives the simple formula $x_L = x_k - F(x_k)/F'(x_k)$.

As in the generic algorithm (Algorithm 2.2.2), x_L is interpreted as a new estimate of x^* . If $F(x_L) \neq 0$, a new model may be derived based on the tangent line at x_L , and the process repeated. In this form, a Newton method for zero-finding begins with an initial point x_0 , and generates a sequence of Newton iterates $\{x_k\}$, where x_{k+1} satisfies

$$x_{k+1} = x_k - \frac{F(x_k)}{F'(x_k)}, \quad (2.2.7)$$

and corresponds to a zero of the model function $L_k(x)$. In the language of Algorithm 2.2.2 we have

Algorithm 2.2.3. Newton’s method for zero-finding.

Define $k_{\max}$;

Choose x_0 ;

$k \leftarrow 0$;

while not converged and $k \leq k_{\max}$ do

 Evaluate $F(x_k)$ and $F'(x_k)$;

$x_{k+1} \leftarrow x_k - F(x_k)/F'(x_k)$;

$k \leftarrow k + 1$;

end while

Another popular way to derive Newton’s method is to expand F in a Taylor series about x_k :

$$F(x_k + h) = F(x_k) + hF'(x_k) + h \int_0^1 [F'(x_k + \xi h) - F'(x_k)] d\xi. \quad (2.2.8)$$

(Note that h may be positive or negative.) Assuming that the integral remainder term is “negligible”, the local variation of F is modeled by a *linear function*:

$$F(x_k + h) \approx F(x_k) + hF'(x_k).$$

Letting $x = x_k + h$, the expression on the right is identical to $L_k(x)$ of (2.2.5).

It is not only convenient but also numerically desirable to compute the sequence (2.2.7) in terms of the *step* from one iterate to the next. In this case the Newton iteration is defined by

$$x_{k+1} = x_k + p_k, \quad \text{where } p_k \text{ satisfies } p_k = -\frac{F(x_k)}{F'(x_k)}. \quad (2.2.9)$$

Newton’s method is a fascinating and ubiquitous process that arises in many contexts, with a confusing variety of names. Suppose, for example, that one needs to compute $\sqrt{a}$ for a specific value of $a > 0$, but has neither calculator nor mathematical tables at hand. The well known “divide and average” technique can be used to produce a sequence $\{x_k\}$ of increasingly accurate estimates of $\sqrt{a}$, as follows. Starting with any x_0 , a new estimate x_{k+1} is obtained for $k \geq 0$ by averaging x_k and a/x_k :

$$x_{k+1} = \frac{1}{2} \left(x_k + \frac{a}{x_k} \right). \quad (2.2.10)$$

Examination of (2.2.10) reveals that it is precisely the Newton iteration (2.2.9) corresponding to the function $F(x) = x^2 - a$, since the Newton step p_k is

$$p_k = -\frac{F(x_k)}{F'(x_k)} = -\frac{(x_k^2 - a)}{2x_k} = -\frac{1}{2}x_k + \frac{1}{2}\left(\frac{a}{x_k}\right),$$

and $x_k + p_k$ is the expression (2.2.10).

To illustrate the results, we apply this process to calculate $\sqrt{9}$, i.e., to solve $x^2 - 9 = 0$. Starting with $x_0 = 10$, the values of $x_k - 3$ and of $F(x_k) = x_k^2 - 9$ in Table 2.1 are produced by (2.2.10) for $k = 0, \dots, 5$. (The computations were performed with approximately 16 decimal digits of precision.)

Iteration	$x_k - 3$	$x_k^2 - 9$
0	7	91
1	2.45	20.7025
2	5.5069×10^{-1}	3.60739
3	4.2704×10^{-2}	2.5805×10^{-1}
4	2.9967×10^{-4}	1.7981×10^{-3}
5	1.4966×10^{-8}	8.9795×10^{-8}

Table 2.1: Finding $\sqrt{9}$ with Newton's method.

The values of $|F(x_k)|$ and $|x_k - 3|$ in this example are not simply converging to zero, but are converging *very fast*. In fact, since the last few errors in x_k are effectively *squared* at each iteration, we speculate (correctly) that the iterates are converging *quadratically* to the exact value, i.e., $|x_{k+1} - 3|/|x_k - 3|^2$ is approaching a constant value. Probably the best known feature of Newton's method is this remarkable quadratic convergence near the solution. This behavior will be considered again in the next section.

A vast body of convergence theory has been developed for Newton's method (further details are given in the Notes and Selected Bibliography). We shall delay the proof until we consider Newton's method for the zero of a function of many variables.

In practical terms, the onset of quadratic convergence leads to an enormous reduction in error compared to (say) the linear improvement of bisection. For example, the interval of uncertainty is effectively reduced by a factor of 10^6 during the last *two* Newton iterations shown in Table 2.1, whereas *twenty* bisection steps would be required to achieve a similar reduction.

A major difficulty with Newton's method is that its spectacular convergence rate applies only *locally*, i.e., when the iterates reach a "small enough" neighborhood of x^* . Outside this neighborhood, the linear model function L_k may be inaccurate and misleading. In particular, when x_0 is not close enough to x^* , the Newton iterates may diverge, or may make only negligible progress toward x^* . These difficulties can be overcome by using a *safeguarded* method in which the Newton iterate is accepted only when it appears to make "reasonable" progress. The topic of safeguarded methods will be considered in Section 2.2.4

Although Newton's method must be applied with some caution to a general function F , it has exceptionally strong convergence properties in certain special cases—for example,

when F is twice-continuously differentiable and $F''(x) > 0$ for all x . (The functions e^x and x^2 have these properties.) Suppose that we wish to apply Newton's method to find a zero x^* (whose existence is known) of such a function. Because F'' is positive, the first derivative F' is *strictly increasing*. It follows that F itself is *increasing* to the right of x^* if $F'(x^*) > 0$, and *decreasing* to the right of x^* if $F'(x^*) < 0$.

Furthermore, the linear model of F defined by the Taylor-series expansion (2.2.8) is a consistent *underestimate* of the true value of F . To see why, consider the expansion of $F(x_k + \delta)$ for a Newton iterate x_k and some perturbation δ :

$$F(x_k + \delta) = F(x_k) + \delta F'(x_k) + \frac{1}{2}\delta^2 F''(\xi),$$

where $\xi \in \text{int}(x_k, x_k + \delta)$. Since $F''(\xi)$ is positive by assumption, rearrangement gives

$$F(x_k + \delta) - F(x_k) - \delta F'(x_k) = \frac{1}{2}\delta^2 F''(\xi) > 0. \quad (2.2.11)$$

Assume that $x_k \neq x^*$. If δ is specialized to p_k (the Newton step at x_k), it follows from the relation $F(x_k) + p_k F'(x_k) = 0$ that $F(x_k + p_k) > 0$ (see (2.2.9)). Since the next Newton iterate x_{k+1} is equal to $x_k + p_k$, we see that $F(x_k)$ *must be positive* for any Newton iterate x_k with $k \geq 1$.

Now suppose that $F'(x^*) > 0$ and consider a Newton iterate x_k that lies to the *right* of x^* , i.e., $x_k > x^*$. What can we conclude about x_k ? First, we recall that $F'(x_k) > 0$ and $F(x_k) > 0$, which shows that the Newton step $p_k = -F(x_k)/F'(x_k)$ will be *negative*. Second, we take δ in (2.2.11) as $x^* - x_k$, with the following result:

$$F(x^*) - F(x_k) - \delta F'(x_k) = -F(x_k) - \delta F'(x_k) > 0.$$

Dividing by $F'(x_k)$ gives the inequality

$$-\delta = x_k - x^* > \frac{F(x_k)}{F'(x_k)} = |p_k|,$$

which shows that the magnitude of the (negative) Newton step is *strictly less* than the step from x_k to x^* . Combining these observations, we see that x_{k+1} will lie *strictly between* x^* and x_k .

This analysis shows that, if $F''(x) > 0$ for all x , $F'(x^*) > 0$ and $F'(x_0) > 0$, the Newton iterates satisfy

$$x^* < x_k < \cdots < x_1 \quad (2.2.12)$$

for $k \geq 2$. Hence Newton's method has *guaranteed convergence*, and the iterates converge in a highly special pattern—monotonically from the right, with monotonically decreasing values of F . An analogous examination of the case when $F'(x^*) < 0$ shows that, if $F'(x_0) < 0$, then $x_1 < x_2 < \cdots < x_k < x^*$, and the Newton iterates converge monotonically from the left.

A specific instance of these conditions occurs when finding $\sqrt{a}$ using Newton's method as described above. If x_0 is positive and larger than $\sqrt{a}$, all subsequent iterates converge monotonically to $\sqrt{a}$ from above. If x_0 is positive but smaller than $\sqrt{a}$, the *next* iterate x_1 will lie to the right of $\sqrt{a}$, and the convergence pattern described above will apply beginning at x_1 . (See Table 2.1.)

2.2.3. Secant and regula falsi methods

In solving $F(x) = 0$ with Newton's method, the first derivative of F is evaluated at every iterate. For simple or “textbook” problems, F' is usually easy to calculate. In practice, however, computation of F' is sometimes unacceptably expensive—for example, F may be defined by a subroutine containing many intermediate calculations. Accordingly, it is desirable for practical reasons to derive a method with “fast” (superlinear) convergence, but in which only function values are required to compute the next iterate.

To do so, we retain the basic motivation of Newton's method (modeling F by a straight line), but take the model function $L(x)$ as the straight line that passes through the values of F at the two most recent iterates (the *secant* line). With this strategy, the derivative of L_k at x_k is not taken as the exact derivative $F'(x_k)$, but rather as a finite-difference approximation using x_k and x_{k-1} :

$$L'_k(x_k) = \frac{F(x_k) - F(x_{k-1})}{x_k - x_{k-1}}. \quad (2.2.13)$$

When $L'_k(x_k)$ is given by (2.2.13) and is *nonzero*, the linear model function

$$L_k(x) = F(x_k) + L'_k(x_k)(x - x_k)$$

has a unique zero

$$x_{k+1} = x_k - F(x_k) \frac{x_k - x_{k-1}}{F(x_k) - F(x_{k-1})}. \quad (2.2.14)$$

The zero-finding method whose iterates are defined by (2.2.14) is called the *secant method*. The steps of a generic secant method are given in Algorithm 2.2.4.

Algorithm 2.2.4. A generic secant method for zero-finding.

```

Choose  $x_0, x_1$ ;
 $x \leftarrow x_1$ ;
Evaluate  $F(x_0)$ ;
 $k \leftarrow 1$ ;
while not converged do
    Evaluate  $F(x)$ ;
     $L' \leftarrow (F(x) - F(x_0))/(x - x_0)$ ;
     $x_0 \leftarrow x$ ;  $x \leftarrow x - F(x)/L'$ ;
     $k \leftarrow k + 1$ ;
end while
```

(A word of warning: the next iterate (2.2.14) should *not* be computed from the algebraically equivalent form

$$x_{k+1} = \frac{x_{k-1}F(x_k) - x_kF(x_{k-1})}{F(x_k) - F(x_{k-1})},$$

which suffers from extreme rounding error when the iterates become close.)

As with Newton's method, the “pure” secant method (2.2.14) produces a new estimate of the zero rather than a reduced interval of uncertainty. In contrast to Newton's method,

the values of F at two points are required to initiate the secant method. Only one new function evaluation is required at each iteration, so that the work per iteration (measured in terms of function evaluations) is comparable to that of the bisection method.

If $F'(x^*) \neq 0$, and x_0 and x_1 are sufficiently close to x^* , the secant method has a *superlinear* convergence rate $r = (1 + \sqrt{5})/2 \approx 1.6180$ (see Definition 1.5.8). Thus, the secant method can achieve very rapid convergence (although not quite as rapid as the convergence of Newton's method).

We now apply the secant method to solve $x^2 - 9 = 0$, i.e., to calculate $\sqrt{9}$. Starting with $x_0 = 10$ and $x_1 = 1$, the values of $x_k - 3$ and of $F(x_k) = x_k^2 - 9$ in Table 2.2 are produced by (2.2.14) for $k = 0, \dots, 8$. (The computations were performed with approximately 16 decimal digits of precision.)

Iteration	$x_k - 3$	$x_k^2 - 9$
0	7.0	91
1	-2.0	-8
2	-1.2727	-6.0165
3	0.93333	6.4711
4	-0.20985	-1.2151
5	-2.9131×10^{-1}	-0.17394
6	1.0611×10^{-3}	6.3678×10^{-3}
7	-5.1761×10^{-6}	-3.1056×10^{-5}
8	-9.1523×10^{-10}	-5.4914×10^{-9}

Table 2.2: Finding $\sqrt{9}$ with the secant method.

Comparison of the last few iterates here with the analogous Newton iterates in Table 2.1 shows the slightly less rapid convergence of the secant method.

Despite its favorable local convergence properties, the secant method is not “perfect”. If $F(x_k)$ and $F(x_{k-1})$ have opposite signs, F must have a root between x_k and x_{k-1} . In this case, the secant iterate x_{k+1} defined by (2.2.14) is an *interpolation* step, and also lies between x_k and x_{k-1} . However, if $F(x_k)$ and $F(x_{k-1})$ have the same sign, so that the secant step is an *extrapolation*, x_{k+1} will lie *outside* the interval defined by x_{k-1} and x_k . Since there is no *a priori* bound on the secant step, the iterates may diverge. It is also possible (in theory) for the secant method to cycle indefinitely among the same three points.

Because the danger of divergence is associated with extrapolation, it might seem plausible to modify the secant method so that the model function represents a straight line connecting two points at which F has opposite sign. The *regula falsi* (“false position”) method is a variation of the secant method based on ensuring that $F(x_k)F(x_{k-1}) < 0$. The method begins with x_0 and x_1 satisfying $F(x_0)F(x_1) < 0$. Assuming that $F(x_{k-1})F(x_k) < 0$ at iteration k , the values of x_{k+1} and f_{k+1} are calculated using (2.2.14). Whereas the iteration counter is simply incremented in the secant method, (in effect “discarding” x_{k-1}), the regula falsi method discards either x_{k-1} or x_k , depending on which function value agrees in sign with $F(x_{k+1})$.

Unfortunately, the regula falsi method can be extremely *inefficient*. Somewhat surprisingly, although the method was designed to correct a deficiency in reliability of the secant

method, it can also be “unreliable” in the sense that the improvement at each step may become arbitrarily small. In particular, its performance can be *much worse* than bisection (even on well-behaved functions); its rate of convergence is linear, and the asymptotic error constant can be arbitrarily close to 1. The explanation for this behavior is that requiring $F(x_k)$ and $F(x_{k-1})$ to be of opposite sign may mean that a “poor” point will not be discarded for many iterations. Hence, the approximation (2.2.13) to F' remains inaccurate and progress is negligible.

The difference between the secant and regula falsi methods may be illustrated by finding a root of $x^6 - 1$ starting with $x_0 = 0.9$ and $x_1 = 10$. The iterates for both methods are shown in Table 2.3. Although the first three iterates are (necessarily) identical, the secant method thereafter converges rather quickly to the desired zero, whereas the regula falsi method “creeps” along because it is not permitted to discard x_1 (a “bad” point).

Iteration	Secant		Regula Falsi	
	x_k	$F(x_k)$	x_k	$F(x_k)$
0	.9	−.46856	.9	−.46856
1	10	999999	10	999999
2	.900004	−.46854	.900004	−.46854
3	1.03225	.20979	.900008	−.46853
4	.991350	-5.0789×10^{-2}	.900013	−.46851
5	.999322	-4.0610×10^{-3}	.900017	−.46850
6	1.00001	8.8931×10^{-5}	.900021	−.46848
7	.99999997	-1.5084×10^{-7}	.900026	−.46847
8	1.000000	-5.5892×10^{-12}	.900030	−.46845

Table 2.3: Solving $x^6 - 1 = 0$ with the secant and regula falsi methods.

The purpose of mentioning the regula falsi method (which is obviously not to be recommended!) is to emphasize that seemingly reasonable algorithmic modifications, especially those designed to increase reliability, may fail to achieve the desired improvements, and (even worse) may destroy the properties they wish to retain.

2.2.4. Safeguarded zero-finding algorithms

The bisection algorithm is a special example of the general zero-finding technique of “interval reduction”, in which a sequence of intervals of strictly decreasing size is generated until criterion (2.2.2) is satisfied. At every step, the “next” interval is constructed using a function value at a “new” point u in the existing interval, where u can be obtained from bisection, Newton’s method, the secant method, or some other procedure.

The best methods available for zero-finding are called *safeguarded* procedures. Their design is based on combining an interval reduction method (such as bisection) with a rapidly convergent method (such as Newton’s method or the secant method). The intent is to produce an algorithm that will converge rapidly if F is well-behaved, but is not much less efficient than the guaranteed method in the worst case. (Note that regula falsi fails to satisfy this latter criterion!)

For illustration, consider a safeguarded secant method. Throughout the execution of

the algorithm, we maintain both an interval of uncertainty (as in an interval reduction method) *and* a record of the two “best” points (to achieve the most accurate secant approximation). At each iteration, three conceptually distinct points are available: b , the “best” point, is the point found thus far for which $|F(b)|$ is least, and also one endpoint of the interval of uncertainty; a is the other endpoint of the interval of uncertainty, so that $F(b)F(a) < 0$; and c is the immediately preceding value of b (so that in some cases $a = c$). The secant fit (2.2.14) is then performed with b and c (the two “best” points), yielding a trial point s . Before accepting s as the next point, a safeguarded procedure checks whether s is “reasonable”, and computes an alternative point if it is not.

The definition of “reasonable” includes several requirements. First, an interval reduction step is taken if the secant step does not lie between a and b , i.e., in the known interval of uncertainty. For example, consider applying a safeguarded secant method to find the zero of $x^6 - 729$ in $[0, 5]$, where the exact solution is $x^* = 3$. The first four “pure” secant iterates x_k , $k = 0, \dots, 3$ and the values of $F(x_k) = x_k^6 - 729$ are shown in Table 2.4.

Iteration	x_k	$F(x_k)$
0	0.0	-729
1	5.0	14896
2	0.23328	-728.99984
3	1.0552×10^6	1.3805×10^{36}

Table 2.4: Divergent secant iterates.

The change of sign in F between x_0 and x_1 indicates that the zero must lie in $[x_0, x_1]$, so that in this case $[0, 5]$ is known to be an interval of uncertainty. Because F is highly nonlinear, however, the linear model is inaccurate, and the fourth secant iterate x_3 lies *outside* the known interval of uncertainty. Therefore, F should not even be evaluated at x_3 , and the next trial point should be chosen in some other way. This form of “safeguard” protects against nonsensical iterates that might be generated by an inaccurate model.

The “opposite” situation can arise in which the secant iterates become “bogged down” (*inside* the interval of uncertainty) because of an inaccurate model. A safeguarded algorithm accordingly *forces* an interval reduction step to be taken if the interval of uncertainty has not been sufficiently reduced after a specified number of iterations. This type of strategy ensures that the worst-case number of function evaluations for the safeguarded method is not significantly more than that required by a pure interval reduction method.

To illustrate why such a protection might be necessary, consider applying the secant method to find a zero of $x^6 - 1$ in $[0.5, 10]$, as shown in Table 2.5. (The function is the same as in Table 2.3, but x_0 is farther from x^* .)

Observe that the interval of uncertainty is hardly reduced at all between iterations 3 and 9 (from $[0.500009, 5.7498]$ to $[0.500584, 5.7232]$). In contrast, six function evaluations in the bisection method would be guaranteed to reduce the interval of uncertainty by a factor of 64, and hence an interval reduction step would be mandated in a safeguarded method.

Finally, protections are included to prevent successive iterates from becoming “too close”. If the secant step (2.2.14) is computed with nearly equal values of x , rounding errors may cause the predicted point to be poorly defined. To prevent this occurrence, a

Iteration	x_k	$F(x_k)$
0	.5	-.984375
1	10	999999
2	.500009	-.984373
3	5.7498	3.6131×10^4
4	.500152	-.984346
5	.500295	-.984319
6	5.7383	3.5700×10^4
7	.500440	-.984292
8	.500584	-.984265
9	5.7232	3.5142×10^4

Table 2.5: Slowly convergent secant iterates.

“small” distance is imposed as a lower bound on the permitted step.

2.3. Zerofinding for Functions of Many Variables

Now we turn to the general n -dimensional zero-finding problem of finding a vector x^* at which the m nonlinear scalar-valued functions $\{F_i(x)\}$, $i = 1, 2, \dots, m$, are simultaneously zero. If $F(x)$ denotes the m -vector with components $F_i(x)$, then the problem is to find x^* such that $F(x^*) = 0$.

Before attempting to find a zero of F , it would obviously be helpful to know if a zero actually exists! Unfortunately, the issue of existence of a zero when $F(x)$ is allowed to contain arbitrary nonlinear functions is in general complex and difficult to analyze. To see why, let us review some crucial properties of *linear* equations that simplify matters. One can view solving the linear system $Ax = b$ as finding a zero of $F(x)$, where $F(x) = Ax - b$. Suppose that we wish to solve such a system. In the space of n real variables, where $n > 1$, a single linear equation has the form $a^T x = \beta$, and specifies that x must lie in a certain hyperplane. If $\text{rank}(A) = r > 0$, every solution of $Ax = b$ can be written in the form $x_0 + Zv$, where x_0 satisfies $Ax_0 = b$ and Z contains $n - r$ linearly independent columns that form a basis for the null space of A , i.e., $AZ = 0$. The Jacobian of $F(x) = Ax - b$ is simply the constant matrix A , whose i th row is the normal vector of the i th equation. If A has linearly independent rows, a solution of $Ax = b$ *must exist*. When A is nonsingular (i.e., is square and has full rank), the solution of $Ax = b$ is a unique point. Even when the rows of A are not known to be linearly independent, finite procedures are known that can check whether $Ax = b$ is compatible. If not, no solution can exist; if so, a solution can be computed in a straightforward manner.

Sadly, *not one* of these desirable properties applies to a general zero-finding problem! Given a single nonlinear function $F(x)$ in n real variables, there may be no points satisfying $F(x^*) = 0$, regardless of the value of n . For example, consider the two-variable function $F(x_1, x_2) = x_1^2 + x_2^2 + 3$. Another possibility is that there may be a unique zero, regardless of the number of variables. For example, for any $n > 0$, the origin is the only point satisfying $\sum_{i=1}^n x_i^4 = 0$. In dealing with nonlinear zero-finding, we must accept that it is not possible in general to determine the existence of a solution, the number of degrees of freedom, or a simple representation of points satisfying the equations.

Despite these general cautions, the set of zeros of m “well-behaved” nonlinear equations has the same structure as in the linear case, and may be viewed as characterized by $n - m$ degrees of freedom. For example, the equation $\sum_{i=1}^n x_i^2 = 1$ is satisfied only by points that lie on the “unit ball” in n -space, and can be parameterized in terms of $n - 1$ other variables. We shall normally assume that the number of functions is equal to the number of variables, so that in writing $F(x) = 0$, F contains n nonlinear functions as components.

If there are *many* variables it is generally impossible to define an interval of uncertainty for x^* . In this situation it is necessary to use a termination criterion of the form $\|F(x_k)\| \leq \tau_F$ (analogous to (2.2.1)) and replace (2.2.2)) by $\|x_{k+1} - x_k\| \leq \tau_x$.

2.4. Newton's Method

Newton's method for a zero of a vector-valued function is motivated as in the univariate case—to develop a “simple” local model L_k of F . Given a current iterate x_k (recall that x_k is an n -vector), a simple *local model* of the function $F(x)$ at x_k is developed whose zeros can be characterized “easily”. The set of points that simultaneously satisfy all the model equations is then taken as an estimate of the solution of the original nonlinear system.

Let x_k be a point where $F(x_k) \neq 0$. An obvious candidate for $L_k(x)$ is a function whose value and derivative at x_k match those of $F(x)$ at x_k , i.e., $L_k(x_k) = F(x_k)$ and $L'_k(x_k) = F'(x_k)$. These conditions are satisfied by an infinite number of smooth model functions. Among all these, however, the most straightforward and obvious is the linear local model $L_k(x) = a_k + A_k(x - x_k)$. Substitution of the two conditions in this linear model gives $a_k = F(x_k)$ and $A_k = J(x_k)$, where $J(x)$ denotes the $m \times n$ Jacobian matrix of F . The local linear model is then $L_k(x) = F(x_k) + J(x_k)(x - x_k)$. Note that each F_i is approximated by a *linear function* $(L_k(x))_i$ of the form

$$(L_k(x))_i = F_i(x_k) + F'_i(x_k)(x - x_k) = F_i(x_k) + \nabla F_i(x_k)^T(x - x_k), \quad (2.4.1)$$

which is the same as saying that each F_i is approximated by its tangent hyperplane at x_k .

We would expect the linear model to be quite accurate in any neighborhood of x_k in which F were approximately linear. The size of this neighborhood will depend on the smoothness of F —i.e., on the behavior of the derivatives of F . For example, L_k will be accurate in any neighborhood of x_k where the second derivatives of F_i are “small”, or, equivalently, where the first derivatives “do not vary very much”. These statements can be made more precise using a closed-form expression for the error in $L_k(x)$. From the Taylor-series expansion with the integral form of the remainder we have

$$\begin{aligned} F(x) - L_k(x) &= F(x) - F(x_k) - J(x_k)(x - x_k) \\ &= \int_0^1 (J(x_k + \xi(x - x_k)) - J(x_k))(x - x_k) d\xi. \end{aligned}$$

(see §1.7.3). Using standard norm inequalities we obtain

$$\|F(x) - L_k(x)\| \leq \|x - x_k\| \int_0^1 \|J(x_k + \xi(x - x_k)) - J(x_k)\| d\xi, \quad (2.4.2)$$

which implies that the error in the linear model depends not only on the distance of x from x_k , but also the total amount that $J(x)$ varies from $J(x_k)$ on the line segment $[x_k, x_k +$

$\xi(x - x_k)]$. This expression may be simplified nicely if $J(x)$ has a certain smoothness property known as Lipschitz continuity.

Definition 2.4.1. *The matrix-valued function J is said to be Lipschitz continuous at x if there exists an open convex set D and a scalar γ such that $x \in D$ and*

$$\|J(x) - J(y)\| \leq \gamma \|x - y\|, \quad \text{for all } y \in D. \quad (2.4.3)$$

Essentially, this definition implies that the elements of J do not vary too rapidly in a neighborhood of x . It should be emphasized that if the components of F were twice continuously differentiable then J would be automatically Lipschitz continuous, with Lipschitz constant bounded by the magnitude of the largest second derivative. Thus, although requiring Lipschitz continuity of J is stronger than merely requiring J to be continuous, it is weaker than requiring $J(x)$ to be differentiable.

With the assumption (2.4.3) of Lipschitz continuity we obtain the very concise bound on the magnitude of the error in the linear model:

$$\|F(x) - L_k(x)\| \leq \gamma \|x - x_k\|^2 \int_0^1 \xi d\xi = \frac{1}{2} \gamma \|x - x_k\|^2. \quad (2.4.4)$$

This bound implies that the error in the model function is bounded by the Lipschitz constant times the square of the distance from x to x_k .

2.4.1. The Newton iteration

The set of points for which each linear model function vanishes is given by $x_k + p_k$, where p_k is any solution of the $m \times n$ linear system

$$J(x_k)p_k = -F(x_k). \quad (2.4.5)$$

If $\text{rank}(J(x_k)) = m$, i.e., the Jacobian at x_k has full row rank, this system must have a solution. If $m = n$ and J_k is nonsingular, the solution is unique.

A “pure” Newton method for zero-finding begins with an initial point x_0 , and generates a sequence of Newton iterates $\{x_k\}$, where x_{k+1} is given by $x_k + p_k$, and p_k (the *Newton direction*) satisfies (2.4.5):

$$x_{k+1} = x_k + p_k, \quad \text{where } J(x_k)p_k = -F(x_k). \quad (2.4.6)$$

(Compare these definitions with those of the one-variable case (2.2.9)).

Newton’s method for the zero of a function F of many variables may be summarized in the following algorithm:

Algorithm 2.4.1. Newton’s method for a zero.

Choose x_0 ;

$k \leftarrow 0$;

while not converged do

 Evaluate $F(x_k)$ and $J(x_k)$;

 Find p_k such that $J(x_k)p_k = -F(x_k)$;

$x_{k+1} \leftarrow x_k + p_k$;

end while

Note that this algorithm is not well-defined when the system $J(x_k)p_k = -F(x_k)$ does not have a solution. When $J(x_k)$ is nonsingular, it is sometimes convenient in theoretical contexts to write p_k in the mathematically equivalent form

$$p_k = -J(x_k)^{-1}F(x_k). \quad (2.4.7)$$

2.4.2. Convergence properties

We illustrate the performance of Algorithm 2.4.1 by applying it to a problem whose function and Jacobian are given by

$$F(x) = \begin{pmatrix} x_1^2 - x_2 \\ -x_1 + x_2^2 \end{pmatrix} \quad \text{and} \quad J(x) = \begin{pmatrix} 2x_1 & -1 \\ -1 & 2x_2 \end{pmatrix}. \quad (2.4.8)$$

This problem has two zeros, one at $x^* = (0,0)^T$ and the other at $x^* = (1,1)^T$. Table 2.6 shows the result of using Algorithm 2.4.1 with starting point $x_0 = (2,2)$. The iterates converge to the zero $x^* = (1,1)^T$. Note that, as in the univariate case, the iterates converge quadratically.

k	$\ x_k - x^*\ $	$\ F(x_k)\ $
0	1.4	2.8
1	0.47	6.3×10^{-1}
2	9.4×10^{-2}	1.0×10^{-1}
3	5.6×10^{-3}	5.6×10^{-3}
4	2.2×10^{-5}	2.2×10^{-4}
5	3.3×10^{-10}	3.3×10^{-10}

Table 2.6: Example of convergence of Newton's method.

Unfortunately, even on this simple example, Newton's method may not converge from every starting point. If the run is repeated with initial point $x_0 = (2,0)$, the iterates *diverge*, with $\|F(x_k)\|$ taking the values 4.5, 1.6×10^1 , 2.6×10^2 , 6.6×10^4 , 4.3×10^9 , 1.8×10^{19} , 3.4×10^{38} , 1.2×10^{77} , ...

The next theorem shows that if we start sufficiently close to a zero x^* , then Newton's method converges. Moreover, if Newton's method converges, then it usually converges quadratically.

Theorem 2.4.1. (*Convergence of Newton's Method for vector-valued functions.*) Let $F : \mathcal{D} \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a continuously differentiable function on an open convex set $\mathcal{D}$, and assume that $F(x^*) = 0$ for some $x^* \in \mathcal{D}$ and that $J(x^*)$ is nonsingular. Then there exists an open neighborhood S containing x^* such that, for any x_0 in S , the Newton iterates (2.4.6) are well defined, remain in S and converge to x^* at a q -superlinear rate. Moreover, if

$$\|J(x) - J(x^*)\| \leq \sigma \|x - x^*\|, \quad x \in S, \quad (2.4.9)$$

for some constant scalar $\sigma > 0$, then the sequence converges q -quadratically to x^* .

Proof. Let θ be a fixed scalar in $(0, 1)$. Since $J(x)$ is continuous at x^* and $J(x^*)$ is nonsingular, there is an open neighborhood S containing x^* and a positive scalar μ such that

$$\|J(x)^{-1}\| \leq \mu \quad \text{and} \quad \|J(y) - J(x)\| \leq \frac{\theta}{\mu} \quad (2.4.10)$$

for every x and y in S . Suppose that $x_k \in S$. Some simple rearrangement of (2.4.6) and the application of the relation $F(x^*) = 0$ gives

$$\begin{aligned} x_{k+1} - x^* &= J(x_k)^{-1} \left(F(x^*) - (F(x_k) + J(x_k)(x^* - x_k)) \right) \\ &= J(x_k)^{-1} (F(x_k + h) - L_k(x_k + h)), \end{aligned}$$

where $h = x^* - x_k$ and $L_k(x_k + h)$ is the linear model evaluated at $x_k + h$. Taking norms and using the first inequality in (2.4.10) implies that

$$\|x_{k+1} - x^*\| \leq \mu \|F(x_k + h) - L_k(x_k + h)\|.$$

This expression directly relates the error in the new iterate with the error in the linear model. Using the bound (2.4.2) gives

$$\begin{aligned} \|x_{k+1} - x^*\| &\leq \mu \left\| \int_0^1 [J(x_k + \xi h) - J(x_k)] h d\xi \right\| \\ &\leq \mu \left\{ \max_{0 \leq \xi \leq 1} \|J(x_k + \xi h) - J(x_k)\| \right\} \|h\|. \end{aligned} \quad (2.4.11)$$

If x_k lies in S then so will $x_k + \xi h$, and the second condition of (2.4.10) gives

$$\|x_{k+1} - x^*\| \leq \theta \|x_k - x^*\|.$$

Since $\theta < 1$, this implies that if $x_0 \in S$, then $x_k \in S$ for $k = 1, 2, \dots$, and that $\{x_k\}$ converges to x^* .

Now we consider the question of the *rate* of convergence of x_k . Define

$$\beta_k = \mu \left\{ \max_{0 \leq \xi \leq 1} \|J(x_k + \xi(x^* - x_k)) - J(x_k)\| \right\}, \quad (2.4.12)$$

and assume that $x_0 \in S$, with μ defined as above. The continuity of J and the convergence of $\{x_k\}$ to x^* imply that β_k converges to zero. The inequality (2.4.11) can be written as

$$\|x_{k+1} - x^*\| \leq \beta_k \|x_k - x^*\|, \quad (2.4.13)$$

which establishes the q -superlinear convergence of $\{x_k\}$ to x^* .

In order to verify q -quadratic convergence, we will need the assumption (2.4.9). Let ξ^* denote the value of ξ for which the maximum in (2.4.12) is achieved, and let y^* denote $x_k + \xi^*(x^* - x_k)$. Then

$$\|J(y^*) - J(x_k)\| \leq \|J(y^*) - J(x^*)\| + \|J(x^*) - J(x_k)\|. \quad (2.4.14)$$

Applying (2.4.13) in (2.4.14) and (2.4.12) gives

$$\beta_k \leq 2\mu\sigma \|x_k - x^*\|.$$

Hence, $\{x_k\}$ satisfies the definition of quadratic convergence to x^* . ■

This theorem implies that Newton's method will converge if x_0 is sufficiently close to x^* . The set of initial points x_0 for which Newton's method converges is known as the *domain of convergence*. Theorem 2.4.1 tells us that the size of the domain of convergence depends on two (usually independent) properties of F : the degree of nonlinearity (characterized by the scalar σ); and the proximity of $J(x^*)$ to a singular matrix (characterized by $\mu^* = \|J(x^*)^{-1}\|$).

The factor μ^* also affects the *rate* of convergence in the sense that the ratio $\|x_{k+1} - x^*\|/\|x_k - x^*\|^2$ will decrease more rapidly for a problem with a smaller value of μ^* . For example, if $J(x^*)$ nonsingular but $\|J(x^*)^{-1}\|$ is large, the convergence rate will look linear rather than quadratic. To illustrate this, consider the function

$$F(x) = \begin{pmatrix} x_1^2 - x_2 \\ 2x_1 - x_2 - \alpha(x_2^2 - 1) - 1 \end{pmatrix}, \quad (2.4.15)$$

where α is a fixed parameter in $[0, 1]$. For any fixed value of α , F has a zero at $x^* = (1, 1)^T$, with Jacobian

$$J(x^*) = \begin{pmatrix} 2 & -1 \\ 2 & -1 - 2\alpha \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ 2 & -1 \end{pmatrix} + O(\alpha).$$

Clearly, the smaller the value of α , the closer $J(x^*)$ is to a singular matrix. Table 2.7 shows the result of using Algorithm 2.4.1 with starting point $x_0 = (2, 2)$ for the case $\alpha = 10^{-2}$. Observe that, despite this relatively large value of α , the rate of convergence is not recognizably quadratic since the number of correct figures in x_k does not double at each iterate close to x^* .

k	$\ x_k - x^*\ $	$\ F(x_k)\ $
0	1.4	2.2
1	1.1	2.6×10^{-1}
2	0.51	6.4×10^{-2}
3	0.24	1.6×10^{-2}
4	0.10	3.9×10^{-3}
5	3.6×10^{-2}	8.8×10^{-4}
6	8.3×10^{-3}	1.6×10^{-4}
7	7.0×10^{-4}	1.2×10^{-5}
8	5.7×10^{-6}	9.8×10^{-8}
9	3.9×10^{-10}	6.7×10^{-12}

Table 2.7: Example of “slow quadratic” convergence of Newton's method. This table shows the result of using Newton's method on the problem (2.4.15), starting at $x_0 = (2, 2)$ for $\alpha = 10^{-2}$. The rate of convergence is not recognizably quadratic since the number of correct figures in x_k does not double at each iterate close to x^* .

When x_0 is not close enough to x^* , the linear model function $L_k(x)$ may be inaccurate and misleading and the Newton iterates may diverge, or may make only negligible progress

toward x^* . In later sections we will show that these difficulties can be overcome by using a *safeguarded* method in which the Newton iterate is accepted only when it appears to make “reasonable” progress.

2.5. The Finite-Difference Newton Method

In our discussion of Newton’s method thus far, we have assumed that the Jacobian can be evaluated at any value of x . For many practical problems, this assumption is unrealistic—for example, F may be computed by performing a complex sequence of calculations, such as solving a set of partial differential equations or running a simulation. Therefore, we now consider ways of retaining as many of the beneficial properties of Newton’s method as possible without the cost of evaluating the Jacobian J .

One of the most successful of these methods involves approximating the partial derivatives of F by a finite-difference approximation. The differentiability of F implies the existence of a limit:

$$\lim_{\delta \rightarrow 0} \frac{1}{\delta} \{F(x + \delta e_j) - F(x) - J(x)e_j\delta\} = 0,$$

where e_j is the j th unit vector. This implies that if we fix δ as some small value h_j , known as the *finite-difference interval*, then the vector $y_j = (1/h_j) \{F(x + h_j e_j) - F(x)\}$ will be an approximation to the j th column of $J(x)$. Let h denote a vector with components h_j . The matrix $J(x, h)$ such that

$$J(x, h) = \begin{pmatrix} y_1 & y_2 & \cdots & y_n \end{pmatrix}, \quad \text{with} \quad y_j = \frac{1}{h_j} \{F(x + h_j e_j) - F(x)\}, \quad (2.5.1)$$

satisfies $J(x, h) \approx J(x)$ and is known as a *forward-difference Jacobian* at x . The calculation of $J(x, h)$ requires $n + 1$ evaluations of the vector-valued function $F(x)$: one at each of the points $x, x + h_1 e_1, x + h_2 e_2, \dots$, and $x + h_n e_n$.

By analogy with the Newton equation (2.4.5), a direction p_k associated with $J(x_k, h)$ can be defined by

$$J(x_k, h)p_k = -F(x_k).$$

A method in which $J(x_k, h)$ replaces $J(x_k)$ is called a *finite-difference Newton method*.

Not surprisingly, the effectiveness of this method depends critically on how accurately $J(x_k, h)$ approximates $J(x_k)$. If we assume that the Jacobian is Lipschitz continuous, we obtain the following convenient forms for the error in $J(x, h)$.

Theorem 2.5.1. *Let $\mathcal{D}$ denote an open convex subset of $\mathbb{R}^n$. Let $F : \mathcal{D} \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a continuously differentiable function whose derivative F' is Lipschitz continuous in $\mathcal{D}$ with Lipschitz constant γ . If $\|\cdot\|$ denotes any norm such that $\|e_j\| = 1$, then*

$$\|[J(x, h) - J(x)]e_j\| \leq \frac{1}{2}\gamma|h_j| \quad \text{and} \quad \|J(x, h) - J(x)\| \leq \frac{1}{2}\gamma\|h\|_\infty.$$

Proof. From the definition of $J(x, h)$ we have

$$\begin{aligned} \|[J(x, h) - J(x)]e_j\| &= \frac{1}{|h_j|} \|F(x + h_j e_j) - F(x) - J(x)h_j e_j\| \\ &= \frac{1}{|h_j|} \|F(x + h_j e_j) - L(x + h_j e_j)\|, \end{aligned}$$

where $L(x+d)$ denotes the linear model $L(x) = F(x) + J(x)d$. Equation (2.4.4) may now be used to give

$$\| [J(x, h) - J(x)]e_j \| \leq \frac{1}{|h_j|} \frac{1}{2} \gamma \|h_j e_j\|^2 = \frac{1}{2} \gamma |h_j|.$$

Similarly

$$\|J(x, h) - J(x)\|_1 = \max_{1 \leq j \leq n} \| [J(x, h) - J(x)]e_j \|_1 \leq \max_{1 \leq j \leq n} \frac{1}{2} \gamma |h_j| = \frac{1}{2} \gamma \|h\|_\infty,$$

which is what we wanted to show. ■

A major question in the calculation of $J(x_k, h)$ in (2.5.1) is the choice of finite-difference interval h . The next two results indicate that, at least in theory, h must go to zero as $x_k \rightarrow x^*$ if quadratic convergence is to be maintained. It is convenient to present these results in the context of a more general class of methods characterized by the linear model

$$L_k(x) = F(x_k) + A_k(x - x_k),$$

where A_k is a nonsingular approximation to $J(x_k)$. This model is used to define the following algorithm:

Algorithm 2.5.1. Approximate Newton method for a zero.

Choose x_0 ;

$x \leftarrow x_0$;

$k \leftarrow 0$;

while not converged do

Find p such that $A_k p = -F(x)$;

$x \leftarrow x + p$;

Redefine A ;

$k \leftarrow k + 1$;

end while

To analyze this method we first derive the error in the linear model at an arbitrary point $x_k + d$.

Lemma 2.5.1. *Let $\mathcal{D}$ denote an open convex subset of $\mathbb{R}^n$. Let $F : \mathcal{D} \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a continuously differentiable function whose derivative F' is Lipschitz continuous in $\mathcal{D}$ with Lipschitz constant γ . Let $\{A_k\}$ be a sequence of nonsingular matrices. Define the sequence $\{x_k\}$ such that $x_{k+1} = x_k + p_k$, where p_k satisfies $A_k p_k = -F(x_k)$. Then, for any $d \in \mathbb{R}^n$,*

$$\|F(x_k + d) - L_k(x_k + d)\| \leq \frac{1}{2} \gamma \|d\|^2 + \|[A_k - J(x_k)]d\|. \quad (2.5.2)$$

Proof. Let $L_k^N(x)$ denote the linear model $L_k^N(x) = F(x_k) + J(x_k)(x - x_k)$ associated with the Newton iteration. For any $d \in \mathbb{R}^n$,

$$\|F(x_k + d) - L_k(x_k + d)\| \leq \|F(x_k + d) - L_k^N(x_k + d)\| + \|L_k^N(x_k + d) - L_k(x_k + d)\|.$$

From (2.4.4), a bound on the error in the first term—the error in Newton model—is given by $\|F(x_k + d) - L_k^N(x_k + h)\| \leq \frac{1}{2}\gamma\|d\|^2$. Substituting for L_k^N and L_k , we obtain

$$\begin{aligned}\|F(x_k + d) - L_k(x_k + h)\| &\leq \frac{1}{2}\gamma\|d\|^2 + \|F(x_k) + J(x_k)d - (F(x_k) + A_k d)\| \\ &= \frac{1}{2}\gamma\|d\|^2 + \|[J(x_k) - A_k]d\|,\end{aligned}$$

as required. ■

Theorem 2.5.2. *Let F be a function satisfying the assumptions of Lemma 2.5.1. If ϵ_k and ϵ_{k+1} denote the errors $\epsilon_k = x_k - x^*$ and $\epsilon_{k+1} = x_{k+1} - x^*$, then*

$$\|\epsilon_{k+1}\| \leq \|A_k^{-1}\| \left\{ \frac{1}{2}\gamma\|\epsilon_k\| + \frac{\|[A_k - J(x_k)]\epsilon_k\|}{\|\epsilon_k\|} \right\} \|\epsilon_k\|. \quad (2.5.3)$$

Proof. Rearranging the equation $x_{k+1} = x_k - A_k^{-1}F(x_k)$ and using the identity $F(x^*) = 0$ gives

$$\begin{aligned}x_{k+1} - x^* &= A_k^{-1} \left(F(x^*) - (F(x_k) + A_k(x^* - x_k)) \right) \\ &= A_k^{-1} (F(x_k + d) - L_k(x_k + d)),\end{aligned}$$

where $d = x^* - x_k$ and $L_k(x_k + d)$ denotes the linear model evaluated at $x_k + d$. The result follows by taking norms and applying Lemma 2.5.1. ■

This implies that a sufficient condition for superlinear convergence is

$$\lim_{k \rightarrow \infty} [A_k - J(x_k)] \frac{\epsilon_k}{\|\epsilon_k\|} = 0.$$

and a sufficient condition for quadratic convergence is that there exist a positive scalar σ and an integer K such that for all $k \geq K$,

$$\|A_k - J(x_k)\| \leq \sigma\|x_k - x^*\|. \quad (2.5.4)$$

When $J(x_k, h_k)$ is given by (2.5.1), Theorem 2.5.1 implies that condition (2.5.4) holds for all k only if $h_k \rightarrow 0$ as $x_k \rightarrow x^*$. However, this result does *not* apply in practice. In finite precision arithmetic a “too-small” finite-difference interval is disastrous numerically, because of inaccuracies in the resulting $J(x_k, h_k)$ due to “cancellation” error. For small h we have $F(x_k + he_j) \approx F(x_k)$ and consequently, when y_j is computed from (2.5.1), the error in computing $F(x)$ is revealed (magnified) by the cancellation of similar terms.

To discuss the effects of cancellation error on the calculation of approximate derivatives, consider the univariate case in which we wish to approximate $F'(x)$ by the finite difference

$$\delta(F, h) = \frac{F(x + h) - F(x)}{h}. \quad (2.5.5)$$

So far, discussion of the error in a finite difference has been restricted to the *truncation error*, which is the error in $\delta(F, h)$ resulting from the neglected terms in the Taylor-series

expansion of $F(x + h)$ (see Theorem 2.5.1 for the result in the multivariate case). If the differential form of Taylor-series remainder is used, the truncation error can be written as

$$T(h) \equiv \delta(F, h) - F'(x) = \frac{1}{2}hF''(x + \xi h), \quad (2.5.6)$$

where ξ is a point in $(0, 1)$. The truncation error can be *bounded* by an expression that decreases with $|h|$, but its exact value depends on the unknown point $x + \xi h$.

Unfortunately, $T(h)$ is not the only error present when $\delta(F, h)$ is evaluated on a computer. In a “real” calculation of δ , the quantities to be subtracted in (2.5.5) are subject to errors that occur because computer arithmetic is generally not exact. To see this, consider any real number y represented by its infinite decimal expansion

$$y = \pm 10^e \times \left(\frac{d_1}{10^1} + \frac{d_2}{10^2} + \frac{d_3}{10^3} \cdots \right),$$

where the d_i are integers such that $1 < d_1 \leq 9$, $0 \leq d_i \leq 9$, $i > 1$, and e is a signed integer exponent. (Here we use a decimal expansion for simplicity, but other forms are possible, with binary being the most common on digital computers.) The decimal expansion may be written in shorthand form as

$$y = \pm 10^e \times .d_1 d_2 d_3 \cdots, \quad (2.5.7)$$

where the “.” denotes the decimal point and the d_i are known as the decimal digits.

On a computer, there are only a finite number of digits (say t) available to represent a number. In this case every number z represented on the machine is held in the t -digit form

$$z = \pm 10^e \times .d_1 d_2 d_3 \cdots d_{t-1} d_t,$$

with $1 < d_1 \leq 9$, $0 \leq d_i \leq 9$, $1 < i \leq t$. The set of all z of this form is known as the set of *representable numbers*, and we say that any representable number is held in *finite precision*. The number $m = .d_1 d_2 d_3 \cdots d_t$ is known as the *mantissa* and satisfies $\frac{1}{10} \leq m \leq 1$.

Any number y of the form (2.5.7) must be represented on a computer by discarding the least significant digits to give a representable number $fl(y)$ known as the *floating-point value* of y . More precisely, $fl(y)$ is defined as representable number that is nearest to y , and the process of choosing $fl(y)$ is known as *rounding*. Note that the nearest representable number to y may have a least significant digit $\bar{d}_t$ that differs from d_t by one unit. Hence, the floating-point value of the exact representation (2.5.7) is

$$fl(y) = \pm 10^e \times .d_1 d_2 d_3 \cdots d_{t-1} \bar{d}_t,$$

where the “rounded” value $\bar{d}_t$ differs from d_t by at most one. Note that regardless of the form of rounding, we have $|fl(y) - y| \leq \frac{1}{2}10^{-t}10^e$.

Unfortunately, the computed value of $F(x)$ may have more than only one incorrect digit in the mantissa. This is a consequence of the fact that the result of a single arithmetic operation between two representable numbers y and z is not usually representable. This implies that a small rounding error is made during every calculation. During the course of a sequence of calculations, these rounding errors can accumulate and cause the computed value of $F(x)$ to have several incorrect digits in its mantissa. This may be illustrated

schematically as follows: let $\hat{F}(x)$ denote the computed value of $F(x)$ and write $\hat{F}(x)$ in the form

$$\hat{F}(x) = \pm 10^e \times .d_1 d_2 \cdots d_p \bar{d}_{p+1} \cdots \bar{d}_t,$$

where it is assumed that the barred digits $\bar{d}_{p+1}, \dots, \bar{d}_t$ are incorrect because of the cumulative effects of rounding error during the calculation of F .

Definition 2.5.1. *The value $\hat{F}(x)$ is said to have relative precision 10^{-p} if its mantissa has at least p correct decimal digits.*

We stress that this is a “rule of thumb” definition that requires the exponents of $\hat{F}(x)$ and $F(x)$ to be identical.

If F is small in magnitude (i.e., $|F| \ll 1$), then F will almost always have very low relative precision (i.e., p will be close to one). This happens because the leading zeros of the *unnormalized* form

$$\hat{F}(x) = \pm .00 \cdots 00 d_1 d_2 \cdots d_p \bar{d}_{p+1} \cdots \bar{d}_t$$

are “correct” digits of the mantissa before renormalization (i.e., before the exponent is adjusted). This happens, for example, when F is the difference of two numbers of similar size. Hence, $F(x)$ may still be accurate in an absolute sense even though the relative precision is poor. To allow for this case we add the following rule-of-thumb.

Definition 2.5.2. *The value $\hat{F}(x)$ is said to have absolute precision 10^{-p} if $\hat{F}(x)$ has at least p correct decimal digits, including the leading decimal places.*

A function F is said to have precision ϵ if it has relative precision ϵ when $F \geq 1$ and absolute precision ϵ when $F < 1$.

Now we return to the calculation of the difference $F(x+h) - F(x)$ in finite-precision arithmetic. Let $\hat{F}(x)$ and $\hat{F}(x+h)$ denote the computed values of $F(x)$ and $F(x+h)$. If h is small enough that both $\hat{F}(x+h)$ and $\hat{F}(x)$ have the same exponent and the same precision, then the relevant representations are

$$\begin{aligned} \hat{F}(x) &= 10^e \times .d_1 d_2 \cdots d_p \bar{d}_{p+1} \cdots \bar{d}_t \\ \hat{F}(x+h) &= 10^e \times .d'_1 d'_2 \cdots d'_p \bar{d}'_{p+1} \cdots \bar{d}'_t. \end{aligned}$$

(We have assumed for simplicity that F is positive.) Since h is small, F will be close to $F(x+h)$ and the values $F(x)$ and $F(x+h)$ will have a certain number of leading digits in common. Suppose, for arguments sake, that h is chosen so that the first three digits coincide, giving $d'_1 = d_1$, $d'_2 = d_2$ and $d'_3 = d_3$. In this case, the first three digits in the difference $\hat{F}(x+h) - \hat{F}(x)$ cancel and the mantissa will be of the form

$$.b_1 b_2 b_3 \cdots b_{p-3} \bar{b}_{p-2} \cdots \bar{b}_t,$$

so that we would expect the precision in $\delta(F, h)$ to be approximately $10^{-(p-3)}$, i.e., we would lose approximately three digits of precision in the approximate derivative. Clearly,

the smaller the value of h , the larger the number of identical leading digits of $\widehat{F}(x+h)$ and $\widehat{F}(x)$. In particular, if h is chosen small enough, *all* the correct digits will match, and

$$d'_1 = d_1, \quad d'_2 = d_2, \quad \dots \quad d'_p = d_p.$$

For such a value of h , there will be *no* correct figures in the difference, and $\delta(F, h)$ will have no precision.

These statements can be made more precise in the following discussion. First, it is necessary to define the absolute and relative errors in $F(x)$.

Definition 2.5.3. *The absolute error in $\widehat{F}(x)$ is*

$$\epsilon_A(x) = F(x) - \widehat{F}(x).$$

The relative error in $\widehat{F}(x)$ is

$$\epsilon_R(x) = \frac{F(x) - \widehat{F}(x)}{F(x)}.$$

If $F(x) \neq 0$ then $\epsilon_A(x) = \epsilon_R(x)F(x)$.

Observe that if $\widehat{F}(x)$ has precision ϵ , then $\epsilon \approx \epsilon_R(x)$ unless $|F|$ is small, in which case $\epsilon \approx \epsilon_A(x)$. When $\widehat{F}(x)$ has precision ϵ , a suitable model for the error in $\widehat{F}(x)$ is

$$|\widehat{F}(x) - F(x)| \sim \epsilon(1 + |F(x)|).$$

Now we will show that if $\widehat{F}(x)$ is computed with precision ϵ , then the error in $\delta(F, h)$ resulting from rounding errors will usually be of the order of ϵ/h .

From the definition of the absolute error we have

$$\widehat{F}(x) = F(x) + \epsilon_A(x) \quad \text{and} \quad \widehat{F}(x+h) = F(x+h) + \epsilon_A(x+h). \quad (2.5.8)$$

Let $\delta(\widehat{F}, h)$ denote the value of δ computed with the inexact function values (2.5.8). Then the difference between the computed and exact versions of δ satisfies

$$C(h) \equiv \delta(\widehat{F}, h) - \delta(F, h) = \frac{\epsilon_A(x+h) - \epsilon_A(x)}{h}. \quad (2.5.9)$$

This error, which is only present because of inaccuracies in the computed value of F , is the *cancellation error* defined informally earlier. If the magnitude of the absolute error in F is bounded by ϵ_A for all $x \in [x, x+h]$, then $C(h) \leq 2\epsilon_A/h$, so that broadly speaking, a *bound* on the cancellation error *increases* as $|h|$ decreases (the opposite effect from the truncation error (2.5.6)).

In the context of choosing the finite-difference interval for a finite-difference Newton method, h must be not only small enough to give satisfactory convergence near the solution, but also large enough to ensure that $J(x_k, h)$ retains adequate accuracy. Fortunately, a “sensible” choice of h appears to give good results in most practical problems. Returning to the univariate case, we note that the total error is the sum of the truncation error $T(h)$ (2.5.6) and the cancellation error (2.5.9), where changes in the size of $|h|$ tend to have (roughly) opposite effects on the sizes of these errors. The “ideal” finite-difference interval

would yield the smallest error in $\delta(F, h)$, but unfortunately is impossible (in general) to compute. We therefore settle for computing an interval that approximately minimizes a bound E on the total error

$$|T(h) + C(h)| \leq E = \frac{1}{2}h|F''(x + \xi h)| + \frac{|\epsilon_A(x + h) - \epsilon_A(x)|}{h}. \quad (2.5.10)$$

If the magnitude of the absolute error in F is bounded by ϵ_A for all $x \in [x, x + h]$, we have

$$|T(h) + C(h)| \leq \frac{1}{2}h|F''(x + \xi h)| + \frac{2\epsilon_A}{h}.$$

The “optimal” interval h^* that minimizes this bound is given by

$$h^* = 2\sqrt{\frac{\epsilon_A}{|F''(x + \xi h)|}}. \quad (2.5.11)$$

However, rather than attempting to estimate the “optimal” interval for each variable at every iterate of a finite-difference Newton method, it is typical to compute a set of intervals at the initial point x_0 , and to retain the same intervals as long as numerical difficulties are not encountered. The almost universal choice for this fixed interval is the square root of the precision to which the components of F can be calculated. To demonstrate that this is a plausible choice we need to reconsider the calculation of a difference interval for a univariate function F . Assume that F has precision ϵ . We also assume that the second derivative is not zero.

If $|F(x)|$ is small, then $\epsilon_A \approx \epsilon$ and as long as $|F''(x + \xi h)|$ is not close to zero, the expression (2.5.11) for the optimal interval implies that $h^* \sim \sqrt{\epsilon}$.

If $|F(x)|$ is not small, we have

$$\epsilon_A = |F| \epsilon_R \approx |F| \epsilon. \quad (2.5.12)$$

Suppose also that F and all its derivatives are of comparable magnitude in the region of interest, so that

$$|F| \sim |F''|. \quad (2.5.13)$$

(When this relation holds, we tend to regard F as “well scaled”.) Applying (2.5.12) and (2.5.13) in (2.5.11), it follows that

$$h^* \sim \sqrt{\epsilon} \quad (2.5.14)$$

as before.

Practical (i.e., *computable*) estimates of h^* must estimate the truncation error $\delta(F, h) - F'(x)$ more precisely using finite differences to estimate the second derivatives of F .

Finite-difference Newton methods typically mimic Newton’s method almost exactly even to the extent of appearing to have an asymptotic q-quadratic rate of convergence. For example, a finite-difference Newton method applied to the two-variable problem (2.4.8) gives results that are virtually indistinguishable from those of Table 2.6. The explanation for this phenomenon is that the linear rate is very fast (the linear convergence constant is proportional to h). Consequently, close to the solution, the quadratic term in the error is dominant except in a small neighborhood of the solution. On most problems the termination criteria are satisfied before the neighborhood exhibiting linear convergence is reached.

2.6. Newton's Method with Backtracking

We have already seen that the Newton iterates may fail to converge, or may diverge if the local model is inaccurate. However, when Newton's method *does* converge, the magnitude of each F_i and hence the norm of F eventually decrease monotonically. This behavior suggests that it might be possible to force convergence from *any* starting point by making sure that the *norm* of F decreases at each iteration. Suppose that instead of moving along the Newton direction from x_k to the point $x_k + p_k$, we move a fraction α_k of the step along p_k to the point $x_k + \alpha_k p_k$. The precise value of the fraction, or *step length* as it is more commonly called, is chosen to ensure that $\|F(x_k + \alpha_k p_k)\| < \|F(x_k)\|$. In effect, we are using the length of the vector F as a measure of progress of the algorithm.

These methods for zero-finding involve techniques associated with unconstrained minimization. The most common choice of norm to be minimized is the two-norm. Any minimizer of the two-norm of $F(x)$ is a minimizer of the squared two-norm of $F(x)$, and we accordingly consider how to minimize the "merit function"

$$\frac{1}{2}\|F(x)\|^2 = \frac{1}{2}F(x)^T F(x) = \frac{1}{2} \sum_{i=1}^n F_i(x)^2.$$

Other choices of norm are possible, but the two-norm is exceptionally convenient. In particular, we will show that if the Jacobian of $F(x_k)$ is nonsingular and $F(x_k) \neq 0$, it is always possible to take a positive step along the Newton direction $-J(x_k)^{-1}F(x_k)$ that decreases $\|F\|$ (and hence $\frac{1}{2}\|F\|^2$).

The standard technique is to try $\alpha_k = 1$ first, and if necessary, reduce α_k by a factor of $\frac{1}{2}$ until the required reduction in $\|F\|$ is achieved. This procedure is known as *backtracking*. If the backtracking is to be successful it is essential that there always exist points such that $\|F(x_k + \alpha p_k)\| < \|F(x_k)\|$. (If no such points exist, any algorithm for finding them is doomed to failure.)

Definition 2.6.1. (*Direction of decrease.*) A direction p is known as a direction of decrease for $\|F(x)\|$ at x if there exists a positive σ such that

$$\|F(x + \alpha p)\| < \|F(x)\|, \quad \text{for all } \alpha \in (0, \sigma). \quad \blacksquare$$

We define a norm $\|\cdot\|$ as being *smooth* if $\|x\|$ is a continuously differentiable function of x . Examples of smooth norms are the two norm and ellipsoidal norms $\|x\| = (x^T B x)^{1/2}$, with B symmetric positive definite. For any smooth norm, $\|F(x)\|$ is also continuously differentiable and a simple condition on the derivative will indicate whether or not p is a direction of decrease. Let $m(\alpha)$ define the univariate function

$$m(\alpha) = \|F(x_k + \alpha p_k)\|.$$

The function m is continuously differentiable as long as $\|\cdot\|$ is smooth, and the directional derivative of $\|F\|$ at x along p is defined as

$$m'(0) = \left. \frac{d}{d\alpha} m(\alpha) \right|_{\alpha=0} = \left. \frac{d}{d\alpha} \|F(x_k + \alpha p_k)\| \right|_{\alpha=0}.$$

If the directional derivative is negative for any smooth norm, then the Nonzero Derivative Theorem (see Theorem 1.6.5) implies that there must exist an interval of positive values

of α such that $\|F(x_k + \alpha p_k)\| < \|F(x_k)\|$. The next result shows that the directional derivative along the Newton direction at $\alpha = 0$ is always negative as long as $F \neq 0$, so that the Newton direction is indeed a direction of decrease for $\|F\|$.

Lemma 2.6.1. *Let $F : \mathcal{D} \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ be continuously differentiable. Let x be any point in $\mathcal{D}$ such that $\|F(x)\| > 0$, where $\|\cdot\|$ denotes an arbitrary norm on $\mathbb{R}^m$. If p is a direction such that $J(x)p = -F(x)$, then*

$$\lim_{\alpha \rightarrow 0^+} \frac{\|F(x + \alpha p)\| - \|F(x)\|}{\alpha} = -\|F(x)\|.$$

Proof. Since F is continuously differentiable, it must hold that

$$\|F(x + \alpha p)\| - \|F(x)\| = \|F(x) + \alpha J(x)p + r(\alpha)\| - \|F(x)\|, \quad (2.6.1)$$

where $r(\alpha)$ is a vector-valued function such that $\|r(\alpha)\| = O(\alpha^2)$. From the definition of p we have

$$F(x) + \alpha J(x)p = F(x) - \alpha F(x) = (1 - \alpha)F(x).$$

Using this identity in (2.6.1) we obtain

$$\|F(x + \alpha p)\| - \|F(x)\| = \|(1 - \alpha)F(x) + r(\alpha)\| - \|F(x)\|,$$

If we use the triangle inequality $\|x\| - \|y\| \leq \|x + y\| \leq \|x\| + \|y\|$, we obtain

$$-\alpha\|F(x)\| - \|r(\alpha)\| \leq \|F(x + \alpha p)\| - \|F(x)\| \leq -\alpha\|F(x)\| + \|r(\alpha)\|,$$

where we have assumed without loss of generality that $\alpha < 1$. The required result follows from the fact that $\lim_{\alpha \rightarrow 0^+} \|r(\alpha)\|/\alpha = 0$. ■

For the remainder of this section we will restrict our attention to the two-norm of F .

Lemma 2.6.2. *Let $F : \mathcal{D} \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ be continuously differentiable. Let x be any point in $\mathcal{D}$ such that $\|F(x)\| > 0$. Let p denote any nonzero direction. Then there exists an σ ($\sigma > 0$) such that*

$$\frac{d}{d\alpha} \|F(x + \alpha p)\| = \frac{1}{\|F(x + \alpha p)\|} p^T J(x + \alpha p)^T F(x + \alpha p), \quad \text{for all } \alpha \in [0, \sigma]. \quad (2.6.2)$$

Moreover, if p satisfies the equations $J(x)p = -F(x)$, then

$$\left. \frac{d}{d\alpha} \|F(x + \alpha p)\| \right|_{\alpha=0} = -\|F(x)\|,$$

and p is a direction of decrease for F at x .

Proof. Define $m(\alpha) = \|F(x + \alpha p)\|$. Equation (2.6.2) follows from the repeated application of the chain-rule to obtain $m'(\alpha)$.

If $J(x)$ is nonsingular, the Newton equations $J(x)p = -F(x)$ give

$$m'(\alpha)|_{\alpha=0} = \frac{1}{\|F(x)\|} p^T J(x)^T F(x) = -\frac{F(x)^T F(x)}{\|F(x)\|} = -\|F(x)\|.$$

The nonzero derivative theorem Theorem 1.6.5, implies that p is a direction of decrease.

If α_k can be chosen such that $\|F(x_{k+1})\| < \|F(x_k)\|$, the sequence $\{\|F(x_k)\|\}$ must converge because $\|F(x)\|$ is bounded below by zero (see Theorem 1.5.3). However, some care must be taken with the choice of step lengths if the $F(x_k)$ are to converge to zero. False convergence will occur if the step lengths $\{\alpha_k\}$ are inadvertently chosen so that $\|F\|$ is reduced by an ever-smaller amount at each iteration. For example, consider finding a zero of the function $F(x) = x$ with $\alpha_k = 1/2^{k+1}$, starting at the point $x_0 = 2$. Then $p_k = -x_k$ for all k and we generate the sequence of iterates 2, 1, $\frac{3}{4}$, $\frac{27}{32}$, which converges to $\bar{x} \approx .5776$. This example shows how a poor choice of α_k may cause $\|x_{k+1} - x_k\|$ to converge to zero even when $\|p_k\|$ is *not* converging to zero! (Recall from Theorem 2.4.1 that the Newton direction p_k is such that $p_k \rightarrow 0$ as $x_k \rightarrow x^*$.) To guard against this difficulty, we must choose α_k so that the sequence $\{\alpha_k\}$ can never converge to zero unless $\|p_k\|$ converges to zero.

A suitable definition of α may be derived from the properties of the model function $L_k(x)$ (see (2.4.1)). Recall that the choice of p_k is based on the observation that F is closely approximated by $L_k(x)$ in a sufficiently small neighborhood of x_k . If this is the case, the reduction in $\|F\|$ will be approximately equal to the reduction in $\|L_k\|$, i.e.,

$$\|F(x_k)\| - \|F(x_k + p_k)\| \approx \|L_k(x_k)\| - \|L_k(x_k + p_k)\|.$$

This implies that the backtracking procedure may be interpreted as follows. Starting with the guess of $\alpha_k = 1$, α_k is reduced by a constant factor until

$$\|F(x_k)\| - \|F(x_k + \alpha_k p_k)\| \approx \|L_k(x_k)\| - \|L_k(x_k + \alpha_k p_k)\|.$$

This implies that α_k is reduced until the change in $\|F\|$ is consistent with the change predicted by the linear model. To be more specific, we define a fixed parameter μ ($0 < \mu < 1$) and require that α_k gives a reduction in $\|F\|$ that is no worse than μ times the reduction in $\|L_k(x)\|$. This gives the condition

$$\|F(x_k)\| - \|F(x_k + \alpha p_k)\| \geq \mu(\|L_k(x_k)\| - \|L_k(x_k + \alpha p_k)\|). \quad (2.6.3)$$

The following result implies that we can always find an α_k sufficiently small that satisfies this inequality.

Lemma 2.6.3. *Let $F : \mathcal{D} \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ be continuously differentiable. At any point x_k in $\mathcal{D}$ such that $F(x_k) \neq 0$, define the affine model $L_k(x) = F(x_k) + F'(x_k)(x - x_k)$. If p is any vector such that $J(x_k)p \neq 0$, then*

$$\lim_{\alpha \rightarrow 0} \frac{\|F(x_k)\| - \|F(x_k + \alpha p)\|}{\|L_k(x_k)\| - \|L_k(x_k + \alpha p)\|} = 1.$$

Proof. Consider the univariate functions

$$\phi(\alpha) = \|F(x_k + \alpha p)\| \quad \text{and} \quad \psi(\alpha) = \|L_k(x_k + \alpha p)\|.$$

Then

$$\frac{\|F(x_k)\| - \|F(x_k + \alpha p)\|}{\|L_k(x_k)\| - \|L_k(x_k + \alpha p)\|} = \frac{\phi(\alpha) - \phi(0)}{\psi(\alpha) - \psi(0)} = \left[\frac{\phi(\alpha) - \phi(0)}{\alpha} \right] \left[\frac{\alpha}{\psi(\alpha) - \psi(0)} \right].$$

Taking the limit as $\alpha \rightarrow 0$ gives

$$\lim_{\alpha \rightarrow 0} \frac{\|F(x_k)\| - \|F(x_k + \alpha p)\|}{\|L_k(x_k)\| - \|L_k(x_k + \alpha p)\|} = \frac{\phi'(0)}{\psi'(0)}.$$

It remains to be shown that $\phi'(0)$ is equal to $\psi'(0)$. Lemma 2.6.2 gives

$$\phi'(0) = \frac{F(x_k)^T J(x_k)p}{\|F(x_k)\|} \quad \text{and} \quad \psi'(0) = \frac{L_k(x_k)^T L'(x_k)p}{\|L_k(x_k)\|}.$$

The result now follows from the fact that $L_k(x_k) = F(x_k)$ and $L'(x_k) = J(x_k)$. ■

The condition (2.6.3) can be simplified as follows. Using the definition of $L_k(x)$ and the definition of the Newton direction gives

$$\begin{aligned} \|L_k(x_k)\| - \|L_k(x_{k+1})\| &= \|F(x_k)\| - \|F(x_k) + F'(x_k)(x_k + \alpha_k p_k - x_k)\| \\ &= \alpha_k \|F(x_k)\|. \end{aligned}$$

Substituting for $\|L_k(x_k)\| - \|L_k(x_{k+1})\|$ in (2.6.3) leads to the following condition,

$$\|F(x_k + \alpha_k p_k)\| < (1 - \alpha_k \mu) \|F(x_k)\|. \quad (2.6.4)$$

This condition may be regarded as defining an α_k that enforces a *sufficient* decrease in $\|F(x)\|$.

It should be evident from (2.6.4) that provided the sequence $\{\alpha_k\}$ does not converge to zero, then $\|F(x_k)\|$ can only become arbitrarily close to $\|F(x_k + \alpha_k p_k)\|$ if $\|F(x_k)\|$ converges to zero (which is the objective of the whole process). An implementation of Newton's method with backtracking takes the following form:

Algorithm 2.6.1. Newton's method with backtracking.

Fix μ such that $0 < \mu < 1$;

Choose x_0 ;

$x \leftarrow x_0$;

$k \leftarrow 0$;

while not converged do

 Evaluate $J(x)$;

 Solve $J(x)p = -F(x)$;

$\alpha \leftarrow 1$;

while $\|F(x + \alpha p)\| > (1 - \alpha\mu)\|F(x)\|$ **do**

$\alpha \leftarrow \alpha/2$;

end while

$x \leftarrow x + \alpha p$;

end while

Note that each step length is of the form $\alpha_k = 1/2^{j_k-1}$, where j_k is an integer such that $j_k \geq 1$.

We illustrate the performance of Newton's method with backtracking, by reconsidering the problem of finding x_1 and x_2 such that $F(x_1) = x_1^2 - x_2 = 0$, $F_2 = -x_1 + x_2^2 = 0$. We have

$$F(x) = \begin{pmatrix} x_1^2 - x_2 \\ -x_1 + x_2^2 \end{pmatrix}, \quad \text{with} \quad J(x) = \begin{pmatrix} 2x_1 & -1 \\ -1 & 2x_2 \end{pmatrix}.$$

In Table 2.8 we give the iterates of Newton's method with x_0 given by the point $(2, 0)$, for which Algorithm 2.4.1 diverged. Note that in this case, the iterates converge quadratically to the zero $x^* = (0, 0)^T$.

k	n_F	α_k	$\ F(x_k)\ $
0	1	—	4.5
1	4	0.25	3.3
2	5	1.0	1.0
3	6	1.0	3.8×10^{-1}
4	7	1.0	1.1×10^{-1}
5	8	1.0	1.3×10^{-2}
6	9	1.0	1.6×10^{-4}
7	10	1.0	2.6×10^{-8}
8	11	1.0	6.6×10^{-16}

Table 2.8: The iterates of Newton's method with a backtracking line search.

2.7. Treatment of Singularity

Consider the solution of the linear system $Jp = -F$. If J is nonsingular, then $p = -J^{-1}F$ is the unique solution of $Jp = -F$.

If J is singular with rank r ($r < n$), then either there is an infinite number of solutions of $Jp = -F$, or no solution exists at all. If $r < n$, a solution exists only if F can be written as a linear combination of the columns of J , i.e., if $F \in \text{range}(J)$, where $\text{range}(J)$ denotes the *range space* or *column space* of J —the set of vectors $\{y \mid y = Jp \text{ for some } p\}$. In this case, the Newton system is said to be *compatible* or *consistent*. If $r < n$ and the Newton equations are compatible, then there are an infinite number of solutions of $Jp = -F$. This may be shown by considering the *null-space* of J , which is the $(n - r)$ -dimensional subspace of $\mathbb{R}^n$ composed of vectors z such that

$$\text{null}(J) = \{z \mid Jz = 0\}.$$

It follows that if p is a solution of $Jp = -F$, and z is a vector in $\text{null}(J)$, then $\bar{p} = p + z$ is also a solution of $Jp = -F$ since $J\bar{p} = J(p + z) = Jp + Jz = -F$.

If the Newton system is *not* compatible, we choose p so that Jp is as close as possible to the right-hand side $-F$. We define “close as possible” to mean that p must give the smallest possible distance d between Jp and $-F$, i.e., p minimizes $\|d\| = \|Jp + F\|$ for some vector norm $\|\cdot\|$. Although a solution of this problem exists, whatever the choice of norm, it is not unique. To show this we consider any p^* that minimizes $\|Jp + F\|$, and add any vector z in $\text{null}(J)$. It follows that $\|Jp + F\| = \|J(p + z) + F\|$ and $\bar{p} = p + z$ gives

the same optimal norm as p . Note that a minimizing value of p can be used even when the Newton equations are compatible, since $\|Jp + F\|$ is zero at any solution of $Jp = -F$.

The most common choice of norm is the two-norm, which leads to the *linear least-squares* problem:

$$\underset{p \in \mathbb{R}^n}{\text{minimize}} \quad \|Jp + F\|_2. \quad (2.7.1)$$

If p is not unique, it seems sensible to choose the *smallest* p that solves the least-squares problem. The *least-length* least-squares solution, denoted by p_+ , is a solution of (2.7.1) that has the least value of $\|p\|_2$. The least-length solution has the nice properties that it is unique and can be defined (implicitly) using a matrix analogous to the inverse of J .

Theorem 2.7.1. *The least-length least-squares solution p_+ is unique, and there exists a unique $n \times n$ matrix J^+ such that $p_+ = -J^+F$. The matrix J^+ is known as the pseudoinverse of J . ■*

If J is nonsingular, the pseudoinverse is just J^{-1} , as we would expect.

The pseudoinverse of J is easily defined in terms of its *singular-value decomposition*. When J is $n \times n$, it can be written as

$$J = USV^T, \quad (2.7.2)$$

where U and V are $n \times n$ orthogonal matrices, and S is an $n \times n$ diagonal matrix, $S = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$, with $\sigma_i \geq 0$, $i = 1, \dots, n$. The nonnegative numbers $\{\sigma_i\}$ are called the *singular values* of J , and (2.7.2) is called the *singular-value decomposition* (SVD). The singular-value decomposition is arguably the most informative general representation of a matrix. The convention is usually adopted that $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$, so that $\sigma_1(J)$ denotes the *largest* singular value of J .

If J has rank r and $r > 0$, then J has exactly r strictly positive singular values, so that $\sigma_r > 0$ and $\sigma_{r+1} = \dots = \sigma_n = 0$. If J is nonsingular, all its singular values are nonzero. The singular values of J are the square roots of the eigenvalues of $J^T J$ (which are identical to the square roots of the eigenvalues of JJ^T).

Theorem 2.7.2. *If U , S and V denote the matrices associated with the SVD (2.7.2) of J , then the pseudoinverse of J is given by*

$$J^+ = V \begin{pmatrix} S_r^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^T.$$

where S_r is the diagonal matrix of positive singular values $S_r = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$. ■

The least-length solution is now easily found. Suppose that the columns of U and V be partitioned so that

$$U = \begin{pmatrix} U_r & U_{n-r} \end{pmatrix} \quad \text{and} \quad V = \begin{pmatrix} V_r & V_{n-r} \end{pmatrix}. \quad (2.7.3)$$

where U_r and V_r contain the first r columns of U and V . It follows from Theorem 2.7.1 that

$$p_+ = - \begin{pmatrix} V_r & V_{n-r} \end{pmatrix} \begin{pmatrix} S_r^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} U_r^T \\ U_{n-r}^T \end{pmatrix} F = -V_r S_r^{-1} U_r^T F.$$

If J is nonsingular, $J^+ = VS^{-1}U^T = J^{-1}$. In this case, $p = -VS^{-1}U^TF$.

The next result shows that under circumstances that hold most of the time, a modified Newton direction is a direction of decrease for the two-norm of F . This result uses the property of the SVD that certain columns of U and V can be used to generate the range- and null-spaces of both J and J^T . In particular, every $z \in \text{null}(J)$ can be written as $z = V_{n-r}w$, for some $(n-r)$ -vector w (which implies that the columns of V_{n-r} form an orthonormal basis for $\text{null}(J)$). This can be verified using the partition (2.7.3) in the definition of

$$\begin{aligned} Jz &= U \begin{pmatrix} S_r & 0 \\ 0 & 0 \end{pmatrix} V^T V_{n-r}w = U \begin{pmatrix} S_r & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_r^T \\ V_{n-r}^T \end{pmatrix} V_{n-r}w \\ &= U \begin{pmatrix} S_r & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ w \end{pmatrix} = 0. \end{aligned}$$

Similarly, any $z \in \text{null}(J^T)$ can be written as $z = U_{n-r}y$ for some y , and the columns of U_{n-r} form an orthonormal basis for $\text{null}(J^T)$.

Lemma 2.7.1. *Let $F : \mathcal{D} \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ be continuously differentiable. Let x be any point in $\mathcal{D}$ such that $F(x) \notin \text{null}(J(x)^T)$. Then the direction $p_+ = -J(x)^+F(x)$ is a direction of decrease for $\|F(x)\|_2$ at x .*

Proof. Define $m(\alpha) = \|F(x + \alpha p)\|_2$. As in the proof of Lemma 2.6.2, we have

$$m'(\alpha)|_{\alpha=0} = \frac{1}{\|F(x)\|_2} p^T J(x)^T F(x).$$

Substituting for the modified Newton direction $p = -J(x)^+F(x)$ gives

$$m'(\alpha)|_{\alpha=0} = \frac{1}{\|F(x)\|_2} p^T J(x)^T F(x) = -\frac{F(x)^T (J(x)J(x)^+)^T F(x)}{\|F(x)\|_2}.$$

Using the singular value decomposition of J , we have

$$JJ^+ = U \begin{pmatrix} S_r & 0 \\ 0 & 0 \end{pmatrix} V^T V \begin{pmatrix} S_r^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^T = U_r U_r^T.$$

It follows that

$$m'(\alpha)|_{\alpha=0} = -\frac{F(x)^T U_r U_r^T F(x)}{\|F(x)\|_2} = -\frac{\|U_r^T F(x)\|_2^2}{\|F(x)\|_2}.$$

We assert that $U_r^T F(x) \neq 0$. Otherwise, if $U_r^T F(x) = 0$, then $F(x) = U_{n-r}w$ for some nonzero vector w , and it follows that $F(x) \in \text{null}(J(x)^T)$, contradicting the assumption of the lemma. It follows that

$$m'(\alpha)|_{\alpha=0} = -\frac{\|U_r^T F(x)\|_2^2}{\|F(x)\|_2} < 0,$$

and p must be a direction of decrease for $\|F\|_2$. ■

The condition that $F(x)$ must not lie in the null-space of J^T requires some explanation. Simple differentiation gives the gradient of $\phi(x) = \|F(x)\|_2$ as

$$\nabla\phi(x) = \frac{1}{\|F(x)\|} J(x)^T F(x).$$

If x is *minimizer* of $\phi(x)$ such that $\phi(x) > 0$, then $\nabla\phi(x) = 0$. This must imply that $J(x)^T F(x) = 0$ and $F(x)$ lies in the null-space of J^T at a local minimizer of $\|F(x)\|_2$. This is not surprising, because under suitable conditions, it is possible to show that satisfaction of a sufficient decrease condition (2.6.4) ensures that the iterates will converge to a *local minimizer* of $\|F(x)\|$. It follows that we are implicitly treating the problem of solving $F(x) = 0$ as the problem of minimizing $\|F(x)\|$. For any norm, $\|F(x)\|$ is zero only if $F(x)$ is zero, and hence any zero of $F(x)$ must be a minimizer of $\|F(x)\|$. The opposite implication does not follow, however, since the norm of $F(x)$ may be minimized at a point that is not a zero of F —for example, let $F(x) = x^2 + 3$.

It is important to note that in zero finding with an n -dimensional F , x_k can converge to an erroneous local minimizer only if $J(x)$ can (and does) become singular. Fortunately, there are many important applications in which x can be restricted to lie in a region in which J is nonsingular. In this case, convergence is assured.

In summary, we state a robust form of Newtons' method with a back-tracking line search.

Algorithm 2.7.1. Modified Newton with backtracking.

```

Fix  $\mu$  such that  $0 < \mu < 1$ ;
Choose  $x_0$ ;
 $x \leftarrow x_0$ ;
 $k \leftarrow 0$ ;
while not converged do
    Evaluate  $J(x)$ ;
     $p \leftarrow -J(x)^+ F(x)$ ;
     $\alpha \leftarrow 1$ ;
    while  $\|F(x + \alpha p)\| > (1 - \alpha\mu)\|F(x)\|$  do
         $\alpha \leftarrow \alpha/2$ ;
    end while
     $x \leftarrow x + \alpha p$ ;
     $k \leftarrow k + 1$ ;
end while

```

To illustrate the performance of this modified Newton method, consider finding a zero of the functions $F_1 = \arctan x_1 - x_2$ and $F_2 = \frac{1}{4}x_1^2 - x_2$ starting at the point $x_0 = (1, -10)^T$. At x_0 we have

$$F(x_0) = \begin{pmatrix} 10.785 \\ 10.250 \end{pmatrix}, \quad \text{and} \quad J(x_0) = \begin{pmatrix} \frac{1}{2} & -1 \\ \frac{1}{2} & -1 \end{pmatrix},$$

where the components of $F(x_0)$ are given to five figures. A “pure” Newton method is undefined at x_0 since $J(x_0)$ is singular, and there is no p such that $J(x_0)p = -F(x_0)$.

However,

$$p_0 = -J(x_0)^+ F(x_0) = \begin{pmatrix} -4.207 \\ 8.414 \end{pmatrix}$$

The components of $J(x_0)^T F(x_0)$ are $(10.52, -21.03)^T$, which implies that p_0 is a direction of decrease for $\|F(x)\|$ at x_0 . After this modification, Algorithm 2.7.1 requires six (unmodified) iterations to converge.

Chapter 3

Unconstrained Optimization

3.1. Introduction

This chapter is concerned with the determination of an *unconstrained minimizer* of a nonlinear function $f(x)$. Broadly speaking, an unconstrained minimizer x^* is a point at which $f(x)$ achieves its minimum value. In this context, $f(x)$ is known as the *objective function*.

Throughout this chapter we shall consider problems that are at least continuously differentiable, i.e., all first derivatives exist and are continuous. In some cases, we go further and assume that the second derivatives also exist and are continuous. A solution of the problem will always be denoted by x^* .

To understand the theory of unconstrained minimization, it is helpful to start with the simplest case: the minimization or maximization of a function of one variable. For simplicity, we address only the problem of *minimization*. This restriction causes no loss of generality: finding a maximum of f is equivalent to finding a minimum of $-f$. Given a function f defined on an open interval $\mathcal{D}$, and an interval $\mathbb{F} \subset \mathcal{D}$, the problem of interest can be expressed as:

$$\text{find } x^* \in \mathbb{F} \text{ such that } f(x^*) = \min\{f(x) \mid \text{for all } x \in \mathbb{F}\} \quad (3.1.1)$$

If a point x^* satisfying (3.1.1) exists, it is known as a *global minimizer* of f on the interval $\mathbb{F}$. In this case, $f(x^*)$ is known as the *global minimum* of f in $\mathbb{F}$. Note that this problem involves finding two quantities: a point x^* and a value $f(x^*)$. The term *minimizer* refers to the argument x^* , whereas the *minimum* is the associated function value $f(x^*)$.

If f is continuous and $\mathbb{F}$ is the closed bounded interval $\{x : a \leq x \leq b\}$, a famous theorem of analysis—the extreme value theorem—states that a continuous function defined on a closed bounded interval always attains its minimum and maximum value, so that this problem always has at least one solution. However, this solution may be “constrained” in the sense that x^* may lie on the boundary of $[a, b]$ (e.g., consider the function $f(x) = x$ on the interval $[0, 1]$). In this case the minimizer will change if the end-points of the interval $[a, b]$ are changed, which implies that the solution is determined by the interval $[a, b]$ as well as the function f . Constrained minimizers will be considered later. Here we consider so-called *unconstrained local minimizers*, which are minimizers characterized solely by the properties of f .

The concept of “minimization” is inherently *relational*, and thus implies that a solution can be defined only in terms of *its relationship to surrounding points*. Essentially, $f(x^*)$ is a local minimum if $f(x^*)$ is less than or equal to $f(x)$ for all “points around x^* ”. These points are conveniently characterized by points lying in the so-called “open ball” $\mathcal{B}(x^*, \delta) = \{x \mid \|x - x^*\| < \delta\}$. With this idea in mind, we give the following precise characterization of an unconstrained local minimizer of f .

Definition 3.1.1. (*Unconstrained local minimizer.*) Let f be a function defined on an open convex set $\mathcal{D}$ contained in $\mathbb{R}^n$. A point x^* is a unconstrained local minimizer of f if there exists an open ball $\mathcal{B}(x^*, \delta)$ contained in $\mathcal{D}$ such that

$$f(x^*) \leq f(x) \quad \text{for all } x \in \mathcal{B}(x^*, \delta). \quad (3.1.2)$$

A point x^* is a strict or proper minimizer if this relation holds with strict inequality except at x^* itself, i.e., $f(x^*) < f(x)$ for all $x \neq x^*$, $x \in \mathcal{B}(x^*, \delta)$.

Definition 3.1.2. (*Isolated unconstrained local minimizer.*) A unconstrained local minimizer x^* is isolated if there exists an open ball $\mathcal{B}(x^*, \delta)$ contained in $\mathcal{D}$ in which x^* is the unique unconstrained local minimizer. ■

It is worth noting that in each of these definitions, it is not necessary to use a “regular” region to define points neighboring to x^* . The comparison may be done over the interior of *any* compact (i.e., closed and bounded) region S in $\mathbb{R}^n$. (Boundary points of S must be excluded because boundary points at which f is decreasing could define extraneous “constrained” minimizers of f .) The following definition is a generalization of Definition 3.1.1.

Definition 3.1.3. Let f be a function defined on an open convex set $\mathcal{D}$. A point x^* is a unconstrained local minimizer of f if there is a compact set S contained in $\mathcal{D}$ such that

$$x^* \in \text{int}(S) \quad \text{and} \quad f(x^*) \leq f(x) \quad \text{for all } x \in S. \quad (3.1.3)$$

A more general form of the extreme value theorem may be used to show that the problem $\min\{f(x) \mid x \in S\}$ always has at least one solution. This implies that x^* is a local minimizer if it is an interior solution of $\min\{f(x) \mid x \in S\}$.

In the study of optimization algorithms it is often helpful to be able to visualize the properties of a function of n variables. A useful definition is that of a *level set*.

Definition 3.1.4. (*Level set and level curve.*) Let f be a function defined on an open convex set $\mathcal{D} \subset \mathbb{R}^n$. Given a scalar γ , the level set $\mathcal{L}(\gamma)$ is defined as $\mathcal{L}(\gamma) \triangleq \{x \mid x \in \mathcal{D}, f(x) \leq \gamma\}$. The boundary of the level set, $\{x \mid x \in \mathcal{D}, f(x) = \gamma\}$ is known as the level curve associated with γ . ■

For example, if $f(x) = x_1^2 + x_2^2$ and $\gamma = 1$, the level set $\mathcal{L}(1)$ includes the boundary and interior of the circle $x_1^2 + x_2^2 \leq 1$. Note that a level set may be empty for some choices of γ (e.g., for negative γ in the case of $f(x) = x_1^2 + x_2^2$). A convex function has either empty or convex level sets. The proof is left as an exercise.

If the level set associated with a continuous function f is *compact*, i.e., *closed and bounded*, then a standard result of analysis states that f must take on its maximum and

minimum values at points in the level set. Without the two assumptions of boundedness and closedness, it is easy to give examples of level sets that may not even contain a minimizer. For example, consider the univariate function $f(x) = x^3$, which is smooth but *unbounded below* because $f(x) \rightarrow -\infty$ as $x \rightarrow -\infty$. Hence, the level set $f(\gamma)$ is unbounded for *any* γ . Functions that are bounded below can also have unbounded level sets. For example, consider $f(x) = e^x$, which satisfies $f(x) > 0$ for all x . However, the lower bound is not attained for any finite x , and thus the set of x for which $f(x) \leq \gamma$ ($\gamma > 0$) is *unbounded*. On the other hand, a function can be unbounded below and yet have a bounded level set. For example, if $f(x) = x^2 - 1/x^2$, f is unbounded below as $|x|$ approaches zero, yet the set of x for which $f(x) \leq 0$ is the closed bounded interval $[-1, 1]$. (Since functions with singularities are *not* twice-continuously differentiable, they have already been excluded from consideration.)

In many cases, a level set can consist of several isolated subsets, each of which can be bounded or unbounded. Consider the function $f(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$, whose level curves are depicted in Figure 3.1. This function has a local minimizer at the point $x = (\frac{1}{2}, -1)^T$, but $f(x) \rightarrow 0$ as $x_1 \rightarrow -\infty$ and $x_2 \rightarrow +\infty$. In this case, if $0 < \gamma \leq 1$, the level set has two isolated subsets, one of which is bounded and the other is unbounded. Note that $\mathcal{L}(\gamma)$ is either empty or unbounded for all γ .

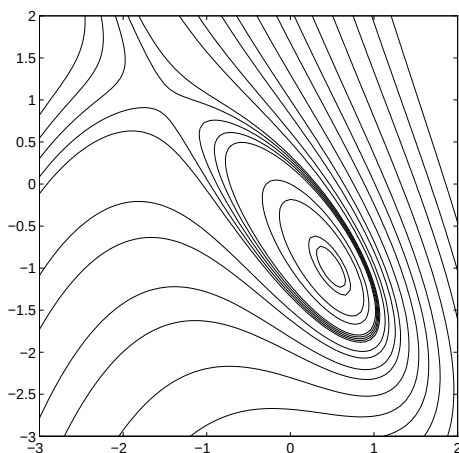


Figure 3.1: Level curves of the function $f(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$. This function has a local minimizer at $x = (\frac{1}{2}, -1)^T$, but $f(x) \rightarrow 0$ as $x_1 \rightarrow -\infty$ and $x_2 \rightarrow +\infty$. In this case, the level set $\mathcal{L}(\gamma)$ has two isolated subsets for $0 < \gamma \leq 1$. One of these subsets is bounded, the other is unbounded.

3.2. Optimality Conditions

We are now ready to derive practical necessary and sufficient conditions for optimality based on the values of the derivatives of f at a candidate solution x^* . The definition 3.1.1

is not “practical”, since it offers no insight into how one might actually verify that x^* possesses the specified property of a minimizer. (Realistically, it would be impossible to check the value of f at the infinite set of points in an interval containing x^* .) Fortunately, information about the behavior of a function in an entire neighborhood can be deduced from properties at a single point when certain smoothness properties hold.

3.2.1. Functions of one variable

We start by considering the simplest case, which involves a function f of one variable. In this case, we assume that f is defined on a set $\mathcal{D}$ consisting of points in the open interval (a, b) , and we take the open ball $\mathcal{B}(x^*, \delta)$ as the open interval $\{x \mid |x - x^*| < \delta\}$.

If f is continuously differentiable on (a, b) (i.e., the derivative f' exists and is continuous) then the following two theorems give *necessary* conditions (i.e., conditions that must hold) at a unconstrained local minimizer of f .

Theorem 3.2.1. (*First-order necessary condition for a univariate local minimizer*). *If f is continuously differentiable on (a, b) , a point $x^* \in (a, b)$ is a local minimizer of f only if $f'(x^*) = 0$.*

Proof. Suppose that x^* is a minimizer such that $f'(x^*) \neq 0$. For the moment, assume that $f'(x^*) < 0$. The continuity of f' implies that we can always find a small positive number δ such that $x^* + h \in (a, b)$ and $f'(x^* + h) < 0$ for all $0 < h \leq \delta$. It follows that if $0 < h \leq \delta$, then $\int_0^1 f'(x^* + \xi h) d\xi \leq 0$ for all $0 \leq \xi \leq 1$. The Taylor-series expansion with integral remainder term gives

$$f(x^* + h) - f(x^*) = \int_0^1 f'(x^* + \xi h) h d\xi < 0.$$

This implies that if $f'(x^*) < 0$ then there is a positive constant δ for which $f(x^* + h) < f(x^*)$, for all $h \in (0, \delta)$. A similar argument shows that if $f'(x^*) > 0$, there is a positive constant δ for which $f(x^* + h) < f(x^*)$, for all $h \in (-\delta, 0)$. Clearly $f'(x^*)$ must be zero, since otherwise, any sufficiently small neighborhood of x^* would include points with lower values of f . ■

In our proof we used the so-called “integral form” of the remainder for a Taylor-series expansion. The expansion of $f(x + h)$ is usually written with a linear remainder term involving the next highest derivative evaluated at some point in $[x, x + h]$, i.e.,

$$f(x + h) = f(x) + f'(x + \theta h)h, \quad \text{where } 0 \leq \theta \leq 1.$$

Here and elsewhere we use the equivalent first-order expansions with an *integral* remainder term, which may be stated as

$$f(x + h) = f(x) + \int_0^1 f'(x + \xi h) h d\xi.$$

In the single-variable case, either form of the remainder is acceptable, but we prefer the integral form because this generalizes to the case where f is a vector-valued function.

We conclude from Theorem 3.2.1 that a zero derivative is a *necessary* condition for x^* to be a local minimizer. Such a condition is designated as *first-order* because it involves the first derivative of f . A point x^* at which $f'(x^*) = 0$ is called a *stationary point* of f . Although $f'(x^*)$ must be zero when x^* is a minimizer, the first derivative may be zero at points that are not unconstrained minimizers—for example, consider the function $\sin x$ at $x = \pi/2$. The next theorem shows that if a function has continuous *second* derivatives, then a further condition must hold at a minimizer.

Theorem 3.2.2. (*Second-order necessary conditions for a univariate local minimizer*). Let f denote a twice-continuously differentiable on (a, b) . A point $x^* \in (a, b)$ is an unconstrained local minimizer of f only if

$$f'(x^*) = 0 \quad \text{and} \quad f''(x^*) \geq 0.$$

Proof. Theorem 3.2.2 is established by contradiction. Assume that x^* is a minimizer such that $f''(x^*) < 0$.

The Taylor-series expansion with a second-order integral remainder term gives the objective at $f(x^* + h)$ in the form

$$f(x^* + h) = f(x^*) + hf'(x^*) + h^2 \int_0^1 (1 - \xi) f''(x^* + \xi h) d\xi.$$

If an additional term $\frac{1}{2}h^2 f''(x^*)$ is included on the right-hand side of this expression, we obtain

$$f(x^* + h) = f(x^*) + hf'(x^*) + \frac{1}{2}h^2 f''(x^*) + h^2 \int_0^1 [f''(x^* + \xi h) - f''(x^*)](1 - \xi) d\xi.$$

Using the result of Theorem 3.2.1, $f'(x^*)$ must be zero at a minimizer, and as long as h is not zero we can rearrange the last equation so that

$$\frac{f(x^* + h) - f(x^*)}{h^2} = \frac{1}{2}f''(x^*) + \int_0^1 [f''(x^* + \xi h) - f''(x^*)](1 - \xi) d\xi. \quad (3.2.1)$$

The assumed continuity of f'' implies that $f''(x^* + \xi h) \rightarrow f''(x^*)$ as $\xi \rightarrow 0$ and it follows that the integrand on the right-hand side can be made as small as we wish by choosing h sufficiently small. In particular, since $f''(x^*)$ is negative, the value of h can always be chosen small enough that the right-hand side of (3.2.1) is negative. We can state this argument precisely as follows. Let $\lambda = f''(x^*)$ and let δ be a positive number such that, for all $|h| \leq \delta$, the inequality

$$\frac{1}{4}\lambda < \int_0^1 [f''(x^* + \xi h) - f''(x^*)](1 - \xi) d\xi < -\frac{1}{4}\lambda$$

holds. It follows that $(f(x^* + h) - f(x^*)) / h^2 \leq \frac{1}{2}\lambda - \frac{1}{4}\lambda = \frac{1}{4}\lambda < 0$, and $f(x^*) > f(x^* + h)$ for all $|h| \leq \delta$, which contradicts the assumption that x^* is a minimizer of f . Consequently, x^* cannot be a local minimizer if $f''(x^*) < 0$. ■

The condition $f''(x^*) \geq 0$ is called a *second-order* necessary condition because it involves the second derivative of f . Note that $f''(x)$ gives the *curvature* of f at x , and the

requirement that $f''(x^*) \geq 0$ is a statement that the curvature at a minimizer must be non-negative. Pictorially, f cannot curve downward at a minimizer.

It is important to realize that if $f'(x^*) = 0$ and $f''(x^*) = 0$, then x^* may not be a local unconstrained minimizer, even though x^* satisfies the second-order necessary conditions $f'(x^*) = 0$ and $f''(x^*) \geq 0$. For example, both the first and second derivative of $f(x) = x^3$ are zero at $x^* = 0$, yet zero is not a minimizer of this function. A zero second derivative may or may not imply optimality (for example, $f(x) = x^4$ *does* have a minimizer at $x = 0$). The problem is that if $f''(x^*) = 0$, then it is not possible to determine the optimality of x^* without looking at derivatives higher than second order.

The practical benefit of *necessary* optimality conditions is that their failure to be satisfied at a point immediately implies that the point cannot be a minimizer. *Sufficient* optimality conditions such as those in the following theorem have a different flavor, in that their satisfaction at a point *guarantees* that the point is a minimizer. The next theorem is proved later in the section.

Theorem 3.2.3. (*Sufficient conditions for a univariate local minimum*). Let $f(x)$ be twice-continuously differentiable. If $f'(x^*) = 0$ and $f''(x^*) > 0$, then x^* is an isolated local minimizer of f . ■

It is important to note that a strict minimizer is not necessarily an isolated minimizer. For example, consider the function

$$f(x) = \begin{cases} x^2 (1 + x^2 + \sin(1/x)) & \text{if } x \neq 0; \\ 0 & \text{if } x = 0. \end{cases}$$

This function has the property that $f(x) \geq 0$ for all $x \in \mathbb{R}$, and $x = 0$ is a strict local (and global) minimizer of f . However, in every neighborhood of 0, $f'(x)$ changes sign infinitely often and f has infinitely many local minimizers in each of these neighborhoods.

3.2.2. Functions of many variables

We now return to the problem of finding the least value of a function of many variables. Throughout this section, the gradient and Hessian of f will be denoted by $g(x)$ and $H(x)$.

Every point x in a neighborhood of x^* can be reached by taking the appropriate step along a vector p in the direction of $x - x^*$. Hence, by allowing a vector p to take on every value in $\mathbb{R}^n$ we can characterize the neighborhood of x^* as points of the form $x^* + \alpha p$ with α sufficiently small. This allows us to define a minimizer in terms of the univariate function $f(x^* + \alpha p)$ defined for all possible directions p emanating from x^* . More precisely, the definition of a minimizer can be restated as follows: a point x^* is a minimizer of f if and only if for every vector $p \in \mathbb{R}^n$, there exists a positive σ such that

$$f(x^*) \leq f(x^* + \alpha p) \quad \text{for all } \alpha \in [0, \sigma]. \quad (3.2.2)$$

This is the same as saying that $\alpha^* = 0$ must be a minimizer of the univariate function $f(x^* + \alpha p)$ for every p in $\mathbb{R}^n$. If $\psi(\alpha) \triangleq f(x^* + \alpha p)$, the first- and second-order necessary conditions for $\alpha^* = 0$ to be a minimizer of $\psi(\alpha)$ are

$$\psi'(0) = 0 \quad \text{and} \quad \psi''(0) \geq 0.$$

Using the chain rule of differentiation (see §1.7.2), the derivatives are given by

$$\psi'(0) = g(x^*)^T p \quad \text{and} \quad \psi''(0) = p^T H(x^*) p.$$

The next theorem follows immediately.

Theorem 3.2.4. (*Necessary condition for a local minimizer*). Let f be a twice continuously differentiable function defined on an open convex set $\mathcal{D} \subset \mathbb{R}^n$. A point $x^* \in \mathcal{D}$ is a local minimizer of f only if

$$\nabla f(x^*) = g(x^*) = 0 \quad \text{and} \quad p^T H(x^*) p \geq 0 \quad \text{for every } p \in \mathbb{R}^n. \quad (3.2.3)$$

Proof. If x^* is a local minimizer of f , then the restricted function $\psi(\alpha) = f(x^* + \alpha p)$ has a local minimizer $\alpha^* = 0$ for every $p \in \mathbb{R}^n$. This implies that $g(x^*)^T p = 0$ for all p , which must mean that $\nabla f(x^*) = 0$, since we may choose $p = e_1, p = e_2, p = e_3, \dots, p = e_n$, where e_j is the j th coordinate vector. Also, since $\psi''(0) = p^T H(x^*) p$, the second statement follows from the second-order necessary condition for the optimality of $\alpha^* = 0$. ■

The condition $p^T H(x^*) p \geq 0$ for every $p \in \mathbb{R}^n$ is equivalent to the condition that $H(x^*)$ be positive semi-definite (see §1.2.1). This implies the following equivalent characterization of optimality.

Theorem 3.2.5. (*Second-order necessary condition for a local minimizer*). Let f be a twice continuously differentiable function defined on an open convex set $\mathcal{D} \subset \mathbb{R}^n$. A point $x^* \in \mathcal{D}$ is a local minimizer of f only if $g(x^*) = 0$ and $H(x^*)$ is positive semidefinite. ■

The requirement of Theorem 3.2.4 that $g(x^*) = 0$ is called a *first-order condition* for optimality, since it involves the first partial derivatives of f . The necessary condition of Theorem 3.2.5, which involves the second partial derivatives of f , is called a *second-order condition*.

A point $x^* \in \mathcal{D}$ such that $g(x^*) = 0$ is called a *stationary point* of f . Theorem 3.2.4 shows that a local minimizer of a smooth function is a stationary point. However, a zero gradient at x^* is not enough to ensure that x^* is a minimizer. For example, the gradient is also zero at every local *maximizer* of f . Furthermore, the gradient can be zero at a point that is neither a local minimizer nor a local maximizer; such a point is called a *saddle point*. For example, consider the function $f(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$ of Figure 3.1. This function has a local minimizer at the point $x = (\frac{1}{2}, -1)^T$ and a saddle point at $x = (-\frac{3}{2}, 1)$. As $x_1 \rightarrow -\infty$ and $x_2 \rightarrow +\infty$ both f and the components of g go to zero.

It is important to note that the first- and second-order conditions above are *necessary* but not *sufficient* conditions for optimality. To emphasize this point we could say that an *equivalent* form of Theorem 3.2.4 is that if $g(x^*) \neq 0$ or $H(x^*)$ is indefinite, then x^* cannot be a local minimizer of f .

As discussed in §1.2.1, positive-definiteness condition may be conveniently stated in terms of the eigenvalues of $H(\bar{x})$ as follows:

1. Compute $g(\bar{x})$, $H(\bar{x})$, and the eigenvalues of $H(\bar{x})$.

2. If $g(\bar{x}) \neq 0$ or there is at least one negative eigenvalue of $H(\bar{x})$, then $\bar{x}$ cannot be a local minimizer of f .

Since Theorems 3.2.4 and 3.2.5 provide only necessary conditions for optimality, we cannot say that an $\bar{x}$ satisfying the conditions above will be an unconstrained local minimizer. The following example illustrates this point. Consider the function $f(x) = x_1^3 \cos x_3 + x_2^2$, which has gradient and Hessian

$$g(x) = \begin{pmatrix} 3x_1^2 \cos x_3 \\ 2x_2 \\ -x_1^3 \sin x_3 \end{pmatrix}, \quad H(x) = \begin{pmatrix} 6x_1 \cos x_3 & 0 & -3x_1^2 \sin x_3 \\ 0 & 2 & 0 \\ -3x_1^2 \sin x_3 & 0 & -x_1^3 \cos x_3 \end{pmatrix}.$$

At the point $\bar{x} = 0$ we have

$$f(\bar{x}) = 0, \quad g(\bar{x}) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \text{and} \quad H(\bar{x}) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

so that $g(\bar{x})$ is zero and $H(\bar{x})$ has eigenvalues $(0, 2, 0)$. Our verification procedure above implies that $\bar{x}$ satisfies the first- and second-order necessary conditions for optimality.

However, $\bar{x}$ is not a minimizer. Consider the direction $d = (-1, 0, 1)^T$ and the values of f restricted to the line segment

$$x(h) = \bar{x} + \alpha d = \begin{pmatrix} -\alpha \\ 0 \\ \alpha \end{pmatrix}.$$

We have $f(x(\alpha)) = -\alpha^3 \cos \alpha \leq f(\bar{x})$ for all $\alpha \in [0, \pi/2]$. This implies that $\bar{x}$ cannot be a local minimizer of f since every open ball $\mathcal{B}(x^*, \delta)$ with radius δ less than $\pi/2$ must contain points that have lower objective values than $f(\bar{x})$.

In order to state that a point x^* is a local minimizer, we must use the following *sufficient* conditions for optimality, which exclude the case where the Hessian is singular.

Theorem 3.2.6. (*Sufficient conditions for a local minimizer*). Let f be a twice continuously differentiable function defined on an open convex set $\mathcal{D} \subset \mathbb{R}^n$. If $x^* \in \mathcal{D}$ satisfies $g(x^*) = 0$ and $H(x^*)$ positive definite, then x^* is an isolated local minimizer of f .

Proof. Since $H(x^*)$ is positive definite, its smallest eigenvalue $\lambda_{\min}$ satisfies

$$\lambda_{\min} = \min_{s^T s = 1} s^T H(x^*) s > 0.$$

Since $H(x)$ is continuous, there exists a positive δ such that, for every $\|s\| < \delta$, the point $x^* + s$ lies in $\mathcal{D}$ and

$$\|H(x^* + s) - H(x^*)\| < \frac{1}{2} \lambda_{\min}.$$

Using an approach similar to that of Theorem 3.2.2, we obtain

$$\begin{aligned} \frac{f(x^* + s) - f(x^*)}{s^T s} &= \frac{1}{2s^T s} s^T H(x^*) s + \frac{1}{s^T s} \int_0^1 s^T (H(x^* + \xi s) - H(x^*)) s (1 - \xi) d\xi \\ &\geq \frac{1}{2} \lambda_{\min} - \frac{1}{2} \max_{0 \leq \xi \leq 1} \|H(x^* + \xi s) - H(x^*)\| \\ &> \frac{1}{4} \lambda_{\min}. \end{aligned}$$

Therefore, $f(x^* + s) > f(x^*)$ for all $\|s\| < \delta$ ($s \neq 0$), which proves that x^* is an isolated local minimizer.

In order to show that x^* is an isolated minimizer, we must show that there exists a neighborhood of x^* such that for every point, say $\bar{x}$ in this neighborhood and different from x^* cannot be a local minimizer. A necessary condition for $\bar{x}$ to be a local minimizer is $g(\bar{x}) = 0$. Hence it is sufficient to show that in $\mathcal{B}$, all points other than x^* have a nonzero gradient, i.e., $g(\bar{x}) \neq 0$. Let $\bar{x} \in \mathcal{B}$ satisfy $g(\bar{x}) = 0$. If we write $\bar{x} = x^* + p$ for some $p \neq 0$, then by Taylor's theorem $g(x^* + p)^T p = g(x^*)^T p + p^T H(x^* + \theta p)p$, where $\theta \in (0, 1)$. Since $g(x^*) = 0$ and $p^T H(x^* + \theta p)p > 0$ for $\|p\|$ sufficiently small, it follows that $g(x^* + p)^T p > 0$, which is a contradiction. It follows that $g(\bar{x}) \neq 0$ and hence x^* is truly isolated.

■

It is typical that proofs of necessary conditions are proved by contradiction (we assume that the various conditions are not true, and then show that the given point cannot be a minimizer), whereas proofs of sufficient conditions tend to be “constructive”, i.e., we show that the definition of a minimizer is satisfied by the given point when the stated conditions apply. Development of conditions that are both necessary and sufficient is extremely complicated, and will not be considered here.

3.2.3. An alternative derivation

In this section we consider a more “constructive” derivation of optimality conditions that will give some insight into the formulation of algorithms. In particular, we will focus on directions for which condition (3.2.2) *does not hold*. Such vectors are of particular interest since they provide a direction along which lower values of f can be found.

Definition 3.2.1. A direction p is a direction of decrease for f at x if there exists a positive σ such that $f(x + \alpha p) < f(x)$, for all $\alpha \in (0, \sigma)$. ■

The following theorem gives an equivalent definition of a minimizer in terms of all directions p emanating from x^* .

Theorem 3.2.7. Let f be a function defined on an open convex set $\mathcal{D}$. A point $x^* \in \mathcal{D}$ is a local minimizer of f only if no direction of decrease exists at x^* . ■

If f is continuously differentiable, it is possible to give a simple verifiable sufficient condition under which a direction of decrease exists. If f is initially decreasing along p , i.e., if

$$\left. \frac{d}{d\alpha} f(x + \alpha p) \right|_{\alpha=0} = g(x)^T p < 0,$$

then p must be a direction of decrease. To show this we use the continuity of f to guarantee the existence of a positive σ such that $x + \xi p \in \mathcal{D}$ and $g(x + \xi p)^T p < 0$ for all $\xi \in [0, \sigma]$. For any $\alpha \in (0, \sigma)$, the Taylor-series expansion with integral remainder term gives

$$f(x + \alpha p) - f(x) = \int_0^\alpha g(x + \xi p)^T p d\xi < 0.$$

This implies that any direction for which $g(x)^T p < 0$, there is a positive constant σ for which $f(x + \alpha p) < f(x)$, for all $\alpha \in (0, \sigma)$.

A vector p such that $g(x)^T p < 0$ is known as a *descent direction* for f at x . A descent direction is always a direction of decrease for f , but a direction of decrease is not necessarily a descent direction. For example, if x is a stationary point of f at which the Hessian has some negative eigenvalues, any eigenvector v associated with a negative eigenvalue is a direction of decrease at x . However, $p = v$ is not a descent direction since $g(x)^T p = 0$ for all p .

These results imply an equivalent first-order optimality condition that x^* is a local minimizer only if no descent directions exist at x^* , i.e., only if $g(x^*)^T p \geq 0$ for all $p \in \mathbb{R}^n$. In the unconstrained case, this is the same as saying that $g(x^*)^T p = 0$ for all $p \in \mathbb{R}^n$, since otherwise we could always define a descent direction $\bar{p} = -p$ for which $g(x^*)^T \bar{p} < 0$.

If second derivatives are available, a larger class of directions of decrease can be identified. In particular, p will be a direction of decrease if either $g(x)^T p < 0$, or $g(x)^T p = 0$ and $p^T H(x)p < 0$. The first condition is just the statement that p is a descent direction, which we have already shown to be a direction of decrease. The second condition follows from the second-order Taylor-series expansion

$$\frac{f(x^* + \alpha p) - f(x^*)}{\alpha^2} = \frac{1}{2} p^T H(x^*) p + \int_0^1 p^T (H(x^* + \xi \alpha p) - H(x^*)) p (1 - \xi) d\xi.$$

Since $p^T H(x^*) p < 0$ there must exist a positive σ such that

$$\frac{1}{2} p^T H(x^*) p + \int_0^1 p^T (H(x^* + \xi p) - H(x^*)) p (1 - \xi) d\xi < 0,$$

for all $\alpha \in (0, \sigma)$. This implies that $f(x^* + \alpha p) - f(x^*) < 0$ for all $\alpha \in (0, \sigma)$ and p is a direction of decrease.

A vector p such that $p^T H(x)p < 0$ is known as a *direction of negative curvature* for f at x . Note that if p is a direction of negative curvature then so is $-p$. It follows that the sign of a direction of negative curvature can always be selected to ensure that $g(x)^T p \leq 0$.

3.2.4. Properties of quadratic functions

Quadratic model functions play an important role in the development of minimization algorithms, and we accordingly devote this section to a brief look at certain key properties of the quadratic function

$$q(x) = c^T x + \frac{1}{2} x^T H x, \quad (3.2.4)$$

where c is a constant vector and H is a symmetric matrix. Note that the gradient of (3.2.4) is

$$\nabla q(x) = c + Hx, \quad (3.2.5)$$

and that $\nabla^2 q(x) = H$ (i.e., the Hessian is constant). Since the third derivative is constant, the Taylor-series expansion of $q(x + d)$ has three terms and is given by

$$q(x + d) = q(x) + d^T \nabla q(x) + \frac{1}{2} d^T H d. \quad (3.2.6)$$

In order for x to be a *stationary point* of $q(x)$, it must hold that

$$\nabla q(x) = c + Hx = 0, \quad \text{or} \quad Hx = -c. \quad (3.2.7)$$

If H is nonsingular, this shows that $q(x)$ has a unique stationary point. If, in addition, H is positive definite, then Theorem 3.2.6 implies that the stationary point is an *isolated local minimizer*, denoted by x^* . In the positive-definite case, the point x^* for which $Hx^* = -c$ is the (unique) minimizer of $q(x)$. If $x = 0$, then $q(x) = 0$. The minimal value of q is therefore

$$q(x^*) = c^T x^* + \frac{1}{2} x^{*T} H x^* = c^T x^* + \frac{1}{2} x^{*T} (-c) = \frac{1}{2} c^T x^* \leq 0. \quad (3.2.8)$$

If H is *indefinite* and nonsingular, then the unique stationary point of q is a *saddle point*, i.e., neither a minimizer or a maximizer. In this case, $q(x)$ is *unbounded below*. To see why, note that the definition of an indefinite matrix implies the existence of y such that $y^T H y < 0$. For any scalar γ , the Taylor-series expansion (3.2.6) gives

$$q(x^* + \gamma y) = q(x^*) + \gamma \nabla q(x^*)^T y + \frac{1}{2} \gamma^2 y^T H y = q(x^*) + \frac{1}{2} \gamma^2 y^T H y.$$

The second (negative) term implies that q can be made arbitrarily large and negative by moving from x^* along y . A similar argument shows that $q(x)$ can be made unbounded above by moving along a direction z such that $z^T H z > 0$.

The only remaining possibility is that H is *singular*. In this case, q will have a stationary point if $Hx = -c$ can be satisfied for some vector x . This will occur if and only if c lies in the range space of the columns of H . If $c \notin \text{range}(H)$, no stationary point of $q(x)$ exists, and q is unbounded below. (The term $c^T x$ can be made arbitrarily large and negative by choosing x as an appropriate vector in $\text{null}(H^T)$.)

If c is in the range space of H , then a nonzero stationary point of q exists, but it is *not* unique. The singularity of H implies the existence of a nonzero vector z such that $H z = 0$. If (3.2.7) is satisfied for some vector x^* , so that $c + Hx^* = 0$, then it also holds that $c + H(x^* + \beta z) = 0$ for every scalar β , and hence (3.2.7) holds for an infinite family of vectors $x^* + \beta z$.

If H is positive semidefinite and a vector x^* exists such that $Hx^* = -c$, then x^* is a local minimizer of $q(x)$, as is every vector $x^* + \beta z$, where $H z = 0$. Using the Taylor-series expansion (3.2.6) about x^* , we have

$$q(x^* + \beta z) = q(x^*) + \beta z^T \nabla q(x^*) + \frac{1}{2} \beta^2 z^T H z = q(x^*), \quad (3.2.9)$$

since $\nabla q(x^*) = 0$ and $H z = 0$. For any other direction y such that $H y \neq 0$, the positive semidefiniteness of H implies that $y^T H y > 0$, and hence, again using the Taylor-series expansion,

$$q(x^* + \beta y) = q(x^*) + \frac{1}{2} \beta^2 y^T H y > q(x^*). \quad (3.2.10)$$

The implication of (3.2.9) and (3.2.10) is that x^* is a (nonunique) local minimizer if H is positive semidefinite. Notice, however, that the optimal objective function value $q(x^*)$ is unique.

If H is singular and c is in the range space of H , but H is not positive semidefinite, then q has a nonunique stationary point, but is unbounded below.

The results above are summarized in the following theorem and corollary.

Lemma 3.2.1. *If H is positive semi-definite, then the quadratic function $q(x)$ achieves its bounded global minimum at a point x^* if $Hx^* = -c$. If H is positive definite, the minimizer x^* is unique. If H is positive semi-definite and singular, there is an infinite number of minimizers of the form $x^* + z$, where $z \in \text{null}(H)$. If no vector x exists such that $Hx = -c$, or if H is indefinite, then $q(x)$ is unbounded below. ■*

Corollary 3.2.1. *The necessary and sufficient conditions for x^* to be a global minimizer of the quadratic function $q(x)$ are:*

- (a) $\nabla q(x^*) = c + Hx^* = 0$; and
- (b) H positive semidefinite. ■

To illustrate the two main cases of interest, let

$$H_1 = \begin{pmatrix} 3 & 2 & 1 \\ 2 & 4 & 1 \\ 1 & 1 & 5 \end{pmatrix}, \quad H_2 = \begin{pmatrix} 3 & 2 & 1 \\ 2 & 3 & -1 \\ 1 & -1 & 2 \end{pmatrix}, \quad \text{and} \quad c = \begin{pmatrix} 7 \\ 8 \\ -1 \end{pmatrix}.$$

The matrix H_1 is positive definite, with eigenvalues 6.71, 3.86 and 1.43 (rounded to three digits), and the quadratic function (3.2.4) has a unique minimizer, at $x^* = (-1.70, -1.35, .81)^T$. In contrast, H_2 is positive semidefinite, with eigenvalues 5, 3 and 0, with $\text{null}(H_2)$ defined as the set of vectors

$$H_2 z = 0 \quad \Rightarrow \quad z = \begin{pmatrix} -\gamma \\ \gamma \\ \gamma \end{pmatrix},$$

for any scalar γ . Since c is a linear combination of the columns of H_2 , the associated quadratic function has an infinite family of (weak) local minimizers, of the form

$$x^* = \begin{pmatrix} -1 \\ -2 \\ 0 \end{pmatrix} + \begin{pmatrix} -\gamma \\ \gamma \\ \gamma \end{pmatrix}.$$

The (unique) optimal value of $q(x)$ is assumed at all such points, and, from (3.2.8), is $q(x) = \frac{1}{2}c^T x^* = -\frac{23}{2}$.

3.3. Minimizing a Function of One Variable

3.3.1. Interval reduction techniques

To locate an unconstrained minimizer of a smooth function in a bounded interval, *interval reduction* techniques can be developed that are similar in motivation to the bisection method for zero-finding (see §2.2.1).

Before specifying an interval reduction procedure, it is useful to impose conditions that ensure the existence of an *unconstrained* minimizer in a given interval $[a, b]$. (An analogous condition for zero-finding is that $f(a)f(b) < 0$.) We wish to avoid the situation when a minimizer occurs at an *endpoint* of the interval and does not satisfy the necessary conditions of Theorem 3.2.2—for example, the minimizer of $f(x) = x$ in the interval $[1, 2]$ occurs at $x = 1$, which is not an unconstrained minimizer. Many different sets of conditions can be defined. For example, when f is continuously differentiable, the conditions that $f'(a) < 0$ and $f'(b) > 0$ ensure the existence of an unconstrained minimizer in $[a, b]$.

A “classic” approach to interval reduction is based on *function comparison* within an interval containing an unconstrained minimizer. For simplicity, we assume the property of *unimodality*, which is defined in several different ways in the literature and allows us to make the strong assertion that only a *single* unconstrained minimizer lies in an interval of uncertainty.

Definition 3.3.1. (*Unimodality.*) *The function $f(x)$ is unimodal in the interval $[a, b]$ if there exists a $x^* \in [a, b]$ at which $f(x)$ attains its minimum value, and that, given any two points $u, v \in [a, b]$ for which $u < v$:*

$$\begin{aligned} \text{if } v < x^* \text{ then } f(u) &> f(v); \\ \text{if } u > x^* \text{ then } f(u) &< f(v). \end{aligned}$$

The next definition does not require the existence of the point x^* .

Definition 3.3.2. (*Unimodality.*) *The function $f(x)$ is unimodal in the interval $[a, b]$ if there exists a unique $x^* \in [a, b]$ such that, given any three points $u, v, w \in [a, b]$ for which $u < v < w$:*

$$\begin{aligned} \text{if } f(u) \leq f(v) \text{ then } f(v) &< f(w); \\ \text{if } f(v) \geq f(w) \text{ then } f(u) &> f(v). \end{aligned}$$

If f is continuous, then x^* must exist (see Theorem 1.6.2) and these definitions are equivalent.

Unimodality is relevant in interval reduction methods because it allows information about the location of x^* to be deduced by a comparison of two interior function values. In particular, suppose that a minimizer of f is known to lie in an interval of uncertainty $[a, b]$, that f is unimodal in $[a, b]$, and that the values of f are known at two points, say u and v , where $a < u < v < b$. It follows from unimodality that x^* must lie in $[a, v]$ if $f(v) > f(u)$, and in $[u, b]$ if $f(u) > f(v)$. If $f(u) = f(v)$, we know that x^* must lie between u and v , so that either subinterval is acceptable. (Without an assumption of unimodality or further information, the “correct” choice of subinterval when $f(u) = f(v)$ is unclear.) In all cases, the interval of uncertainty is reduced following the comparison of $f(u)$ and $f(v)$.

To carry out an interval reduction strategy in the context of minimization, we thus require *two* values of f in the current interval. Algorithm 3.3.1 describes a general interval reduction strategy.

Various strategies can be developed for selecting the two points u and v at each step. Given the general principle that evaluation of f is “expensive”, it would be desirable to

Algorithm 3.3.1. A general interval-reduction strategy.

```

 $a \leftarrow a_0; \quad b \leftarrow b_0;$ 
while  $b - a > \tau_x$  do
    Define two points  $u, v \in (a, b)$ , with  $u < v$ ;
    Evaluate  $f(u), f(v)$ ;
    if  $f(u) < f(v)$  then
         $b \leftarrow v$  else  $a \leftarrow u$ ;
end while

```

choose the points so that one of them could be *re-used* at the next iteration, thereby requiring only one new evaluation of f .

The interval reduction strategy that is optimal in the sense of yielding the maximum reduction in the interval of uncertainty for a *specified number* of function evaluations is called *Fibonacci search*. The trial points are chosen based on the well known Fibonacci numbers 1, 1, 2, 3, 5, 8, 13, $\dots$, which define a sequence whose elements $\{F_i\}$ are given by

$$F_0 = F_1 = 1, \quad F_k = F_{k-1} + F_{k-2}, \quad k \geq 2.$$

To use Fibonacci search, the number N of function evaluations is specified in advance, and the initial interval of uncertainty is interpreted as a “scaled” interval of length F_N . It is easiest to explain the idea by example, as follows.

First, we must consider the special case when $N = 2$, so that two function evaluations are allowed. The trial points are the two points in the *original* interval that would correspond to the position of the points $u = F_0$ and $v = F_1$ in the interval $[0, F_2]$, i.e., $[0, 2]$. As $F_0 = F_1 = 1$, two distinct trial points are created by choosing $u = 1$ and $v = 1 + \epsilon$, where ϵ is “negligible” but large enough to make $f(u)$ different from $f(v)$. If $f(u) < f(v)$, the new interval of uncertainty is $[0, 1 + \epsilon]$; if $f(u) \geq f(v)$, the new interval of uncertainty is $[1, 2]$. In either case, the interval has been reduced to essentially *half* its original size.

When $N = 3$, $F_N = 3$, and the original interval of uncertainty is treated as $[0, 3]$. The first two trial points are placed at $u = F_1 = 1$ and $v = F_2 = 2$, i.e., at one-third and two-thirds of the original interval. The reduced interval is $[0, 2]$ if $f(u) < f(v)$ and $[1, 3]$ if $f(u) \geq f(v)$, and has length two in either case. Since the function value at the midpoint of the reduced interval is (by construction) already available, only one additional evaluation is required to obtain a final interval of length at most $F_1 + \epsilon = 1 + \epsilon$.

These cases illustrate the basic property of intervals defined using the Fibonacci numbers: given any Fibonacci number F_k and an interval of uncertainty $[0, F_k]$, if f is evaluated at F_{k-2} and F_{k-1} (the points “ u ” and “ v ” respectively) then the resulting interval of uncertainty is of length F_{k-1} *irrespective of the behavior of $f(x)$* . With an initial interval of uncertainty $[0, F_N]$, f is first evaluated at $u = F_{N-2}$ and $v = F_{N-1}$, giving an interval of uncertainty of length F_{N-1} . The next points u' and v' are just the “scaled” versions of the points F_{N-2} and F_{N-3} in the interval $[0, F_{N-1}]$. The scheme continues in this way, obtaining a sequence of intervals of length $F_N, F_{N-1}, \dots, F_1, F_0$. Whatever the value of N , the last two trial points must be made distinct using the perturbation ϵ .

At each stage of Fibonacci search, one of the trial points coincides with a trial point of the previous step. This property is most obvious after the first step. As described above,

let u and v denote the points F_{N-2} and F_{N-1} selected for the interval $[0, F_N]$. There are two possible outcomes for the interval of uncertainty. If the new interval is $[0, F_{N-1}]$, $u' = F_{N-2}$ and the function at u' is known already. Alternatively, if the new interval of uncertainty is $[F_{N-2}, F_N]$,

$$u' = \frac{F_{N-3}}{F_{N-1}}(F_N - F_{N-2}) + F_{N-2}.$$

If we use the property of the Fibonacci numbers that $F_N = F_{N-1} + F_{N-2}$, then

$$u' = \frac{F_{N-3}}{F_{N-1}}F_{N-1} + F_{N-2} = F_{N-3} + F_{N-2} = F_{N-1},$$

which is just v from the previous step. In either case, only one function value is required at each step, as required.

The definition of u and v is easily generalized to an arbitrary interval $[a_k, b_k]$ by performing an appropriate transformation of points in $[0, F_k]$. At the k -th stage, the trial points in the interval of uncertainty $[a_k, b_k]$ are:

$$u_{k+1} = \frac{F_{N-k-2}}{F_{N-k}}(b_k - a_k) + a_k \quad \text{and} \quad v_{k+1} = \frac{F_{N-k-1}}{F_{N-k}}(b_k - a_k) + a_k,$$

where these formulae apply for all $k = 0, 1, \dots, N-2$. At the last stage we must perturb either u_{N-1} or v_{N-1} by ϵ (the point to be perturbed depends on which was not computed at the previous stage).

A Fibonacci search procedure will produce a final interval of uncertainty of length essentially $1/F_N$ times the length of the original interval. (We must say “essentially” because of the special case associated with the last step of the reduction.)

Although Fibonacci search is of some theoretical interest because of its “optimal” properties, it is rarely used in practice because the number of function evaluations to be permitted must be specified in advance, and the choice of points depends on this number. A less restrictive procedure that is almost identical in behavior when N is not too small is called *golden section search*, and is based on the limiting ratio of the Fibonacci numbers. It can be shown that

$$\lim_{k \rightarrow \infty} \frac{F_{N-1}}{F_N} = \frac{2}{1 + \sqrt{5}} \equiv \tau \approx .6180,$$

where τ satisfies the quadratic equation $\tau^2 + \tau - 1 = 0$.

With a golden section search procedure, the initial interval is “scaled” to be $[0, 1]$, and the two trial points are placed at $u = 1 - \tau$ and $v = \tau$ (i.e., at approximately .3820 and .6180). No matter how the interval is reduced, *one* of the old points will then be in the correct position with respect to the new interval, so that golden section search may thus be viewed as the limiting case of Fibonacci search. With golden section search, the interval of uncertainty is reduced by a factor of τ at every step, and its length converges linearly to zero.

3.3.2. Polynomial interpolation

In interval reduction methods, only a simple comparison of function values is used to choose the next point. Hence, their efficiency does not improve when f is well behaved. As in

the zero-finding case, more rapidly convergent procedures can be developed in which f is modeled by a simple function φ whose minimizer is “easy” to compute, and the minimizer of φ is taken as the next approximation of the minimizer of f .

Since a general straight line has no minimizer, a linear function is not an appropriate model. The obvious next candidate for the model function is a quadratic function, of the form:

$$\varphi(x) = \frac{1}{2}ax^2 + bx + c. \quad (3.3.1)$$

If $a > 0$, φ has a minimizer at x^* such that $ax^* + b = 0$, i.e., $x^* = -b/a$. Three independent pieces of information are required to determine the quadratic function φ (3.3.1), although only a and b are required to calculate x^* .

The particular quadratic model to be used depends on the available information about f . By analogy with the development of Newton’s method (§2.2.2), a quadratic approximation is defined from the first three terms of the Taylor-series expansion of f about the current point x_k :

$$\varphi_k(x) = \varphi_k(x_k + p) = f(x_k) + pf'(x_k) + \frac{1}{2}p^2f''(x_k), \quad (3.3.2)$$

where $p = x - x_k$. This function is a quadratic in p (the step from x_k to x), and its first derivative is

$$\varphi'_k(x) = \varphi'_k(x_k + p) = f'(x_k) + pf''(x_k).$$

If $f''(x_k) > 0$, the step p to the minimizer of the model function (3.3.2) satisfies

$$\varphi'_k(x_k + p) = f'(x_k) + pf''(x_k) = 0,$$

so that the next iterate would be defined by $p = -f'(x_k)/f''(x_k)$, i.e.,

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}. \quad (3.3.3)$$

The iteration scheme (3.3.3) is equivalent to applying Newton’s method (2.2.9) to find a zero of $f'(x)$.

A practical difficulty of the scheme (3.3.3) for univariate minimization is that it requires both first and second derivatives of f . When first but not second derivatives are available, we may use (for example) $f(x_k)$, $f'(x_k)$ and $f(x_{k-1})$ to define (3.3.1). When only function values are available, a quadratic approximation φ (3.3.1) can be defined using the three most recent function values— $f(x_k)$, $f(x_{k-1})$ and $f(x_{k-2})$.

Since *cubic* polynomials are well behaved, another popular strategy is to define the model function φ as

$$\varphi(x) = ax^3 + bx^2 + cx + d. \quad (3.3.4)$$

Four independent pieces of information are needed to define (3.3.4)—most commonly, $f(x_k)$, $f'(x_k)$, $f(x_{k-1})$, $f'(x_{k-1})$. Safeguarded univariate minimization methods based on (3.3.4) are highly developed and widely used in practice.

The cubic (3.3.4) might seem to convey more information than the quadratic (3.3.2). However, a word of caution is in order. Although a more complex model may express a better approximation to the nonlinear function, analysis and interpretation of the model invariably become more difficult. Remember that the purpose of a model function is not

only to “capture the essence” of the nonlinear function, *but also to be “simple” to analyze*. Even a quadratic model may need to be rejected; for example, when it is unbounded below, it seems unlikely to provide good information about the minimizer that we seek. In the case of a cubic model of the form (3.3.4), there may be no real stationary points (e.g., $\varphi(x) = x^3 + x$). Furthermore, even if real stationary points exist, they may not be minimizers of the model function. Therefore, situations can occur in which the interpretation of a cubic model is unclear, and univariate minimization methods based on cubic models tend to be quite complicated.

Methods based upon suitable polynomial models such as (3.3.2) and (3.3.4) display local superlinear convergence under various conditions (as with Newton’s method and the secant method for zero-finding; see §§2.2.2 and 2.2.3). For example, a method based on a quadratic model fitted to the latest three function values has a convergence rate of approximately 1.324. We emphasize, however, that because these methods are based on a well-behaved simple model, they may experience the same difficulty mentioned in our discussion of rapidly convergent zero-finding techniques: when φ does not accurately represent the behavior of f , its minimizer may be a poor estimate of the true minimizer.

Extreme care is necessary if one attempts to “fix up” a superlinearly convergent method to improve reliability, since some seemingly “obvious” modifications may not only destroy fast convergence, but also impair efficiency. (For an analogous situation in zero-finding, see the discussion in §2.2.3 of the regula falsi method.) For example, if one uses a quadratic model with the restriction that the three function values used for the fit always “bracket” the minimizer (e.g., by requiring that the function value at the middle point be lower than the function at either endpoint), the resulting method may display arbitrarily slow linear convergence.

3.3.3. Safeguarded polynomial interpolation

As in zero-finding, the “best” general methods are based on combining rapidly convergent polynomial models with reliable interval reduction procedures. For example, a typical safeguarded method based on a quadratic model derived from three function values keeps an interval of uncertainty (say, $[a, b]$), and the three “best” points found so far. The minimizer s of the quadratic fitted to these three points is accepted as the “new” point u only if: s lies within the interval of uncertainty; “reasonable” progress in reducing the interval has been made during the last few iterations; and s is not “too close” to one of the three points used to obtain the quadratic fit.

3.4. Model-Based Methods for Functions of Many Variables

Algorithms for minimization are closely related to algorithms for zero-finding in the sense that they are iterative and rely on a local model of the problem functions. However, in the case of minimization, there is a much wider choice of available models. As in previous sections $g(x)$ and $H(x)$ are used to denote the gradient ∇f and Hessian $\nabla^2 f$ at a point x .

3.4.1. Choice of model

Since $g(x)$ must be zero at a minimizer, one approach is to use Newton’s method to find a point x^* such that $g(x^*) = 0$. The reader will recall that Newton’s method for the zero

of an n -vector $F(x)$ with Jacobian $J(x)$ involves constructing a local *affine model* of F . If x_k is the current point and $J(x_k)$ is nonsingular, the *Newton direction* p_k is the step from x_k to the zero of the local affine model of F , and is the unique solution of the linear system $J(x_k)p_k = -F(x_k)$. If we take $F(x) \equiv g(x)$ and note that the Jacobian of g is the Hessian $H(x)$, we have the affine model

$$g(x_k + p) \approx g(x_k) + H(x_k)p. \quad (3.4.1)$$

It follows that a zero of $g(x_k) + H(x_k)p$ must satisfy the linear system

$$H(x_k)p_k = -g(x_k). \quad (3.4.2)$$

As background to later developments, we stress that this form of Newton's method is directed *solely toward finding a zero of g* . No additional properties of f are used in deriving (3.4.2), and the quality of p_k as a step to a zero of g depends on the validity of the local affine model of g at x_k .

The second form of Newton's method is designed for *minimization* of f . Recall that a local minimizer satisfies both first- and second-order conditions involving f , and hence is inherently more complicated than a zero of g . A quadratic model of the local variation of f is obtained from the Taylor-series expansion about x_k :

$$f(x_k + p) \approx f(x_k) + g(x_k)^T p + \frac{1}{2} p^T H(x_k) p. \quad (3.4.3)$$

An essential point in any discussion of developing a model subproblem for minimization is that the model must be *appropriate for minimization* (even when this assumption is not be stated explicitly). In particular, the quadratic model (3.4.3) is bounded below only if $H(x_k)$ is positive semidefinite; and the model has a unique minimizer (which is also the unique stationary point) only if $H(x_k)$ is positive definite. The appropriateness of a model for minimization thus inherently involves both its stationarity (to reflect first-order optimality conditions) *and* conditions on positive-definiteness (to reflect second-order optimality conditions).

Under the assumption that $H(x_k)$ is positive definite, there is a unique vector p_k that moves from x_k to the minimizer of the model (3.4.3). This vector satisfies the nonsingular linear system $H(x_k)p_k = -g(x_k)$, which is identical to the system (3.4.2). It follows that when the Hessian is positive definite, the direction p_k derived for minimization is equivalent to the Newton step (3.4.2) to a stationary point of f , i.e., to the zero of the n -dimensional nonlinear system $g(x) = 0$.

The interpretation of $x_k + p_k$ as a zero of a linear model of the gradient explains the rapid local convergence of x_k to x^* when $H(x^*)$ is nonsingular, but does nothing to suggest how the method may be extended to the case where $H(x_k)$ is singular and/or indefinite. By contrast, the interpretation of $x_k + p_k$ as a minimizer of a quadratic model leads to several algorithms that will work when Newton's method will fail.

The choice of model has a significant impact on the strategy used to force convergence from an arbitrary starting point. Convergence may be guaranteed by requiring a sufficient reduction in some *merit function* or *line-search function* at each iteration. For an iterative method, it is important to have some consistent way of deciding whether a new point is "better" than the old point. This idea will recur throughout the discussion of optimization algorithms. If we interpret p_k as being an estimate of the step to a zero of $g(x)$, then it

is appropriate to use the norm of g to measure progress. However, when minimizing f , another, more “natural”, measure of progress is provided by the *value* of f itself. It seems reasonable to require a decrease in f at every iteration, and to ensure that a *descent condition*

$$f(x_{k+1}) < f(x_k) \quad (3.4.4)$$

is satisfied for all $k \geq 0$ (unless $x_k = x^*$). A method that imposes this requirement is called a *descent method*.

The choice of merit function is closely linked to the choice of model function. It is important to be able to show that the merit function can be decreased for a sufficiently small step along the search direction. For example, consider the situation where $H(x_k)$ cannot be computed exactly (e.g., the derivatives are too complicated to evaluate). In both the linear and quadratic model, $H(x_k)$ can be approximated by a matrix B_k . Such approximations are difficult to use with the affine model (3.4.1) because the resulting direction p_k may not be a direction of decrease for $\|g\|$. However, if a positive-definite approximation B_k is used in place of H in the quadratic model, then the resulting direction is always a direction of decrease for f (see below).

3.4.2. The Quadratic Model

In this section we focus on the way in which the new iterate x_{k+1} is computed once the quadratic model has been defined. We consider a “generic” quadratic model of the form

$$q_k(x_k + p) = f(x_k) + g(x_k)^T p + \frac{1}{2} p^T B_k p, \quad (3.4.5)$$

where B_k is a symmetric matrix that approximates the Hessian $H(x_k)$ in some sense. Unless explicitly stated otherwise, the matrix B_k is assumed to be positive definite. Precisely how this model is used will depend upon the particular method employed. Two broad classes of method are available: *line-search methods* and *trust-region methods*. All these methods are *descent* methods in the sense of (3.4.4). These methods are considered in more detail in §§3.5–3.6.

The use of an approximate Hessian is appropriate when either second derivatives are not available, and/or the minimizer of the quadratic model (3.4.3) does not exist. The minimizer of the quadratic model $q_k(x_k + p)$ satisfies

$$B_k p_k = -g(x_k). \quad (3.4.6)$$

and a positive definite B_k always ensures that p_k is well-defined.

3.4.3. The Method of Steepest-Descent

The method of steepest-descent is based on one of the simplest possible choices of B_k in the quadratic model (3.4.5). Given an approximate minimizer x_k , the method of steepest descent defines the local quadratic model

$$q_k(x_k + p) = f(x_k) + g(x_k)^T p + \frac{1}{2} p^T p, \quad (3.4.7)$$

which is the model (3.4.5) with $B_k = I$. From (3.4.6), $q_k(x_k + p)$ has a unique global minimizer p_k such that

$$p_k = -g(x_k). \quad (3.4.8)$$

It is apparent from this expression that p_k is independent of $f(x_k)$, which is to be expected since an additive constant in the objective does not alter the location of a minimizer.

This search direction is known as the *classical steepest-descent* direction. The steepest-descent direction was so named because p_k is the vector that also minimizes the ratio of $g(x_k)^T p$ and the Euclidean length of p , i.e., p_k solves

$$\min \frac{g(x_k)^T p}{\|p\|}. \quad (3.4.9)$$

The Cauchy-Schwarz inequality $|x^T y| \leq \|x\| \|y\|$ implies that

$$g(x_k)^T p \geq -\|g(x_k)\| \|p\|,$$

where equality holds for a nonzero vector p only when p is a negative multiple of $g(x_k)$. The “optimal” p is thus the *negative gradient* (3.4.8) and is the direction of (normalized) greatest reduction in f at any nonstationary point x_k . (We shall speak of “the” steepest-descent direction as the negative gradient itself, but any negative multiple of $g(x_k)$ can also be viewed as a direction of steepest descent.)

Despite its advantages of simplicity and guaranteed convergence, the steepest-descent method is an extremely *inefficient* algorithm for general unconstrained minimization. Broadly speaking, the steepest-descent method is inadequate because the model (3.4.7) ignores the curvature of f , which is the crucial element in locating a minimum efficiently. Even when α_k is chosen as the exact step to a minimizer of f along p_k , the steepest-descent method converges linearly. Furthermore, the error constant (the reduction in the error at each step) depends on the ratio of the largest and smallest eigenvalues of the Hessian matrix at x^* , and can be arbitrarily close to unity.

Suppose that the steepest-descent method is used to minimize a quadratic—the simplest function with a bounded minimizer (see §3.2.4):

$$q(x) = c^T x + \frac{1}{2} x^T H x, \quad (3.4.10)$$

where H is symmetric and positive definite. Since $q(x)$ is quadratic, there exists a closed-form expression for the univariate minimizer of the function $q(x_k + \alpha p_k)$. An α_k defined as this minimizer will produce a much larger reduction in q than would be obtained from a backtracking line search.

Let $\lambda_{\max}$ and $\lambda_{\min}$ denote the largest and smallest eigenvalues of H . As $k \rightarrow \infty$, the steepest-descent iterates satisfy

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \leq \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}}, \quad (3.4.11)$$

where equality can be achieved. Equivalently, we may write

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \leq \frac{\text{cond}(H) - 1}{\text{cond}(H) + 1}, \quad (3.4.12)$$

where “cond” refers to the two-norm ($\text{cond}(H) = \lambda_{\max}/\lambda_{\min}$). Relation (3.4.12) shows that only a painfully small improvement may be produced by a steepest-descent iteration, even for a quadratic function whose Hessian is not particularly ill-conditioned.

To illustrate this behavior, we apply the steepest-descent method to the simple three-variable quadratic function $q(x) = x^T H x$, with

$$H = \begin{pmatrix} 1000 & 50 & 2 \\ 50 & 10 & 2 \\ 2 & 2 & 1 \end{pmatrix}, \quad (3.4.13)$$

and $\text{cond}(H) = 2082$. (All numbers in this example are rounded to four digits). Since H is positive definite, $q(x)$ is minimized for $x^* = (0, 0, 0)^T$, where $q(x^*) = 0$. Starting with $x_0 = (1, 1)^T$, for which $q(x_0) = 1119$, the first steepest-descent iteration gives

$$p_0 = -g_0 = \begin{pmatrix} -1052 \\ -62 \\ -5 \end{pmatrix}, \quad \alpha_0 = 9.976 \times 10^{-4}, \quad x_1 = \begin{pmatrix} -.04944 \\ .9382 \\ .9950 \end{pmatrix},$$

with $q(x_1) = 11.14$. After 19 further iterations, the result is

$$x_{20} = \begin{pmatrix} 5.121 \times 10^{-3} \\ -8.025 \times 10^{-2} \\ 2.967 \times 10^{-1} \end{pmatrix},$$

with $q(x_{20}) = 4.840 \times 10^{-2}$. Even after 100 iterations, q has been reduced only to 3.250×10^{-4} !

Despite the poor performance of the steepest-descent method as a general-purpose algorithm, it is widely used as a “fallback”, in the sense that a direction related to the steepest-descent direction may be used when more sophisticated methods fail.

3.4.4. Newton's Method

Our discussion of the steepest-descent method shows that a good model of curvature is the key to success in choosing a search direction. To develop such a model, we turn to the Taylor-series expansion of f about x_k :

$$f(x_k + p) = f(x_k) + g(x_k)^T p + \frac{1}{2} p^T H_k p + O(\|p\|^3), \quad (3.4.14)$$

where $g(x_k) = \nabla f(x_k)$ and $H_k = \nabla^2 f(x_k)$. When $\|p\|$ is sufficiently small, higher-order terms may be ignored, and we obtain a “natural” quadratic model $q(x_k + p)$ that approximates $f(x_k + p)$:

$$q(x_k + p) = f(x_k) + g(x_k)^T p + \frac{1}{2} p^T H_k p. \quad (3.4.15)$$

(The model function is written in terms of p to reflect its role in defining the search direction.)

If the Hessian of f at x_k is positive definite, the discussion in §3.2.4 shows that the quadratic model (3.4.15) has a unique minimizer, which we denote by p_k^N and call the *Newton direction* or *Newton step*. Using (3.2.7), p_k^N satisfies the following *Newton equations*

$$H_k p_k^N = -g(x_k). \quad (3.4.16)$$

(Since the elements of the inverse of H_k are never needed in practice, these equations will deliberately *not* be written in the mathematically equivalent form $p_k^N = -H_k^{-1}g(x_k)$.)

The Newton step will be a “good” approximation to the step to the minimizer of f when the quadratic model (3.4.15) closely approximates the local behavior of the nonlinear function f —for example, when $\|p\|$ is “small”. Remarkably, the “ideal” properties of Newton’s method carry over without significant modification to the n -dimensional case.

3.4.5. Local Convergence

We first mention selected properties of a “pure” Newton method for unconstrained minimization in which the iterates are given by

$$x_{k+1} = x_k + p_k^N, \quad (3.4.17)$$

where p_k^N is defined by (3.4.16). For a general nonlinear function f with smooth second derivatives, the pure Newton method (3.4.17) converges *quadratically* to an unconstrained local minimizer x^* if the initial iterate x_0 is sufficiently close to x^* and $H(x^*)$ is positive definite.

Theorem 3.4.1. *(Convergence of a pure Newton method). Let $f(x)$ be three times continuously differentiable, and assume that there exists $x^* \in \mathbb{R}^n$ such that $g(x^*) = 0$ and $H(x^*)$ is positive definite. Then there exists a neighborhood $N(x^*)$ such that for $x_0 \in N(x^*)$, the Newton iterates defined by (3.4.16) and (3.4.17) are well defined, remain in $N(x^*)$ and converge quadratically to x^* .*

For example, consider the function

$$f(x_1, x_2) = 10(x_1^2 - x_2)^2 + (x_1 - 1)^2,$$

whose (unique) minimizer is $x^* = (1, 1)^T$, with $f(x^*) = 0$. Starting the pure Newton method (3.4.17) at $x_0 = (1.1, 0.9)^T$, the values of $\{\|x_k - x^*\|_2\}$ and $f(x_k)$, $k = 0, \dots, 4$ (rounded to three figures).

Iteration	$\ x_k - x^*\ $	$f(x_k)$
0	.141	.971
1	.199	7.42×10^{-3}
2	6.70×10^{-3}	5.42×10^{-4}
3	9.49×10^{-5}	1.80×10^{-9}
4	1.66×10^{-9}	3.25×10^{-17}

Table 3.1: Pure Newton iterates.

3.5. Line Search Methods

The purpose of a line search is to ensure that the sequence $\{x_k\}$ is an *improving* of approximate minimizers, where an improvement is defined as the calculation of a lower

value of the objective function. An intuitively appealing strategy for choosing α_k is that $f(x_k + \alpha p_k)$ should be decreased “as much as possible” during the current iteration. If p_k is a descent direction, then $g(x_k)^T p_k < 0$, and $f(x_k + \alpha p_k)$ is a *decreasing* function at the origin. This suggests choosing α_k as a *positive minimizer* of the univariate function $f(x_k + \alpha p_k)$; i.e., α_k is chosen as a solution of the problem:

$$\underset{\alpha > 0}{\text{minimize}} \quad f(x_k + \alpha p_k).$$

This calculation is known as an *exact line search*. The resulting step α_k must satisfy

$$0 = \left. \frac{d}{d\alpha} f(x_k + \alpha p_k) \right|_{\alpha=\alpha_k} = g(x_k + \alpha_k p_k)^T p_k = g(x_{k+1})^T p_k, \quad (3.5.1)$$

which implies that the gradient is orthogonal to the search direction at x_{k+1} .

If f is a quadratic with positive-definite Hessian H , the univariate minimizer is easily calculated in closed form as

$$\alpha_k = -\frac{g(x_k)^T p_k}{p_k^T H p_k}. \quad (3.5.2)$$

If f is a general nonlinear function, under mild conditions on f , a univariate minimizer of $f(x_k + \alpha p_k)$ can be determined using the techniques of §3.3. In the general case, however, an exact linear search is not to be recommended. First, the univariate minimization method can require a substantial number of evaluations of f in order to calculate an accurate minimizer. Moreover, no guarantee exists for a *general function* that the computed minimum will satisfy $f(x_k + \alpha p_k) < f(x_k)$. (This can be guaranteed only in certain special cases, such as when $H(x)$ is uniformly positive definite.) In order to guarantee an improvement in f , it would be necessary to compute the minimizer *nearest* to $\alpha = 0$. This calculation is as difficult as finding a global minimizer of $f(x_k + \alpha p_k)$.

Although these difficulties preclude the use of an exact line search in practice, exact searches often pop up the analysis of algorithms, especially with reference to the properties of quadratic functions. A desirable feature of *practical* step length criteria is that there should be a *range* of suitable values for α , some of which are “better” in the sense of producing a larger reduction in f . However, an (inevitable?) tradeoff exists between the effort needed to find a “good” α and the overall efficiency of the descent method, which is performing a (possibly large) number of iterations. Choosing α_k as a good approximation to the nearest positive minimizer of f would give a nearly optimal reduction in the objective function at every iteration, but would tend to require an inordinately large number of evaluations of f and/or g during the entire course of the minimization. Ideally, step length criteria should suggest how to find a “good” α with little effort.

3.5.1. The backtracking line search

The requirement that $f(x_{k+1}) < f(x_k)$ and the condition $g(x_k)^T p_k < 0$ are *not* sufficient to ensure that the sequence $\{x_k\}$ converges to a *minimizer* of f . For example, consider finding a local minimizer of the function $f(x) = \frac{1}{2}x^2$ with $\alpha_k = 1/2^{k+1}$, starting at the point $x_0 = 2$. If a local quadratic model based on exact second-derivatives is used to define p_k (see (3.4.6)), we obtain $p_k = -x_k$ and $f(x_{k+1}) < f(x_k)$ for all k . This “direction” generates the sequence of iterates 2, 1, $\frac{3}{4}$, $\frac{27}{32}$, which converges to $\bar{x} \approx .5776$, which is not a minimizer of $\frac{1}{2}x^2$.

In this example, although f is decreased at each step, α_k eventually becomes so small that the decrease in f converges to zero before the gradient has time to converge to zero. This difficulty can be overcome if, at each step, the step length is chosen to produce a *sufficient decrease* in f . Broadly speaking, a step of sufficient decrease is one that guarantees a reduction in f that is proportional to the magnitude of the directional derivative $|g(x_k)^T p_k|$. This idea is best illustrated in the case of the method of classical steepest descent, where $|g(x_k)^T p_k| = \|g(x_k)\|^2$. In this case, it is clear that a step of sufficient decrease will ensure that a small decrease in f must correspond to a small value of $\|g(x_k)\|$.

A sufficient decrease in f can be specified by considering the properties of the local quadratic model $q_k(x_k + p)$ (3.4.5). As q_k is chosen such that $q_k(x_k) = f(x_k)$, it can be seen that if $q_k(x_k + \alpha p_k) \approx f(x_k + \alpha p_k)$, then

$$f(x_k) - f(x_k + \alpha p_k) \approx q_k(x_k) - q_k(x_k + \alpha p_k).$$

If this is not the case, then α can be considered to be too large. The quantity $q_k(x_k) - q_k(x_k + \alpha p_k)$ can be regarded as the change in f predicted by the quadratic model. The idea is to choose α small enough that f is reduced by an amount that agrees with the reduction $q_k(x_k) - q_k(x_k + \alpha p_k)$ predicted by the quadratic model. These statements are justified by the limiting relationship

$$\lim_{\alpha \rightarrow 0} \frac{f(x_k) - f(x_k + \alpha p_k)}{q_k(x_k) - q_k(x_k + \alpha p_k)} = 1, \quad (3.5.3)$$

which implies that the predicted and actual reductions in f become identical as α is reduced.

If a quadratic model is to be used to compare actual and predicted reductions in f it is necessary that the predicted reduction be positive. This will be the case if B_k is positive definite and $0 < \alpha < 2$. The reduction in the model can be written as

$$q_k(x_k) - q_k(x_k + \alpha p_k) = -\alpha g(x_k)^T p_k - \frac{1}{2} \alpha^2 p_k^T B_k p_k. \quad (3.5.4)$$

On substituting $B_k p_k = -g(x_k)$, we obtain

$$q_k(x_k) - q_k(x_k + \alpha p_k) = -\alpha g(x_k)^T p_k + \frac{1}{2} \alpha^2 g(x_k)^T p_k = -\alpha g(x_k)^T p_k (1 - \frac{1}{2} \alpha),$$

which implies that $q_k(x_k) - q_k(x_k + \alpha p_k) > 0$ as long as B_k is positive definite and $0 < \alpha < 2$.

The definition of a sufficient decrease in f involves a preassigned scalar parameter μ such that $0 < \mu < \frac{1}{2}$ (the reason for the upper bound of $\frac{1}{2}$ will be discussed later). A step α_k is considered to be an acceptable step length if the actual reduction in f is no worse than μ times the predicted reduction in f ; i.e., if

$$\frac{f(x_k) - f(x_k + \alpha_k p_k)}{q_k(x_k) - q_k(x_k + \alpha_k p_k)} \geq \mu \quad (3.5.5)$$

(c.f. (3.5.3)). Given an initial estimate α_k^0 of α_k (usually $\alpha_k^0 = 1$), if (3.5.5) is satisfied, then $\alpha_k = \alpha_k^0$. Otherwise, α_k^0 is reduced by multiplying it with “contraction” factor γ_c such that $\gamma_c < 1$, and the test is repeated. Hence, the step α_k is the first member of the sequence $\{\gamma_c^{j-1} \alpha_k^0\}_{j=1}^{\infty}$ such that the condition (3.5.5) is satisfied. This procedure, known as a *backtracking line search*, is the basis of the generic line-search algorithm given in Algorithm 3.5.1.

Algorithm 3.5.1. Generic line-search method.
 Specify constants μ and γ_c such that $0 < \mu < \frac{1}{2}$ and $0 < \gamma_c < 1$;
 Choose x_0 and B_0 ; Evaluate $f(x_0)$, $g(x_0)$;
while not converged do
 Solve $B_k p_k = -g(x_k)$;
 $\alpha_k \leftarrow 1$; $j_k \leftarrow 1$;
 while $f(x_k) - f(x_k + \alpha_k p_k) < \mu(q_k(x_k) - q_k(x_k + \alpha_k p_k))$ **do**
 $\alpha_k \leftarrow \alpha_k \gamma_c$; $j_k \leftarrow j_k + 1$;
 end while
 $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
 Evaluate $g(x_{k+1})$; Define B_{k+1} ;
 $k \leftarrow k + 1$;
end while

Using (3.5.4) and the definition of p_k , this condition may be written as

$$f(x_k + \alpha_k p_k) \leq f(x_k) + \mu \left(\alpha_k g(x_k)^T p_k + \frac{1}{2} \alpha_k^2 p_k^T B_k p_k \right). \quad (3.5.6)$$

If we use the univariate function $M(\alpha) \triangleq f(x_k + \alpha p_k)$ to describe the behavior of f along p_k , then the condition (3.5.6) is equivalent to requiring that $M(\alpha_k) \leq Q(\alpha_k)$, where $Q(\alpha)$ is the quadratic $Q(\alpha) \triangleq f(x_k) + \mu \left(\alpha g(x_k)^T p_k + \frac{1}{2} \alpha^2 p_k^T B_k p_k \right)$. Since B_k is positive definite, the quadratic term of Q is positive, and Q attains its minimum value at the unique stationary point $\alpha^* = -g(x_k)^T p_k / p_k^T B_k p_k = 1$. It follows that $Q(\alpha)$ is minimized by the “pure” Newton step $\alpha = 1$. In general, the sufficient decrease condition $M(\alpha) \leq Q(\alpha)$ may or may not hold at $\alpha = 1$.

It is necessary to show that the backtracking loop of Algorithm 3.5.1 will always terminate in a finite number of steps. Consider the univariate function $\phi(\alpha) \equiv M(\alpha) - Q(\alpha)$. From the definition of $M(\alpha)$ and $Q(\alpha)$, we have $\phi(0) = M(0) - Q(0) = 0$, and $\phi'(0) = M'(0) - \mu M'(0) = M'(0)(1 - \mu) < 0$. It follows from the The Nonzero Derivative Theorem that there exist a nontrivial interval $(0, \sigma)$ such that $\phi(\alpha) < 0$ for all $\alpha \in (0, \sigma)$. Hence, for α sufficiently small, the backtracking line search will give an α in the interval $(0, \sigma)$ such that the condition (3.5.6) will be satisfied.

3.5.2. The Goldstein-Armijo line search

It is not necessary to use the same model to define both p_k and the sufficient decrease criterion (3.5.5). In particular, we may define a sufficient decrease criterion

$$\frac{f(x_k) - f(x_k + \alpha_k p_k)}{\psi(x_k) - \psi(x_k + \alpha_k p_k)} \geq \mu, \quad (3.5.7)$$

in terms of a general function ψ known as the *line-search model function*. The only requirements on ψ are that $\psi_k(x_k) = f(x_k)$ and $\nabla \psi_k(x_k) = g(x_k)$.

Consider $\psi_k(x) \equiv \ell_k(x)$, where ℓ_k is the *affine model* $\ell_k(x_k + p) = f(x_k) + g(x_k)^T p$. The reduction in f predicted by ℓ_k is

$$\ell_k(x_k) - \ell_k(x_k + \alpha_k p_k) = -\alpha_k g(x_k)^T p_k,$$

and the general criterion (3.5.7) simplifies to become

$$f(x_k + \alpha_k p_k) \leq f(x_k) + \mu \alpha_k g(x_k)^T p_k. \quad (3.5.8)$$

This criterion is widely used in optimization and is known as the *Armijo condition*.

The Armijo condition is preferred because it is simpler than (3.5.6) yet forces a more substantial reduction in f . This follows from the inequality

$$\mu \alpha_k g(x_k)^T p_k \leq \mu \alpha_k g(x_k)^T p_k + \frac{1}{2} \alpha_k^2 p_k^T B_k p_k,$$

which implies that if α_k satisfies (3.5.8), then it must also satisfy (3.5.6).

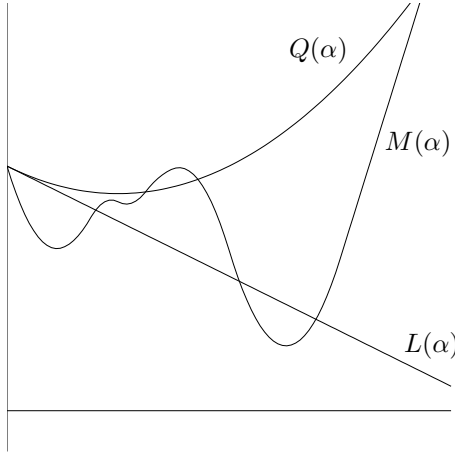


Figure 3.2: This graph depicts a typical $M(\alpha) = f(x_k + \alpha p_k)$ as a function of positive α . The linear and quadratic functions $L(\alpha)$ and $Q(\alpha)$ are superimposed on the graph. Note that any point that lies on the L graph must always lie below the Q graph.

3.5.3. Modified Newton Methods

secUC-indefinite

In discussing Newton-based methods, we have assumed thus far that $H(x)$ is positive definite. Since this condition does not hold for general functions (even those that are bounded below), a strategy must be devised for detecting indefiniteness in the Hessian and defining a suitable Newton-based search direction.

When H_k is not positive definite, the quadratic model (3.4.15) no longer provides clearcut guidance for designing an algorithm, and in fact suggests that an “infinite” step should be taken. Without positive-definiteness, there is neither a “natural” step length nor an “obvious” interpretation of the quadratic model. This lack of certainty means that researchers have had free rein to formulate numerous methods based on differing interpretations of the quadratic model.

One popular strategy is to use the quadratic model $q_k(x_k + p) = f(x_k) + g(x_k)^T p + \frac{1}{2} p^T B_k p$ with B_k equal to $H(x_k)$ or some related positive-definite matrix. In particular, $B_k = H(x_k)$ only when $H(x_k)$ is sufficiently positive definite. With this strategy, B_k is always positive definite and p_k is calculated from the linear system $B_k p_k = -g(x_k)$. (Since this section concerns the logic of a single iteration, we henceforth drop the subscript k .) By requiring B_k to be positive definite, we guarantee that p_k will be a descent direction when $g(x_k) \neq 0$.

3.5.4. The spectral decomposition

Careful analysis is essential in developing a satisfactory definition of the model function Hessian B . (Several “obvious” algorithms have turned out to be ill-defined and/or numerically unstable.) A fundamental tool is a computational procedure for determining whether H is actually positive definite. If not, it makes practical sense to compute B using the same procedure that checks on positive-definiteness.

In some sense, an “ideal” B would be based on knowledge of the eigenvalues of the true Hessian H . Suppose that the complete eigensystem of H is available, i.e.,

$$H = V \Lambda V^T = \sum_{i=1}^n \lambda_i v_i v_i^T, \quad (3.5.9)$$

where the orthonormal columns $\{v_i\}$, $i = 1, \dots, n$, of V are the eigenvectors of H and the diagonal matrix Λ contains the eigenvalues $\{\lambda_i\}$ ordered so that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$. If H is positive definite, its eigenvalues are positive and the spectral decomposition of its inverse is given by $H^{-1} = V \Lambda^{-1} V^T = \sum_{i=1}^n \lambda_i^{-1} v_i v_i^T$. This implies that the Newton direction can be conveniently written as

$$p^N = -H^{-1}g = -V \Lambda^{-1} V^T g = -\sum_{i=1}^n \lambda_i^{-1} v_i v_i^T g. \quad (3.5.10)$$

In our “ideal” method the modified Hessian B is taken as

$$B = V \bar{\Lambda} V^T, \quad (3.5.11)$$

so that B and H have the same eigenvectors, but (possibly) different eigenvalues.

A positive value δ is specified as a lower bound on the smallest eigenvalue of B . If $\lambda_i \geq \delta$, λ_i is considered *sufficiently positive*. If $|\lambda_i| < \delta$, λ_i is said to be *negligible*. Let $\mathcal{I}_+$ denote the subset of indices of sufficiently positive eigenvalues, and $\mathcal{V}_+$ the subspace spanned by the eigenvectors corresponding to these eigenvalues. Let $\mathcal{I}_0$ denote the subset of indices of negligible eigenvalues and $\mathcal{I}_-$ the subset of indices of eigenvalues satisfying $\lambda_i \leq -\delta$, with similar meanings for $\mathcal{V}_0$ and $\mathcal{V}_-$.

Since the positive eigenvalues of H define a “well behaved” quadratic model, a “sensible” approach is to leave these eigenvalues unaltered in defining $\bar{\Lambda}$, and to choose

$$\bar{\lambda}_i = \lambda_i, \quad i \in \mathcal{I}_+. \quad (3.5.12)$$

The question remains as to how B should affect $\mathcal{V}_0$ and $\mathcal{V}_-$.

We mention two possible strategies. First, B can be chosen as the “nearest” matrix to H that is sufficiently positive definite. This implies that we take

$$\bar{\lambda}_i = \delta, \quad i \notin \mathcal{I}_+. \quad (3.5.13)$$

The effect of this modification is to add a positive semidefinite matrix E to H , with $B = H + E$. The size of the modification is given by

$$\|E\|_2 = \|V(\bar{A} - A)V^T\|_2 = \max_{1 \leq i \leq n} |\bar{\lambda}_i - \lambda_i| = \max\{0, \delta - \lambda_n\},$$

where λ_n is the least eigenvalue of H . Note that $E = 0$ if all the eigenvalues of H are sufficiently positive.

An alternative strategy is to “reverse” the effect of H on the subspace associated with “sufficiently negative” eigenvalues, giving

$$\bar{\lambda}_i = \delta, \quad i \in \mathcal{I}_0; \quad \bar{\lambda}_i = -\lambda_i, \quad i \in \mathcal{I}_-. \quad (3.5.14)$$

In this case, the modification matrix E such that $B = H + E$ has norm

$$\|E\|_2 = \max\{0, \delta - \lambda_n, |\lambda_n| - \lambda_n\}.$$

Again, $E = 0$ if all the eigenvalues of H are sufficiently positive (i.e., $\lambda_i \geq \delta$).

Once the modified eigenvalues are defined, the modified Newton direction is given by (3.5.11), i.e.,

$$p = -V\bar{A}^{-1}V^Tg = -\sum_{i=1}^n \bar{\lambda}_i^{-1} v_i v_i^T g, \quad (3.5.15)$$

which is a linear combination of the eigenvectors of B (cf. (3.5.10)).

To illustrate the effect of these modifications, consider a problem with Hessian and gradient given by

$$H = \begin{pmatrix} 1 & 0 \\ 0 & -2 \end{pmatrix}, \quad \text{and} \quad g = \begin{pmatrix} 2 \\ 4 \end{pmatrix}. \quad (3.5.16)$$

The Newton direction defined by the solution of $Hp^N = -g$ is $p^N = (-2, 2)^T$, which corresponds to a saddle point of the quadratic model (3.4.15). In this case, p^N is *not* a descent direction since $g^T p^N = 4$.

Following the first strategy, B is taken as the positive-definite matrix “closest” to H , where a positive value δ is chosen as a lower bound that defines what we mean by “positive”. Thus, if $\lambda_i < \delta$, we set $\bar{\lambda}_i = \delta$ in (3.5.9). It follows that

$$B = \begin{pmatrix} 1 & 0 \\ 0 & \delta \end{pmatrix},$$

which gives $p = (-2, -4/\delta)^T$. Replacing negative eigenvalues by δ in (3.5.9) means that p is dominated by vectors in $\mathcal{V}_-$ and $\mathcal{V}_0$, and $\|p\|$ will usually be large.

The second approach reverses the portion of H in $\mathcal{V}_-$, leaving the influence of $\mathcal{V}_+$ unchanged. This gives

$$B = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix},$$

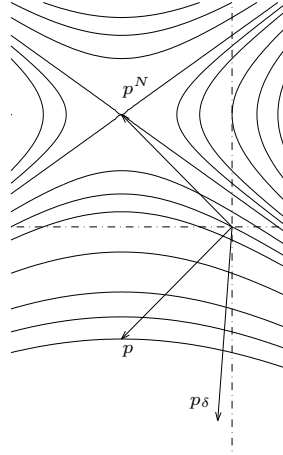


Figure 3.3: Level curves of the quadratic model $q = 2p_1 + 4p_2 + \frac{1}{2}p_1^2 - p_2^2$ defined by H and g (3.5.16). The model has a saddle point at $p = (-2, 2)$. The unmodified Newton direction $p^N = p$ steps uphill to the saddle point. The modified Newton direction p is formed by reversing the negative eigenvalues in the spectral decomposition. The modified Newton direction p_δ is obtained by replacing negative eigenvalues by δ . The value $\delta = 0.1$ was used to compute p_δ . A more realistic value of δ would make p_δ considerably more parallel to the p_2 axis.

with $p = (-2, -2)^T$. This p is also a descent direction, with $g^T p = -12$. In this case, the modified direction has the same norm as the unmodified Newton direction, and hence may be considered to be a better direction than $p = (-2, -4/\delta)^T$.

The Newton direction and its two alternative modifications are depicted in Fig. 3.3. The value $\delta = 0.1$ was used to compute the direction $p = (-2, -4/\delta)^T$. A more realistic value of δ would make p_δ considerably more parallel to the x_2 axis (which is itself parallel to the eigenvector associated with the least eigenvalue of H).

In summary, we have the following modified Newton method for minimizing f .

Algorithm 3.5.2. Newton's method with backtracking.

Specify μ such that $0 < \mu < \frac{1}{2}$ and δ such that $\delta \ll 1$;

Let x_0 be given;

$k \leftarrow 0$;

do while not converged

 Evaluate $g(x_k)$, $H(x_k)$;

 Compute V_k and D_k such that $H(x_k) = V_k D_k V_k^T$;

 Set $\bar{\lambda}_i = \max\{\delta, |\lambda_i|\}$;

 Define $\bar{D}_k = \text{diag}(\bar{\lambda}_i)$;

 Set $p_k = -V_k \bar{D}_k^{-1} V_k^T g(x_k)$;

 Set $\alpha_k = 1$; $j_k = 1$;

do while $f(x_k + \alpha_k p_k) > f(x_k) + \alpha_k \mu g(x_k)^T p_k$;

$\alpha_k \leftarrow \alpha_k/2$; $j_k \leftarrow j_k + 1$;

end do

```

Set  $x_{k+1} = x_k + \alpha_k p_k$ ;
 $k \leftarrow k + 1$ ;
end do

```

In Table 3.2 we give the results of this modified Newton method applied to the function $f(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$, starting at the point $x_0 = (2, 2)^T$ (see Figure 3.1). Convergence to the local minimizer $x^* = (\frac{1}{2}, -1)^T$ occurred in 9 iterations. Note the quadratic convergence. In this run it was necessary to modify the Hessian at the first iteration, where the smallest eigenvalue of the Hessian was -5.5 .

k	n_f	α_k	$f(x_k)$	$\ g(x_k)\ $	$\lambda_{\min}(H(x_k))$
0	0	0.0e+00	3.3250752e+02	5.2e+02	-5.5e+00
1	1	1.0e+00	5.7733341e+01	1.4e+02	2.2e+01
2	2	1.0e+00	1.5525321e+01	4.7e+01	1.3e+01
3	3	1.0e+00	3.5681929e+00	1.5e+01	8.1e+00
4	4	1.0e+00	6.0466619e-01	4.4e+00	4.9e+00
5	5	1.0e+00	5.3774209e-02	1.0e+00	3.2e+00
6	6	1.0e+00	1.0888910e-03	1.3e-01	2.6e+00
7	7	1.0e+00	7.3270158e-07	3.2e-03	2.5e+00
8	8	1.0e+00	3.6462543e-13	2.2e-06	2.5e+00
9	9	1.0e+00	0.0000000e+00	1.1e-12	2.5e+00

Table 3.2: Modified Newton applied to $f(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$, starting at $x_0 = (2, 2)^T$, $\mu = \frac{1}{4}$, $\delta = 10^{-8}$.

3.5.5. Modified Newton as Steepest Descent

Although the steepest-descent method defined by $p_k = -g(x_k)$ usually performs poorly in practice, it is important to note that a different definition of what we mean by “steepest” can give other steepest-descent methods. One of these, to be discussed in this section, is equivalent to the modified Newton method described above.

It was shown in Section 3.4.3 that $p_k = -g(x_k)$ gives the maximum first-order reduction in f for a unit change in the variables. In our analysis, the unit change $d = p/\|p\|$ was defined so that $\|d\| = (d^T d)^{1/2}$. If we use a different norm for d , then we obtain a different algorithm. Because the method depends on the choice of norm, we call the method with $p_k = -g(x_k)$ the *classical* steepest-descent method.

If B is a symmetric positive-definite matrix, then $\|x\|_B^2 \equiv x^T B x$ defines a norm on $x \in \mathbb{R}^n$. This norm is often called an *elliptic* norm because the level curves of the function $\phi = \|x\|_B^2 \triangleq x^T B x$ are ellipsoids.

For clarity, we omit the subscript k . Using the same arguments used to derive the “classical” steepest-descent method, the direction that gives the maximum reduction in f to first order is the solution to the problem

$$\underset{\substack{p \in \mathbb{R}^n \\ p \neq 0}}{\text{minimize}} \quad \frac{g^T p}{\|p\|_B}.$$

The solution of this problem is precisely the modified Newton direction $p = -B^{-1}g$. To see this we use the result that every positive-definite matrix must have a symmetric nonsingular “square root” matrix. For the positive-definite matrix B this implies that there is a nonsingular S such that $B = S^2 = SS = S^TS$. Using the Cauchy-Schwartz inequality we have

$$(g^Tp)^2 = (g^TS^{-1}Sp)^2 \leq (g^TS^{-1}S^{-T}g)(p^TS^TS p) = (g^TB^{-1}g)\|p\|_B^2.$$

This inequality implies that for all $p \neq 0$,

$$-(g^TB^{-1}g)^{\frac{1}{2}} \leq \frac{g^Tp}{\|p\|_B} \leq (g^TB^{-1}g)^{\frac{1}{2}}.$$

Clearly, any p for which the lower bound holds with equality must minimize $g^Tp/\|p\|_B$. This occurs for the p that satisfies the equations $Bp = -g$.

This analysis indicates that a modified Newton method is a steepest-descent method in which the norm used to measure what is the “steepest direction” changes at each iteration. Note that the classical steepest-descent method corresponds to the case $B_k = I$.

Theorem 3.5.1. *Let $\{B_k\}$ be a sequence of symmetric positive-definite matrices such that*

$$\lambda_{\min}(B_k) \geq \delta > 0 \quad \text{and} \quad \lambda_{\max}(B_k) \leq \Delta < +\infty,$$

where δ and Δ are fixed constants, and $\lambda_{\max}(B_k)$ and $\lambda_{\min}(B_k)$ denote the largest and smallest eigenvalues of B_k . If the search direction p_k is defined for every k by $B_k p_k = -g(x_k)$, then the directional derivative $g(x_k)^T p_k$ satisfies

$$\frac{1}{\Delta} \|g(x_k)\|^2 \leq |g(x_k)^T p_k| \leq \frac{1}{\delta} \|g(x_k)\|^2, \quad (3.5.17)$$

$$\delta \|p_k\|^2 \leq |g(x_k)^T p_k| \leq \Delta \|p_k\|^2. \quad (3.5.18)$$

Proof. These results can be proved using standard matrix-vector norm inequalities. However, we will prove them from first principles, without using the properties of matrix norms. For simplicity, we drop the suffix k .

Because $Bp = -g$ with B positive definite, it follows that

$$|g^Tp| = -g^Tp = g^TB^{-1}g.$$

Consider the spectral decomposition $B = VDV^T$, where $D = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ is the diagonal matrix of eigenvalues of B , and V is a matrix of eigenvectors such that $V^TV = VV^T = I$. If w is the vector such that $w = V^Tg$, then

$$g^TB^{-1}g = g^TV D^{-1}V^Tg = w^T D^{-1}w = \sum_{i=1}^n \frac{w_i^2}{\lambda_i}.$$

Since each eigenvalue satisfies $\lambda_i \geq \delta$, we can replace each eigenvalue by its lower bound and write

$$\sum_{i=1}^n \frac{w_i^2}{\lambda_i} \leq \sum_{i=1}^n \frac{w_i^2}{\delta} = \frac{1}{\delta} \sum_{i=1}^n w_i^2 = \frac{1}{\delta} w^T w = \frac{1}{\delta} g^TVV^Tg = \frac{1}{\delta} \|g\|^2.$$

It follows that $|g^T p| \leq (1/\delta)\|g\|^2$, which is the right-most inequality of (3.5.17).

Next we prove the right-most inequality of (3.5.18). If we define $u = V^T p$ and note that $\lambda_i \leq \Delta$, we have

$$|g^T p| = -g^T p = p^T B p \leq u^T D u = \sum_{i=1}^n u_i^2 \lambda_i \leq \Delta \sum_{i=1}^n u_i^2 = \Delta \|p\|^2.$$

The remaining inequalities follow in a similar way. ■

Corollary 3.5.1. *For any $u \in \mathbb{R}^n$, the elliptic norms $\|u\|_{B_k} = (u^T B_k u)^{1/2}$ and $\|u\|_{B_k^{-1}} = (u^T B_k^{-1} u)^{1/2}$ satisfy*

$$\begin{aligned} \gamma \|u\| &\leq \|u\|_{B_k} \leq \Gamma \|u\| \\ \frac{1}{\Gamma} \|u\| &\leq \|u\|_{B_k^{-1}} \leq \frac{1}{\gamma} \|u\|, \end{aligned}$$

where $\gamma^2 = \delta$, $\Gamma^2 = \Delta$, and $\|u\|$ denotes the Euclidean norm $\|u\| = (u^T u)^{1/2}$.

This corollary implies that in the limit, the Euclidean and elliptic norms are equivalent in the sense that the elliptic norm is always within a certain factor of the Euclidean norm, independent of k .

3.5.6. Convergence of Steepest Descent

Next we consider the conditions under which the sequence $\{g(x_k)\}$ converges to zero.

The Goldstein-Armijo line search discussed in Section 3.5.2 defines a new iterate such that

$$f(x_k) - f(x_k + \alpha_k p_k) \geq -\alpha_k \mu g(x_k)^T p_k.$$

Since, by definition, $x_{k+1} = x_k + \alpha_k p_k$ and $g(x_k)^T p_k < 0$, we have

$$f(x_k) - f(x_{k+1}) \geq -\alpha_k \mu g(x_k)^T p_k = \alpha_k \mu |g(x_k)^T p_k|, \quad (3.5.19)$$

and it follows that $f(x_k) > f(x_{k+1})$ and the sequence $\{f(x_k)\}$ decreases monotonically. If it holds that the sequence $\{f(x_k)\}$ is bounded below, then the strictly monotonic sequence $\{f(x_k)\}$ must converge to a limit. The sequence $\{f(x_k)\}$ will certainly be bounded below if f itself is bounded below (for example, $f \geq 0$ for functions of the form $f = \sum_i \phi_i(x)^2$). It is not realistic to expect that f be bounded below at every point in $\mathbb{R}^n$. Fortunately, it is possible to obtain a very convenient characterization of the region in which the iterates can be expected to lie. Since $f(x_{k+1}) < f(x_k)$ for all k , once x_0 has been defined, all subsequent iterates must have an objective value that is less than $f(x_0)$. This implies that all iterates lie in the level set $\{x \mid f(x) \leq f(x_0)\}$ associated with $f(x_0)$.

If the level set associated with a function f is *closed and bounded* in $\mathbb{R}^n$, then a standard result of analysis states that f must take on its maximum and minimum values at points in the level set. If f is defined and continuous for all $x \in \mathbb{R}^n$, then it can be shown that the level sets of f are always closed. However, without the assumption of boundedness, it is easy to give examples of level sets that may not contain a minimizer. For example, consider the univariate function $f(x) = x^3$, which is smooth but *unbounded below* because

$f(x) \rightarrow -\infty$ as $x \rightarrow -\infty$. Hence, the level set $\mathcal{L}(\gamma)$ is unbounded for *any* γ . Note that functions that are bounded below can also have unbounded level sets. For example, consider $f(x) = e^{-x}$, which satisfies $f(x) > 0$ for all x . However, the lower bound is not attained for any finite x , and thus the set of x for which $f(x) \leq \gamma$ ($\gamma > 0$) is *unbounded*. On the other hand, a function can be unbounded below and yet have a bounded level set. For example, if $f(x) = x^2 - 1/x^2$, f is unbounded below as $|x|$ approaches zero, yet the set of x for which $f(x) \leq 0$ is the closed bounded interval $[-1, 1]$. (Since functions with singularities are *not* twice-continuously differentiable, they have already been excluded from consideration.)

Theorem 3.5.2. *Consider the method of classical steepest descent with Goldstein-Armijo linesearch when applied to a continuously differentiable function $f(x)$ defined on $\mathbb{R}^n$. If x_0 is chosen so that the level set $\mathcal{L}(f(x_0))$ is bounded, then the sequence $\{x_k\}$ lies in $\mathcal{L}(f(x_0))$; and either there exists a finite l such that $g(x_l) = 0$ or $\lim_{k \rightarrow \infty} g(x_k) = 0$.*

Proof. To show that $g(x_k)$ converges to zero, we first show that the sequence of function values $\{f(x_k)\}$ converges. Since f is continuous on the bounded level set $\mathcal{L}(f(x_0))$, it follows immediately that f is bounded from below in $\mathcal{L}(f(x_0))$, and the sequence $\{f(x_k)\}$ must converge. However, even though $\{f(x_k)\}$ converges, we cannot say yet if it converges to $f(x^*)$! If $\{f(x_k)\}$ converges, it must follow that $f(x_k) - f(x_{k+1}) \rightarrow 0$, in which case (3.5.19) and $p_k = -g(x_k)$ give

$$\alpha_k \mu |g(x_k)^T p_k| = \alpha_k \mu \|g(x_k)\|^2 \rightarrow 0. \quad (3.5.20)$$

An obvious approach is to use this limit to deduce that $\|g(x_k)\|^2 \rightarrow 0$. There are two cases to consider.

Case 1. First assume that $\{\alpha_k\}$ is uniformly bounded away from zero—i.e., over all the iterations, we never do more than some fixed number of back-tracking steps. Since the step length is of the form $\alpha_k = 1/2^{j_k-1}$ ($j_k \geq 1$), every member of the sequence $\{j_k\}$ (the number of backtracking steps associated with iteration k) must be smaller some fixed m . It follows that

$$f(x_k) - f(x_{k+1}) \geq 2^{-m} \mu \|g(x_k)\|^2.$$

Since $f(x_k) - f(x_{k+1}) \rightarrow 0$ it follows that $\|g(x_k)\| \rightarrow 0$, as required.

Case 2. Suppose now that there is no uniform lower bound on α_k (for example, suppose each line search requires one more backtracking step than the last). Define the set of integers $K = \{k \mid j_k \geq 1\}$ associated with the iterations at which at least one backtracking step was necessary. Since $\{\alpha_k\}$ is not bounded away from zero, K must define an infinite subsequence of the iterations.

For every $k \in K$, define the step $\sigma_k = 2\alpha_k$, which is the “last” step to fail the sufficient reduction test. Then, by definition, $\sigma_k = (\frac{1}{2})^{j_k-1}$ and

$$f(x_k + \sigma_k p_k) \geq f(x_k) + \sigma_k \mu g(x_k)^T p_k, \quad k \in K.$$

(Note the direction of the inequality.) Adding $-\sigma_k g(x_k)^T p_k$ to both sides of this expression gives

$$f(x_k + \sigma_k p_k) - f(x_k) - \sigma_k g(x_k)^T p_k \geq -\sigma_k (1 - \mu) g(x_k)^T p_k.$$

Substituting $p_k = -g(x_k)$ in the right-hand side of this equation leads to the bound

$$f(x_k + \sigma_k p_k) - f(x_k) - \sigma_k g(x_k)^T p_k \geq \sigma_k (1 - \mu) \|g(x_k)\|^2. \quad (3.5.21)$$

The Taylor-series expansion with integral form of the remainder gives

$$f(x_k + \sigma_k p_k) - f(x_k) - \sigma_k g(x_k)^T p_k = \sigma_k \int_0^1 (g(x_k + \xi \sigma_k p_k) - g(x_k))^T p_k d\xi.$$

If this expression is used in the left-hand side of (3.5.21), we obtain

$$\begin{aligned} (1 - \mu) \|g(x_k)\|^2 &\leq \int_0^1 (g(x_k + \xi \sigma_k p_k) - g(x_k))^T p_k d\xi \\ &\leq \int_0^1 \|g(x_k + \xi \sigma_k p_k) - g(x_k)\| \|p_k\| d\xi, \end{aligned}$$

where the last inequality follows from the Cauchy-Schwartz inequality. After substituting $\|p_k\| = \|g(x_k)\|$ on the right-hand side and simplifying, we have

$$\begin{aligned} (1 - \mu) \|g(x_k)\| &\leq \int_0^1 \|g(x_k + \xi \sigma_k p_k) - g(x_k)\| d\xi \\ &\leq \max_{0 \leq \xi \leq 1} \|g(x_k + \xi \sigma_k p_k) - g(x_k)\|. \end{aligned}$$

The continuity of g implies that there exists some $\theta \in (0, 2\alpha_k)$ such that

$$\max_{0 \leq \xi \leq 1} \|g(x_k + \xi \sigma_k p_k) - g(x_k)\| = \|g(x_k + \theta p_k) - g(x_k)\|.$$

Putting all these inequalities together we obtain

$$(1 - \mu) \|g(x_k)\| \leq \|g(x_k + \theta p_k) - g(x_k)\|, \quad (3.5.22)$$

for some $\theta \in (0, 2\alpha_k)$, as required.

Since $\alpha_k \rightarrow 0$, the boundedness of $\mathcal{L}(f(x_0))$ implies that both $\|g(x_k)\|$ and $\theta \|p_k\|$ are bounded above, and it follows that $\theta p_k \rightarrow 0$. The assumed continuity of $g(x)$ implies that $g(x_k + \theta p_k) \rightarrow g(x_k)$ and so $\|g(x_k)\| \rightarrow 0$ as required. ■

The next theorem establishes the convergence of the modified Newton line search method.

Theorem 3.5.3. (*Global convergence of the modified Newton method*). Consider the modified Newton method with Goldstein-Armijo linesearch when applied to a twice continuously differentiable function $f(x)$ defined on $\mathbb{R}^n$. If x_0 is chosen so that the level set $\mathcal{L}(f(x_0))$ is bounded, then the sequence $\{x_k\}$ lies in $\mathcal{L}(f(x_0))$; and either there exists a finite l such that $g(x_l) = 0$ or $\lim_{k \rightarrow \infty} g(x_k) = 0$.

Proof. The Goldstein-Armijo line search defines a new iterate such that

$$f(x_k) - f(x_k + \alpha_k p_k) \geq -\alpha_k \mu g(x_k)^T p_k.$$

Since, by definition, $x_{k+1} = x_k + \alpha_k p_k$ and $g(x_k)^T p_k < 0$, we have

$$f(x_k) - f(x_{k+1}) \geq -\alpha_k \mu g(x_k)^T p_k = \alpha_k \mu \|p_k\|_{B_k}^2 = \alpha_k \mu \|g(x_k)\|_{B_k^{-1}}^2,$$

where $\|g\|_{B_k^{-1}}$ denotes the elliptic norm $(g^T B_k^{-1} g)^{1/2}$. It follows that the objective values satisfy $f(x_k) > f(x_{k+1})$ and the sequence $\{f(x_k)\}$ decreases monotonically, forcing $\alpha_k \mu \|g(x_k)\|_{B_k^{-1}}^2 \rightarrow 0$.

From this point, the analysis is identical to that of Theorem 3.5.2 except that we use the elliptic norm $\|g(x_k)\|_{B_k^{-1}}$ for $g(x_k)$ and the elliptic norm $\|p_k\|_{B_k}$ for p_k . The expression analogous to (3.5.22) is

$$(1 - \mu) \|g(x_k)\|_{B_k^{-1}} \leq \|g(x_k + \theta p_k) - g(x_k)\|_{B_k^{-1}}, \quad (3.5.23)$$

for some $\theta \in (0, 2\alpha_k)$.

The boundedness of $\mathcal{L}(f(x_0))$ implies that the components of $H(x)$ are bounded above and below in $\mathcal{L}(f(x_0))$. This implies that the eigenvalues must also be bounded in absolute value, and that the modification process must give a matrix B_k satisfying $\lambda_{\max}(B_k) \leq \Delta < +\infty$. Similarly, the modification of the spectral decomposition of $H(x_k)$ ensures that the eigenvalues of B_k are never less than δ , and $\lambda_{\min}(B_k) \geq \delta > 0$. It follows from the Corollary to Theorem 3.5.1 that $\|p_k\|$ is bounded above and $\theta p_k \rightarrow 0$ in (3.5.23) above. The assumed continuity of $g(x)$ implies that $g(x_k + \theta p_k) \rightarrow g(x_k)$ and so $\|g(x_k)\| \rightarrow 0$ as required. ■

Equation (3.5.18) implies that for all k ,

$$|g(x_k)^T p_k| \geq \rho \|p_k\| \|g(x_k)\|, \quad (3.5.24)$$

where $\rho = \delta/\Delta$. Because $|x^T y| = \cos \theta \|x\| \|y\|$, for any two vectors x and y , the inequality (3.5.24) implies that the angle between unit vectors in the direction of p_k and $-g(x_k)$ is bounded away from $\pi/2$. Search directions that satisfy condition (3.5.24) are said to be *gradient related*. Equation (3.5.24) also demonstrates the relationship between the magnitude of δ and the size of the condition number of the matrix B_k . The larger the condition number of B_k (i.e., the more “elongated” the level curves of $q_k(x)$), the closer p_k and $g(x_k)$ can come to being orthogonal.

To summarize: the global convergence results concern algorithms that employ a direction of sufficient descent in conjunction with a step length of sufficient decrease. If such an algorithm is applied to a general smooth function, starting at a point x_0 for which the level set $\mathcal{L}(f(x_0))$ is closed and bounded, then the iterates $x_{k+1} = x_k + \alpha_k p_k$ satisfy

$$\lim_{k \rightarrow \infty} g(x_k) = 0.$$

Obviously, no stronger result (such as convergence to a *minimizer* of f) can be proved without utilizing curvature information, since we know from Section 3.2 that a minimizer is distinguished from other stationary points by (at the least) a positive semidefinite Hessian. Nonetheless, in practice the strategy of reducing f at every iteration often means that descent methods converge to minimizers.

3.5.7. Choosing the Line Search Parameter

Until now, the only requirement on the line search parameter is that, at each iteration, it expresses the reduction in the objective function as a fraction of the reduction in the line search model function, i.e., μ must lie in the interval $(0, 1)$. However, there is a danger that if μ is too large (i.e., too close to 1), then the line search may be too restrictive. For example, consider Table 3.3, which gives results for the modified Newton method applied to the function $f(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$, starting at the point $x_0 = (2, 2)^T$. The only difference between this run and that of Table 3.2 is that the line search parameter μ has the value $\frac{2}{3}$ instead of $\frac{1}{4}$. The effect of this change is considerable, with 31 iterations being required instead of 9. The difference is due to the fact that as the solution is approached, the step $\alpha_k = 1$ is not accepted by the Armijo test and a backtracking step is made. This implies that errors in the iterates is no longer related by the expression $\|x_{k+1} - x^*\| \leq \gamma \|x_k - x^*\|^2$, and the usual quadratic termination is confounded.

k	n_f	α_k	$f(x_k)$	$\ g(x_k)\ $
0	0	0.0e+00	3.3250e+02	5.2e+02
1	1	1.0e+00	5.7733e+01	1.4e+02
2	2	1.0e+00	1.5525e+01	4.7e+01
3	3	1.0e+00	3.5681e+00	1.5e+01
4	4	1.0e+00	6.0466e-01	4.4e+00
5	6	5.0e-01	2.3959e-01	2.5e+00
6	8	5.0e-01	8.6413e-02	1.3e+00
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
14	24	5.0e-01	2.6028e-06	6.5e-03
15	26	5.0e-01	6.5215e-07	3.2e-03
16	28	5.0e-01	1.6321e-07	1.6e-03
17	30	5.0e-01	4.0827e-08	8.1e-04
18	32	5.0e-01	1.0209e-08	4.0e-04
19	34	5.0e-01	2.5527e-09	2.0e-04
20	36	5.0e-01	6.3823e-10	1.0e-04
21	38	5.0e-01	1.5956e-10	5.1e-05
22	40	5.0e-01	3.9891e-11	2.5e-05
23	42	5.0e-01	9.9730e-12	1.2e-05
24	44	5.0e-01	2.4930e-12	6.3e-06
25	46	5.0e-01	6.2308e-13	3.2e-06
26	48	5.0e-01	1.5595e-13	1.6e-06
27	50	5.0e-01	3.8805e-14	7.9e-07
28	52	5.0e-01	9.5183e-15	4.0e-07
29	54	5.0e-01	2.1965e-15	2.0e-07
30	56	5.0e-01	7.3217e-16	9.9e-08
31	57	1.0e+00	0.0000e+00	1.4e-15

Table 3.3: Modified Newton applied to $f(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$, starting at $x_0 = (2, 2)^T$, $\mu = 2/3$.

The next result shows that the “natural” step of one will eventually be accepted by

the Armijo test as long as μ is chosen so that $0 < \mu < \frac{1}{2}$.

Theorem 3.5.4. (*Satisfaction of sufficient decrease by unit step*). Assume that the assumptions of Theorem 3.5.3 hold. Moreover, assume that the scalar μ is chosen so that $0 < \mu < \frac{1}{2}$. Then there exists an index k_{bar} such that for all $k \geq k_{\text{bar}}$ for which $g(x_k) \neq 0$, the step $\alpha_k = 1$ satisfies the Goldstein-Armijo criterion

$$f(x_k) - f(x_k + \alpha p_k) \geq -\mu \alpha g(x_k)^T p_k.$$

Proof. First, observe that (3.5.18) and the definition $H(x_k)p_k = -g(x_k)$ imply that

$$\delta \|p_k\|^2 \leq -g(x_k)^T p_k. \quad (3.5.25)$$

Since we know from Theorem 3.5.3 that $g(x_k) \rightarrow 0$, (3.5.25) implies that $p_k \rightarrow 0$. This condition and the descent property of p_k then imply that for all sufficiently large k , $x_k + p_k \in \mathcal{L}(f(x_0))$. We henceforth consider only such values of k .

The condition to be verified is that

$$f(x_k + p_k) - f(x_k) - \mu g(x_k)^T p_k \leq 0. \quad (3.5.26)$$

Using the identity

$$f(x + p) = f(x) + g(x)^T p + \frac{1}{2} p^T H(x) p + \int_0^1 p^T (H(x + \xi p) - H(x)) p (1 - \xi) d\xi,$$

and (3.4.16), the left-hand side of (3.5.26) may be written as

$$\begin{aligned} f(x_k + p_k) - f(x_k) - \mu g(x_k)^T p_k &= \frac{1}{2} (1 - 2\mu) g(x_k)^T p_k \\ &\quad + \int_0^1 p_k^T (H(x_k + \xi p_k) - H(x_k)) p_k (1 - \xi) d\xi. \end{aligned}$$

Since H is continuous in the closed bounded region $\mathcal{L}(f(x_0))$, the quantity ω_k such that

$$\omega_k = \max_{0 \leq \xi \leq 1} \|H(x_k + \xi p_k) - H(x_k)\|$$

is bounded. Applying this definition of ω_k and (3.5.25), gives

$$\begin{aligned} f(x_k + p_k) - f(x_k) - \mu g(x_k)^T p_k &\leq \frac{1}{2} (1 - 2\mu) g(x_k)^T p_k + \frac{1}{2} \omega_k \|p_k\|^2 \\ &\leq \frac{1}{2} (1 - 2\mu - \frac{\omega_k}{\delta}) g(x_k)^T p_k. \end{aligned} \quad (3.5.27)$$

Since $p_k \rightarrow 0$, there must exist an index k_{bar} such that for all $k \geq k_{\text{bar}}$, ω_k will be small enough to satisfy

$$\omega_k < \delta(1 - 2\mu).$$

When ω_k satisfies this equation, the right-hand side of (3.5.27) will be negative for ω sufficiently small (since $\mu < \frac{1}{2}$ and $g(x_k)^T p_k < 0$), and hence (3.5.26) is satisfied for sufficiently large k . ■

3.6. Trust-Region Methods

Trust-region methods are an alternative to line-search methods. At each iteration of a line-search method, the search direction is defined as the unconstrained minimizer of a quadratic model function. In contrast, the idea of *trust-region methods* is to rely on the local quadratic model only as long as it adequately reflects the behavior of f . As the quadratic model is most accurate “near” the current point, one way of ensuring an accurate quadratic model is to restrict the *length* of the search direction. In the most widely used version of this idea, the search direction is determined by the subproblem

$$\begin{aligned} & \underset{p \in \mathbb{R}^n}{\text{minimize}} && q_k(x_k + p) \\ & \text{subject to} && \|p\| \leq \delta_k, \end{aligned}$$

for some positive scalar δ_k . The constraint $\|p\| \leq \delta_k$ can be interpreted as defining a *region of trust*, i.e., a region in which we trust that $q_k(x_k + p) \approx f(x_k + p)$. The scalar δ_k , known as the *trust-region radius*, is updated to reflect the changing behavior of f as the iterations proceed. More details of this procedure are given in §3.6.

In §3.6 below, it will be shown that for certain values of δ_k , the trust-region problem gives a direction p_k such that $f(x_k + p_k)$ is sufficiently lower than $f(x_k)$. This implies that the new iterate can be defined simply as $x_{k+1} = x_k + p_k$, without the need for a step length. Following our convention of using s_k to denote the change in variables $x_{k+1} - x_k$, from this point we will use s to denote the variables associated with the trust-region subproblem.

Trust-region methods are an alternative to line-search methods. They provide a different way of exploiting the generic quadratic model:

$$q_k(x_k + s) = f(x_k) + g(x_k)^T s + \frac{1}{2} s^T B_k s. \quad (3.6.1)$$

Under certain restrictions on s , it holds that

$$\lim_{\|s\| \rightarrow 0} \frac{f(x_k) - f(x_k + s)}{q_k(x_k) - q_k(x_k + s)} = 1.$$

This result implies that if μ is a scalar parameter chosen such that $0 < \mu < 1$, then there exists a positive scalar δ_k such that

$$\frac{f(x_k) - f(x_k + s)}{q_k(x_k) - q_k(x_k + s)} \geq \mu \quad \text{for all } \|s\| \leq \delta_k. \quad (3.6.2)$$

It follows that the sufficient decrease condition may be imposed by means of an explicit constraint on the norm of the search direction. If this value of δ_k is known, a suitable direction s_k can be found by solving the constrained problem

$$\begin{aligned} & \underset{s \in \mathbb{R}^n}{\text{minimize}} && q_k(x_k + s) \\ & \text{subject to} && \|s\| \leq \delta_k. \end{aligned} \quad (3.6.3)$$

The constraint $\|s\| \leq \delta_k$ can be interpreted as defining a *region of trust* in which we can expect f to be represented effectively by its quadratic model. This interpretation has resulted in the problem (3.6.3) being called the *trust-region problem*. Similarly, the quantity δ_k is known as the *trust-region radius*.

Trust-region methods have two important characteristics that distinguish them from line-search methods. First, since $x_k + s_k$ gives a sufficient decrease in f , the new iterate may be defined as $x_{k+1} = x_k + s_k$ without the need for a line search. Second, as long as the trust-region radius δ_k is bounded, the trust-region problem always has a bounded solution, regardless of the eigenvalues of B_k (see Lemma 3.6.1 below). Hence, B_k does not need to be positive definite.

In practice, the precise value of δ_k that gives a sufficient decrease is not known in advance and must be estimated iteratively. If (3.6.2) is not satisfied, δ_k is decreased in steps (and the trust-region problem re-solved) until a suitable value is found.

For efficiency reasons, it may be necessary to *increase* δ_k in later iterations. If the actual reduction in f is comparable to, or better than, the reduction predicted by the model, then it is usually prudent to increase δ_k in order to exploit the accuracy of the model. As the solution is approached, δ_k should be sufficiently large that the “natural” convergence rate of the method is not inhibited. For example, if $B_k = H(x_k)$ and x_k lies in a region where the second-order sufficient conditions for optimality hold, then δ_k should be large enough that the unconstrained minimizer of the quadratic model (3.6.1) be accepted. The precise rate at which δ_k is increased or reduced is governed by parameters η , μ , ω_1 and ω_2 . Algorithm 3.6.1 below describes the details of a simple trust-region algorithm for minimizing f .

Algorithm 3.6.1. Simple Trust-Region Algorithm

Specify constants $k_{\max}$, $0 < \mu < \mu_e < 1$;

$k \leftarrow 0$; $\delta_k \leftarrow 1$; $x_k \leftarrow x_0$;

while $k \leq k_{\max}$ and **not converged** **do**

 Compute s_k as an approximate solution of $\min \{q_k(x_k + s) \mid \|s\| \leq \delta_k\}$;

$\rho_k \leftarrow (f(x_k) - f(x_k + s_k)) / (q_k(x_k) - q_k(x_k + s))$;

if $\rho_k \geq \mu$ **then**

$x_{k+1} \leftarrow x_k + s_k$;

if $\rho_k \geq \mu_e$ **then** $\delta_{k+1} = \max\{\delta_k, 2\|s_k\|\}$ **else** $\delta_{k+1} \leftarrow \delta_k$;

else

$x_{k+1} \leftarrow x_k$; $\delta_{k+1} \leftarrow \frac{1}{2}\|s_k\|$;

end

$k \leftarrow k + 1$;

end while

The choice of norm in the constraint $\|s\| \leq \delta_k$ has important practical and theoretical consequences. An advantage of choosing the two-norm is that it is possible to compute a *global* minimizer of the trust-region problem. If B_k is indefinite, the trust-region problem may have many local minimizers. If only a *local* minimizer is found, there is the possibility that the convergence will be disrupted by the calculation of alternative local minimizers. However, regardless of the choice of norm, the presence of the constraint makes the trust-region step considerably more expensive to compute than the line-search step. In particular, we will show that for each value of δ_k , the trust-region problem must be solved using an iterative method in which each iteration involves a linear system comparable to that solved by a line-search method.

3.6.1. Computing the trust-region direction

The main effort associated with the trust-region method concerns the calculation of the trust-region direction s_k . In this section we consider the solution of this problem when the trust-region constraint is defined using the two-norm. Unless otherwise stated, $\|\cdot\|$ will denote the two-norm. To simplify the discussion, we drop the suffix k and (without loss of generality) discard the constant term in the quadratic model. Let $\mathcal{Q}(s)$ denote the function $g^T s + \frac{1}{2} s^T B s$, which predicts the *change* in the objective corresponding to a step $x + s$. The trust-region problem may then be written in the form

$$\begin{aligned} & \underset{s \in \mathbb{R}^n}{\text{minimize}} && \mathcal{Q}(s) = g^T s + \frac{1}{2} s^T B s \\ & \text{subject to} && \|s\|_2 \leq \delta. \end{aligned} \quad (3.6.4)$$

The objective function $\mathcal{Q}(s)$ is continuous on the closed bounded set $\{s \mid \|s\|_2 \leq \delta\}$, and the existence of a global solution is guaranteed by the Extreme Value Theorem. The algorithm for computing this global solution is based on the following lemma.

Lemma 3.6.1. *Let δ be a given positive constant. A vector s is a global solution of the trust-region problem if and only $\|s\|_2 \leq \delta$ and there exists a $\lambda \geq 0$ such that*

$$(B + \lambda I)s = -g, \quad \lambda(\delta - \|s\|_2) = 0, \quad (3.6.5)$$

with $B + \lambda I$ positive semidefinite.

Proof. Suppose that λ and s satisfy (3.6.5) with $\|s\|_2 \leq \delta$ and $B + \lambda I$ positive definite. Then Lemma 3.2.1 implies that s minimizes the quadratic function

$$\hat{\mathcal{Q}}(w) = g^T w + \frac{1}{2} w^T (B + \lambda I) w.$$

Hence, $\hat{\mathcal{Q}}(w) \geq \hat{\mathcal{Q}}(s)$, which implies that

$$g^T w + \frac{1}{2} w^T B w \geq g^T s + \frac{1}{2} s^T B s + \frac{1}{2} \lambda (s^T s - w^T w) \quad (3.6.6)$$

for all $w \in \mathbb{R}^n$. As $\lambda s^T s = \lambda \delta^2$ and $\lambda \geq 0$ we have

$$g^T w + \frac{1}{2} w^T B w \geq g^T s + \frac{1}{2} s^T B s + \frac{1}{2} \lambda (\delta^2 - w^T w) \quad (3.6.7)$$

whenever $\|w\|_2 \leq \delta$. It follows that $\mathcal{Q}(w) \geq \mathcal{Q}(s)$ for all $\|w\|_2 \leq \delta$ and s solves the trust-region problem (3.6.4).

Now suppose that s solves (3.6.4). If $\|s\|_2 < \delta$ then s is a bounded unconstrained minimizer of $\mathcal{Q}(w)$ and Lemma 3.2.1 implies that (3.6.5) holds with $\lambda = 0$ and B positive semidefinite.

If $\|s\|_2 = \delta$ then s must solve the equality constraint problem $\min\{\mathcal{Q}(w) \mid \|w\|_2 = \delta\}$. The method of Lagrange ensures the existence of a γ such that $\nabla L(s, \gamma) = 0$, where $L(w, \tau) = \mathcal{Q}(w) - \tau(\delta - \|w\|_2)$. Differentiating L with respect to both w and τ , we obtain

$$\nabla L(s, \gamma) = \begin{pmatrix} g + B s + \frac{\gamma}{\|s\|_2} s \\ \delta - \|s\|_2 \end{pmatrix} = 0.$$

This implies that (3.6.5) holds for this s and $\lambda = \gamma/\delta$.

Since s solves the trust-region problem, it must hold that $\mathcal{Q}(s) \leq \mathcal{Q}(w)$ for any feasible w , which implies that

$$g^T w + \frac{1}{2} w^T B w \geq g^T s + \frac{1}{2} s^T B s + \frac{1}{2} \lambda (s^T s - w^T w)$$

for any w such that $w^T w = s^T s$. Substituting the expression $(B + \lambda I)s$ for $-g$ and collecting terms gives

$$\frac{1}{2} (w - s)^T (B + \lambda I) (w - s) \geq 0,$$

for all vectors w of length $\|s\|_2$. As the direction of w is arbitrary, $w - s$ is arbitrary, and this expression implies that $B + \lambda I$ is positive semidefinite.

To show that λ is nonnegative, note that Lemma 3.2.1 implies that (3.6.7) is valid for every $w \in \mathbb{R}^n$. If λ is not positive, then (3.6.7) implies that $\mathcal{Q}(w) \geq \mathcal{Q}(s)$ whenever $\|w\|_2 \geq \|s\|_2$. As s solves the trust-region problem, we must have that s is an unconstrained minimizer of $\mathcal{Q}$ and then Lemma 3.6.1 implies that $\lambda = 0$. Hence $\lambda \geq 0$ as required. ■

Lemma 3.6.1 has several important implications. First, suppose that B is positive definite, and let $s^N = -B^{-1}g$ denote the unconstrained minimizer of the quadratic model. If $\|s^N\|_2 \leq \delta$, then s^N will solve the constrained subproblem, giving $s = s^N$ and $\lambda = 0$. Otherwise, $\|s^N\|_2 > \delta$, and there exists a positive λ and a direction s such that $(B + \lambda I)s = -g$. This implies that if the trust-region radius is sufficiently small, a “modified” Hessian $B + \lambda I$ is used to define the search direction even though B is positive definite. This is in contrast to line-search methods, which always use the minimizer of a positive-definite quadratic model.

When B is positive definite, it is instructive to consider the locus of points $x + s$ obtained by solving a given trust-region problem with different values of δ . We are particularly interested in what happens as $\delta \rightarrow 0$. The proof of Lemma 3.6.1 implies that $\lambda \rightarrow \infty$ as $\delta \rightarrow 0$. Hence, it is sufficient to consider the locus of points $x + s_\lambda$ as $\lambda \rightarrow \infty$, where $s_\lambda = -(B + \lambda I)^{-1}g$. For all $\delta \geq \|s^N\|_2$, the points $x + s_\lambda$ lie at the minimizer of the quadratic model. However, as δ varies continuously from 0 to $\|s^N\|_2$, the points $x + s_\lambda$ describe a path joining x and $x + s^N$ known as the *dog-leg*. It is not difficult to show that this dog-leg path approaches x along the negative gradient vector $-g$. The vector s_λ can be written in terms of the spectral decomposition of B :

$$B = V \Lambda V^T = \sum_{i=1}^n \lambda_i v_i v_i^T,$$

where the columns of V , $\{v_i\}$, $i = 1, \dots, n$, are the eigenvectors of B and Λ is the matrix of eigenvalues $\{\lambda_i\}$ ordered so that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$. Substituting for B in the definition of s_λ gives

$$s_\lambda = -(B + \lambda I)^{-1}g = -\sum_{i=1}^n \frac{v_i^T g}{\lambda + \lambda_i} v_i = -\frac{1}{\lambda} \sum_{i=1}^n \frac{v_i^T g}{1 + \lambda_i/\lambda} v_i. \quad (3.6.8)$$

As $\lambda \rightarrow \infty$, the vector s_λ becomes parallel to $-(1/\lambda) V V^T g = -(1/\lambda)g$. Hence, as the trust-region radius goes to zero, the trust-region direction becomes parallel to a small multiple of $-g$, the steepest-descent direction.

If B is indefinite (in which case the unconstrained model is unbounded below), or $\|s^N\|_2 > \delta$, the scalar λ is positive, and the restriction on the norm of s in (3.6.3) will apply. In this case, for a given δ , the vector s is defined by the solution $\alpha = \lambda$ of the nonlinear equations

$$\varrho(\alpha) \equiv \delta^2 - \|s(\alpha)\|_2^2 = 0, \quad \text{where} \quad (B + \alpha I)s(\alpha) = -g. \quad (3.6.9)$$

These equations can be solved using a one-dimensional variant of Newton's method with appropriate safeguards. The properties of $\varrho(\alpha)$ may be investigated by considering the spectral decomposition of B :

$$\|s(\alpha)\|_2^2 = \|(B + \alpha I)^{-1}g\|_2^2 = \sum_{i=1}^n \left(\frac{v_i^T g}{\alpha + \lambda_i} \right)^2. \quad (3.6.10)$$

It follows that $\|s(\alpha)\|_2^2$ is a rational function with poles at a subset of the eigenvalues of $-B$.

To illustrate some of the features of the trust-region problem, we consider a problem with model given by

$$B = \begin{pmatrix} 1 & 0 \\ 0 & -2 \end{pmatrix}, \quad \text{and} \quad g = \begin{pmatrix} 2 \\ 4 \end{pmatrix}. \quad (3.6.11)$$

The system $Bs^N = -g$ defines a step $s^N = (-2, 2)^T$ to the saddle point of the model (3.4.15). However, s^N is *not* a descent direction for either q or f since $g^T s^N = 4$.

Consider the trust-region subproblem with $\delta = 4$. A simple calculation gives $\|s^N\|_2 < \delta$, but as H is indefinite, a better value of the model must be defined on the boundary of the trust region. The equation $(B + \alpha I)s(\alpha) = -g$ gives $s(\alpha) = (-2/(\alpha + 1), -4/(\alpha - 2))^T$. Hence the required λ is the unique root of the nonlinear equation

$$\delta^2 - \|s(\alpha)\|_2^2 = 16 - \frac{4}{(\alpha + 1)^2} - \frac{16}{(\alpha - 2)^2} = 0$$

for which $B + \alpha I$ is positive semidefinite. The required root is $\lambda = 3.00787$ (to six figures), which gives the (unique) trust-region direction $s = (-0.49902, -3.96875)$. The level curves of the model function and the position of the search directions s^N and s are depicted in Fig. 3.4.

Unfortunately, Newton's method for zero-finding does not work well on problems with singularities. It is better to formulate the equivalent zero-finding problem

$$\vartheta(\alpha) \equiv \frac{1}{\|s(\alpha)\|_2^2} - \frac{1}{\delta^2} = 0. \quad (3.6.12)$$

The function ϑ has no poles, and is better behaved than ϱ . Each evaluation of ϑ in the zero-finding algorithm requires a Cholesky factorization of the matrix $B + \alpha I$ in order to obtain $s(\alpha)$ from (3.6.9) and to check the positive definiteness of $B + \alpha I$.

Lemma 3.6.2. *If $v_n^T g \neq 0$, where v_n is the eigenvector corresponding to the smallest eigenvalue λ_n of B , then $\vartheta(\alpha)$ has a unique zero $\alpha = \lambda$ in $(-\lambda_n, \infty)$.*

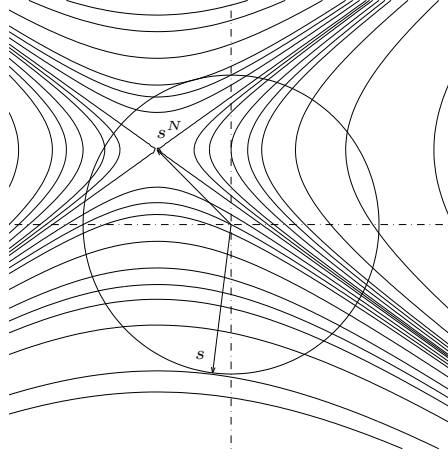


Figure 3.4: Level curves of the quadratic model $q = 2p_1 + 4p_2 + \frac{1}{2}p_1^2 - p_2^2$ given by B and g (3.6.11). The model has a saddle point at $(-2, 2)$ and decreases along directions starting at $(-2, 2)$ parallel to the x_2 axis. The interior of the circle depicts the set of points satisfying the trust-region constraint $\|s\|_2 \leq 4$. There are two local minimizers of q , but the direction $s = (-0.49902, -3.96875)$ (to six figures) corresponding to $\lambda = 3.00787$ is the global solution.

Proof. To prove the existence and uniqueness of a zero $\alpha = \lambda$ of ϑ in $(-\lambda_n, \infty)$ we show that ϑ is increasing and concave in $(-\lambda_n, \infty)$.

If g is not orthogonal to v_n , then γ_n is nonzero in (3.6.10), and hence

$$\lim_{\alpha \rightarrow -\lambda_n} \frac{1}{\|s(\alpha)\|_2} = 0, \quad \lim_{\alpha \rightarrow +\infty} \frac{1}{\|s(\alpha)\|_2} = \infty. \quad (3.6.13)$$

Differentiating $\vartheta(\alpha)$ with respect to α gives

$$\vartheta'(\alpha) = -\frac{w^T s}{\|s\|_2^3}, \quad \text{where} \quad w = \frac{ds}{d\alpha} = (B + \alpha I)^{-2} g. \quad (3.6.14)$$

Premultiplying w by s^T gives

$$w^T s = s^T (B + \alpha I)^{-2} g = -s^T (B + \alpha I)^{-1} s,$$

and (3.6.14) may be written as

$$\vartheta'(\alpha) = \frac{s^T (B + \alpha I)^{-1} s}{\|s\|_2^3}, \quad (3.6.15)$$

which is positive for all $\alpha \in (-\lambda_n, \infty)$.

Differentiating $\vartheta'(\alpha)$ (3.6.14) gives

$$\vartheta''(\alpha) = \frac{3}{\|s\|_2^5} (w^T s)^2 - \frac{1}{\|s\|_2^3} \left(s^T \left(\frac{dw}{d\alpha} \right) + w^T w \right). \quad (3.6.16)$$

Differentiating $s^T w$ with respect to α gives

$$s^T \left(\frac{dw}{d\alpha} \right) = 2w^T w.$$

If this expression is substituted in (3.6.16), ϑ'' may be rearranged so that

$$\vartheta''(\alpha) = -\frac{3}{\|s\|_2^5} ((w^T w)(s^T s) - (w^T s)^2). \quad (3.6.17)$$

The Cauchy-Schwartz inequality implies $(w^T w)(s^T s) - (w^T s)^2 \geq 0$, and $\vartheta''(\alpha)$ is negative for all α .

Finally, since $B + \alpha I$ is positive definite for $\alpha > -\lambda_n$, (3.6.13), (3.6.15) and (3.6.17) imply that

$$\vartheta'(\alpha) \geq 0 \quad \text{and} \quad \vartheta''(\alpha) < 0,$$

so that $\vartheta(\alpha)$ must have a unique zero in $(-\lambda_n, \infty)$. ■

The only exception to this lemma occurs when $v_n^T g = 0$. In this case, the nonlinear equations (3.6.9) and (3.6.12) have no solutions in $(-\lambda_n, \infty)$, and the zero-finding algorithm will fail. This happens, for example, when $g = 0$ and B is indefinite. In the trust-region literature, this situation is known as the “hard-case”. When the hard-case occurs, the optimal value of λ makes $B + \lambda I$ singular, and only part of s (the range-space portion) can be determined from the linear system $(B + \lambda I)s = -g$. The null-space portion must be computed explicitly by estimating a vector z that lies in the null-space of $B + \lambda I$. The length of the null-space portion is then determined from the constraint $\|s\|_2 = \delta$.

To help illustrate the hard case, consider a trust-region problem in which the quadratic model is defined by

$$B = \begin{pmatrix} 1 & 0 \\ 0 & -2 \end{pmatrix}, \quad \text{and} \quad g = \begin{pmatrix} 2 \\ 0 \end{pmatrix}. \quad (3.6.18)$$

Note that $v_n^T g = e_2^T g = 0$, and the calculation of s is one of the hard cases. The search direction must satisfy the system $(B + \lambda I)s = -g$, with $\lambda = 2$, i.e.,

$$\begin{pmatrix} 3 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = -\begin{pmatrix} 2 \\ 0 \end{pmatrix}.$$

This system can be used to define the first component of s , giving $s_1 = -\frac{2}{3}$. The second component must be determined from the trust-region constraint $\delta - \|s\|_2 = 0$, giving $s_2 = \pm\sqrt{\delta^2 - s_1^2} = \pm 3.94405$ (to six figures). Note that the trust-region step is not unique in this case since both $(-\frac{2}{3}, 3.94405)^T$ and $(-\frac{2}{3}, -3.94405)^T$ give the same global minimum of the model. This situation is depicted in Fig. 3.5.

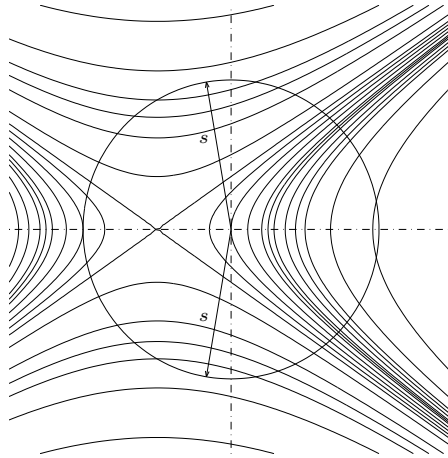


Figure 3.5: Level curves of the quadratic model $q = 2p_1 + \frac{1}{2}p_1^2 - p_2^2$ defined by B and g (3.6.18). This figure depicts the so-called “hard-case” associated with the trust region problem. The trust-region solution is not unique—either of the vectors shown is a suitable search direction.

Chapter 4

Equality Constraints

4.1. Introduction

In its most general form, the *constrained* nonlinear programming problem is to find a point x^* such that $f(x^*)$ achieves its least value over a given set of acceptable values of x . The set of acceptable values is denoted by $\mathbb{F}$ and is known as the *feasible region*. Every point in $\mathbb{F}$ is said to be *feasible*. (All sets are in $\mathbb{R}^n$ unless stated otherwise.)

In this chapter we consider the case where the set $\mathbb{F}$ is defined by means of a set of functional *equalities*. In this case, the optimization problem is written in the general form:

$$\begin{array}{ll} \text{NEP} & \begin{array}{l} \text{minimize} \quad f(x) \\ \text{subject to} \quad c(x) = 0. \end{array} \end{array}$$

where each $c_i(x)$ is a (possibly) nonlinear function of the n variables $x_1, x_2, \dots, x_n$. This problem has m equality constraints and the m functions $\{c_i(x)\}$ are known as the *constraint functions* of the problem. For convenience, we will often consider the vector function $c(x)$ that has the constraint function $c_i(x)$ as its i th component. Throughout, it is assumed that f and c are twice-continuously differentiable—i.e., the second partial derivative of $c_i(x)$ with respect to each of the variables exists and is continuous.

Fig. 4.1 depicts the simple problem with two variables:

$$\begin{array}{ll} \text{minimize} & -x_1x_2^2 \\ \text{subject to} & 2 - x_1^2 - x_2^2 = 0. \end{array} \tag{4.1.1}$$

This problem has one equality constraint $c_1 = 2 - x_1^2 - x_2^2$, which implies that we wish to find the smallest value of f over all points lying on the circle of radius $\sqrt{2}$ centered at the origin. The objective is $f(x) = -x_1x_2^2$, which has the property that $f(x) \rightarrow -\infty$ as $x_1 \rightarrow +\infty$. The objective is clearly *unbounded* below in $\mathbb{R}^n$ and there is no unconstrained minimizer of f . This situation often occurs in constrained optimization and it implies that the constraints determine the location of a solution as well as the objective. The solution may be determined approximately by inspection. If we start at the point $(0, \sqrt{2})$ and move clockwise round the circle, we pass through level curves of f with decreasing values until we are in the region of $(0.8, 1.1)$, where we again start to pass through level curves

with increasing values. In fact the solution lies at $x^* = (0.8165, 1.1547)$ (rounded to four figures).

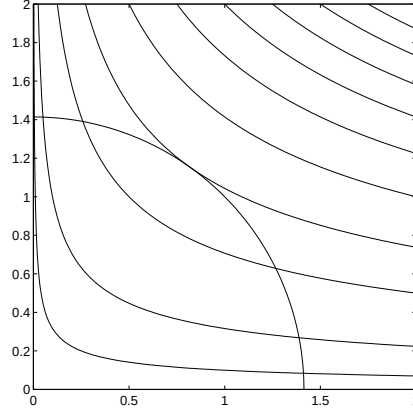


Figure 4.1: This graph depicts the level curves and constraint functions for the two-dimensional example (4.1.1). The feasible region consists of points lying on the circle $x_1^2 + x_2^2 = 2$. The solution lies at the point $x^* = (0.8165, 1.1547)$.

Definition 4.1.1. (*Constrained global minimizer.*) A point x^* is a constrained global minimizer of f if $x^* \in \mathbb{F}$ and $f(x^*) = \min\{f(x) \mid x \in \mathbb{F}\}$. ■

Definition 4.1.2. (*Constrained local minimizer.*) Let f be a function defined for all $x \in \mathbb{R}^n$. A point x^* is a constrained local minimizer of f if there is an open ball $\mathcal{B}(x^*, \delta)$ such that

$$f(x^*) \leq f(x) \quad \text{for all } x \in \mathcal{B}(x^*, \delta) \cap \mathbb{F}.$$

A point x^* is a strict or proper constrained minimizer if this inequality holds with strict inequality except at x^* itself, i.e., $f(x^*) < f(x)$ for all $x \in \mathcal{B}(x^*, \delta) \cap \mathbb{F}$. ■

This definition states that $f(x) \geq f(x^*)$ for all points x in some set that is open relative to the feasible region $\mathbb{F}$ (see Definition 1.4.7).

Definition 4.1.3. (*Isolated constrained local minimizer.*) A constrained local minimizer x^* is isolated if there is an open ball $\mathcal{B}(x^*, \delta)$ such that x^* is the unique constrained local minimizer in $\mathcal{B}(x^*, \delta) \cap \mathbb{F}$. ■

As in the unconstrained case, the region $\mathcal{B}(x^*, \delta)$ may be replaced by the interior of any closed bounded set S containing x^* , as in the next equivalent definition.

Definition 4.1.4. (*Constrained local minimizer.*) The point x^* is a constrained local minimizer of f if there exists a compact set S such that

$$x^* \in \text{int}(S) \cap \mathbb{F} \quad \text{and} \quad f(x^*) = \min\{f(x) \mid x \in S \cap \mathbb{F}\}. \quad \blacksquare$$

We assume throughout that the constraint functions $\{c_i(x)\}$ are at least continuously differentiable. Given a nonlinear constraint function $c_i(x)$, its gradient is denoted by $\nabla c_i(x)$, and its Hessian by $\nabla^2 c_i(x)$. For an m -vector $c(x)$ of constraint functions, the $m \times n$ Jacobian matrix of c is denoted by $J(x)$, whose i th row (the transposed gradient of c_i) is $\nabla c_i(x)^T$. For the objective function $f(x)$, the n -vector $g(x)$ denotes the gradient (vector of first partial derivatives) of f , and the $n \times n$ symmetric matrix $H(x)$ denotes the Hessian (matrix of second partial derivatives) of f .

The most powerful algorithms for constrained optimization are based on seeking a point that satisfies certain *optimality conditions* that are known to hold at a solution. The key role of these conditions in algorithm design has already been discussed in the context of *unconstrained* problems, in which the gradient of the objective function must vanish at the solution, and the Hessian matrix must be positive semidefinite. We seek analogous conditions for constrained problems.

4.2. Optimality Conditions

In the unconstrained case, a minimizer x^* satisfies $f(x) \geq f(x^*)$ for all x in some sufficiently small neighborhood of x^* . In §3.2 it was demonstrated that every point x in this neighborhood of x^* can be reached by taking the appropriate step along a vector p in the direction of $x - x^*$. This implies that allowing p to take on every value in $\mathbb{R}^n$, the neighborhood of x^* can be characterized as points of the form $x^* + \alpha p$ with α sufficiently small. This allows a minimizer to be defined in terms of the univariate problem $f(x + \alpha p)$, defined for every direction p emanating from x^* .

In the constrained case we have the added complication that the values of x in the relation $f(x) \geq f(x^*)$ must be restricted to *feasible points*. However, a similar approach can be used as long as f is compared at neighboring points that are reachable from x^* by a path $x(\alpha)$ that is feasible for every value of α . In general, this path must be *nonlinear* in order to follow the nonlinearities of the constraints. However in some situations, such as when c is an affine function, $x(\alpha)$ can be a linear path $x(\alpha) = x^* + \alpha p$, as in the unconstrained case.

In general, feasibility can be retained with respect to nonlinear equality constraints only by movement along a *nonlinear* feasible path in $\mathbb{R}^n$. For example, the nonlinear constraint $2 - x_1^2 - x_2^2 = 0$ defines a circle centered at the origin, whose curved boundary is a feasible path.

Definition 4.2.1. (*Feasible Path.*) A feasible path is a directed twice-differentiable curve $x(\alpha)$ emanating from a feasible point x , parameterized by the scalar α such that

$$x(0) = x \quad \text{and} \quad c(x(\alpha)) = 0,$$

for all α satisfying $0 \leq \alpha < \sigma$, where $\sigma > 0$, and $dx(\alpha)/d\alpha|_{\alpha=0} \neq 0$.

In order to remain feasible, each constraint function $c_i(x(\alpha))$ must remain identically zero at all points $\alpha \in [0, \sigma)$. This implies that the rate of change, or derivative of each $c_i(x)$ must be zero for all points on the path, with

$$\left. \frac{d}{d\alpha} c_i(x(\alpha)) \right|_{\alpha=0} = \nabla c_i(x)^T p = 0,$$

where p is the vector tangent to $x(\alpha)$ at $\alpha = 0$,

$$p = \left. \frac{d}{d\alpha} x(\alpha) \right|_{\alpha=0} = \lim_{\alpha \rightarrow 0^+} \frac{1}{\alpha} (x(\alpha) - x(0)).$$

A path that is simultaneously feasible for *every* constraint of NEP must have a tangent p such that $\nabla c_i(x)^T p = 0$ for $i = 1, 2, \dots, m$, which implies

$$J(x)p = 0. \quad (4.2.1)$$

This relation states that any tangent p to a feasible path lies in the *null space* of the matrix $J(x)$, i.e., any tangent must be a member of the set $\{z \mid J(x)z = 0\}$. It is worth emphasizing that there are an infinite number of tangents to a *given* feasible path. If the parameterization of the path is changed from α to $\sigma\alpha$, where σ is a positive constant, then $c(x(\sigma\alpha)) = 0$ as before, but the tangent vector is now σp . The scale of α determines the “speed” at which the path is traversed, but not the direction of the path.

Given an $m \times n$ matrix $J(x)$ of rank r , we shall use $Z(x)$ to denote a generic $n \times (n-r)$ matrix with linearly independent columns such that $J(x)Z(x) = 0$. These properties imply that the columns of $Z(x)$ form a basis for the $(n-r)$ -dimensional null space of $J(x)$, i.e., every vector p such that $J(x)p = 0$ can be expressed as a linear combination of the columns of $Z(x)$. Although the null space of $J(x)$ is unique (i.e., it always contains the same vectors), the matrix $Z(x)$ is *not* unique except when $\text{rank}(J(x)) = n$ (in which case $Z(x)$ is vacuous). In fact, if M is any $(n-r) \times (n-r)$ *nonsingular* matrix, the columns of $Z(x)M$ also form a basis for the null space of $J(x)$.

From the definition of a basis, every vector p such that $J(x)p = 0$ may be written as a linear combination of the columns of $Z(x)$, so that

$$J(x)p = 0 \quad \text{if and only if} \quad p = Z(x)p_Z$$

for some vector p_Z .

To illustrate some of the notation introduced above, consider the constraint $2 - x_1^2 - x_2^2 = 0$ at the point $x = (0, \sqrt{2})^T$. A suitable feasible path is given by $x(\alpha) = (\sqrt{2} \sin \alpha, \sqrt{2} \cos \alpha)^T$. Thus

$$\frac{d}{d\alpha} x(\alpha) = \begin{pmatrix} \sqrt{2} \cos \alpha \\ -\sqrt{2} \sin \alpha \end{pmatrix} \quad \text{with} \quad p \equiv \left. \frac{d}{d\alpha} x(\alpha) \right|_{\alpha=0} = \begin{pmatrix} \sqrt{2} \\ 0 \end{pmatrix}.$$

We are particularly interested in the *set* of tangent vectors corresponding to *every* feasible path at x . As feasible paths can be rescaled by the multiplication of an arbitrary positive constant, a given tangent p to a feasible path may be represented by any vector σp , with $\sigma > 0$. Consideration of the set of all such tangents, together with the zero vector, leads to the idea of the *tangent cone* at a feasible point x .

Definition 4.2.2. Let $T(x)$ denote the set of tangent vectors associated with every feasible path emanating from the feasible point x , i.e.,

$$T(x) = \{p \mid p \text{ a nonzero tangent to a feasible path emanating from } x.\}$$

Then $T^0(x) = T(x) \cup \{0\}$ is known as the tangent cone of c at x . ■

Assume that at least one feasible path $x^*(\alpha)$ exists at the feasible point x^* , which is equivalent to the assumption that every neighborhood of x^* contains feasible points different from x^* . (If no feasible path exists, x^* is a unique feasible point that must be optimal independently of the objective function.)

Under this assumption, x^* is a local solution of NEP if and only if $f(x^*) \leq f(x^*(\alpha))$ for α sufficiently small along *all* feasible paths emanating from x^* . The point x^* can be a local solution only if f does not initially decrease along any feasible path, for we have

$$\begin{aligned} f(x^*(\alpha)) &= f(x^*(0)) + \alpha \left. \frac{d}{d\alpha} f(x^*(\alpha)) \right|_{\alpha=0} + O(\alpha^2) \\ &= f(x^*) + \alpha g(x^*)^T p + O(\alpha^2), \end{aligned} \quad (4.2.2)$$

and if $g(x^*)^T p < 0$, $x^*(\alpha)$ can always be chosen so that infinitesimally small α will lead to a decrease in f . The next theorem is almost identical to that of Theorem 3.2.7—the only difference being the restriction of the directions p to the tangent cone.

Theorem 4.2.1. *Assume that f and c are differentiable. Let x^* be a feasible point at which at least one feasible path exists. Then x^* is a local solution of NEP only if $g(x^*)^T p \geq 0$ for every p in the tangent cone. ■*

The next step is to convert this necessary condition into an “algebraic” condition that can be verified computationally. To do this we need to be able to define the tangent cone algebraically. It may be observed that at any feasible point, the tangent cone closely resembles the null-space of the Jacobian, i.e., the set

$$\text{null}(J(x)) = \{z \mid J(x)z = 0\}.$$

For example, since every tangent to a feasible path must satisfy (4.2.1) it follows that $T^0(x) \subset \text{null}(J(x))$. In fact, in most situations, these sets *are* identical and this fact makes it possible to use simple verifiable optimality conditions based on the calculation of Lagrange multipliers.

4.2.1. Lagrange multipliers

The next step in our derivation involves investigating the sign of $g(x^*)^T p$ when p is any vector in $\text{null}(J(x^*))$.

Lemma 4.2.1. *(Existence of Lagrange Multipliers for NEP.) Assume that f and c are differentiable at a feasible point x^* . Then*

$$g(x^*)^T p \geq 0 \quad \text{for all } p \text{ such that } J(x^*)p = 0 \quad (4.2.3)$$

if and only if there exists a vector λ^* such that

$$g(x^*) = J(x^*)^T \lambda^* = \sum_{i=1}^m \lambda_i^* \nabla c_i(x^*). \quad (4.2.4)$$

Proof. If $g(x^*) = J(x^*)^T \lambda^*$ for some λ^* , then for all $p \in \text{null}(J(x^*))$, we have

$$g(x^*)^T p = (J(x^*)^T \lambda^*)^T p = \lambda^{*T} J(x^*) p = 0,$$

so that $g(x^*)^T p \geq 0$.

It remains to show that if there is no vector λ^* such that $g(x^*) = J(x^*)^T \lambda^*$, then there exists a vector p such that $g(x^*)^T p < 0$ and $J(x^*) p = 0$. If no λ^* exists then $g(x^*) \notin \text{range}(J(x^*)^T)$ and $g(x^*)$ can be written uniquely as

$$g(x^*) = g_R + g_N, \quad g_N \neq 0,$$

where $g_R \in \text{range}(J(x^*)^T)$, $g_N \in \text{null}(J(x^*))$, and $g_N^T g_R = 0$. The vector $p = -g_N$ satisfies $J(x^*) p = 0$, with

$$g(x^*)^T p = (g_R + g_N)^T p = -(g_R + g_N)^T g_N = -\|g_N\|_2^2 < 0.$$

This implies that (4.2.3) does not hold, as required. ■

The components of λ^* are known as the *Lagrange multipliers* at x^* , and they are unique if $J(x^*)$ has full row rank. The existence of Lagrange multipliers can be verified computationally, since λ^* can be calculated as the solution of the overdetermined equations $J(x^*)^T \lambda = g(x^*)$.

Lemma 4.2.1 implies that when the tangent cone and $\text{null}(J(x^*))$ are identical sets, then $g(x^*)$ must lie in $\text{range}(J(x^*)^T)$ if x^* is optimal. These sets are “almost” always identical in practice, but some pathological examples exist in which this equivalence doesn’t hold. It is customary to make *regularity assumption* or *constraint qualification* which essentially is an assumption that the pathological case does not apply.

One commonly used constraint qualification is that the rows of $J(x^*)$ are linearly independent. If the problem always has this property, then $T(x^*) = \text{null}(J(x^*))$, the existence of Lagrange multipliers signal the presence of optimality, and the next section can be skipped.

4.2.2. The constraint qualification

When c is nonlinear, there may be nonzero vectors p satisfying $J(x)p = 0$ that are not tangents to a feasible path. For example, consider the two constraints $x_1^3 - x_2 = 0$, and $x_2 = 0$. At the point $x = 0$, the Jacobian is

$$J(x) = \begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix}.$$

Any vector of the form $p = (\gamma, 0)^T$ with $\gamma \neq 0$ is a non-trivial vector in the null-space of $J(x)$. However, x is the unique feasible point for these constraints, and no feasible paths exist. Hence, nonzero vectors p exist satisfying $J(x^*)p = 0$ that are not tangents to any feasible path. It must be emphasized that it is not the uniqueness of x^* that causes the difficulty. For example, consider the constraints $x_1 - x_2^2 = 0$, and $x_2 - x_1^2 = 0$. At $x^* = 0$ the Jacobian is

$$J(x^*) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

and there are no nonzero vectors p such that $J(x^*)p = 0$ because $J(x^*)$ is nonsingular. In this case the origin is a locally unique feasible point and no feasible paths exist.

Unfortunately, if the tangent cone is not identical to the null-space of $J(x^*)$, the existence of Lagrange multipliers may have no bearing on whether or not x^* is a minimizer. For example, consider the problem

$$\begin{aligned} & \underset{x \in \mathbb{R}^2}{\text{minimize}} && \frac{1}{2}(x_1 + 1)^2 \\ & \text{subject to} && x_1^3 - x_2 = 0, \quad x_2 = 0. \end{aligned} \tag{4.2.5}$$

At the point $x^* = 0$ we have

$$g(x^*) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad J(x^*) = \begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix}.$$

Clearly, no λ^* exists such that $g(x^*) = J(x^*)^T \lambda^*$, yet x^* is the unique global constrained minimizer.

We have shown that when $c(x)$ is nonlinear, the condition $J(x)p = 0$ is not *sufficient* to ensure that p is tangent to a feasible path. In order to use Lagrange multipliers to test optimality, a condition is needed that ensures that all null vectors are tangents to feasible paths. Conditions of this kind are called *constraint qualifications*, and have been widely studied. We consider only one constraint qualification for equality constraints.

Definition 4.2.3. (*Constraint qualification for equality constraints.*) *The constraint qualification with respect to the equality constraints $c(x) = 0$ holds at x if x is feasible and every nonzero vector p satisfying $J(x)p = 0$ is tangent to a differentiable feasible path emanating from x . ■*

If the constraints are *linear*, the constraint qualification always holds regardless of the rank of the Jacobian. Consider the affine function $c(x) = Ax - b$ and assume that x satisfies $c(x) = 0$. If A is nonsingular, x is unique and there are no feasible paths. If x is not unique, consider any nonzero vector p such that $Ap = 0$. It is easy to see that p must be a tangent to a feasible path. For example, consider the linear path $x(\alpha) = x + \alpha p$. Clearly p is tangent to $x(\alpha)$ at $\alpha = 0$ (since $x(\alpha)$ is a ray, it is a tangent for all $\alpha \geq 0$). Moreover, $x(\alpha)$ is a feasible path for all α since

$$c(x(\alpha)) = Ax(\alpha) - b = A(x + \alpha p) - b = Ax - b + \alpha Ap = 0.$$

Any move from a feasible point along p produces another feasible point, and we accordingly say that a nonzero direction p satisfying $Ap = 0$ is a *feasible direction* for the constraints $Ax = b$.

In the linear constraint case, Lemma 4.2.1 implies that a x^* is a local solution of NEP only if there exist Lagrange multipliers such that $g(x^*) = A^T \lambda^*$.

For nonlinear constraints, Definition 4.2.3 has the unsatisfactory feature that it cannot easily be verified. In contrast, the following result provides a computationally practical test for verifying satisfaction of the constraint qualification when the constraints are nonlinear.

Theorem 4.2.2. *Assume that $\{c_i\}$, are m continuously differentiable constraint functions. A sufficient condition for the constraint qualification to hold at a feasible point x is that the rows of $J(x)$ are linearly independent.*

Proof. Let p satisfy $J(x)p = 0$. If there is no such p , the constraint qualification holds vacuously. Otherwise p must be expressible in the form Zp_z for some vector p_z . Construct the path $x(\alpha)$ from x by specifying $x(0) = x$ and $dx(\alpha)/d\alpha = Z(x(\alpha))p_z$, for $0 \leq \alpha \leq \sigma$, where $Z(x(\alpha))$ is a matrix whose columns form a basis for the null space of $J(x(\alpha))$. Since the constraint gradients $\{\nabla c_i\}$ are linearly independent at x and c is continuously differentiable, the matrix J is continuous with linearly independent rows in a neighborhood of x . Thus an appropriate representation of $Z(x(\alpha))$ is also continuous and of fixed rank in a neighborhood of x , and the constructed $x(\alpha)$ is a continuous, differentiable path. Expanding c_i along $x(\alpha)$ gives

$$c_i(x(\alpha)) = c_i(x) + \int_0^\alpha \frac{d}{d\xi} c_i(x(\xi)) d\xi = c_i(x) + \int_0^\alpha \nabla c_i(x(\xi))^T p(\xi) d\xi,$$

where $p(\xi)$ denotes the tangent $dx(\alpha)/d\alpha$ to the feasible path at $\alpha = \xi$. Using the identity $p(\xi) = Z(x(\xi))p_z$ and the fact that $c_i(x) = 0$, we have

$$c_i(x(\alpha)) = c_i(x) + \int_0^\alpha \nabla c_i(x(\xi))^T p(\xi) d\xi = \int_0^\alpha \nabla c_i(x(\xi))^T Z(x(\xi))p_z d\xi = 0.$$

The path $x(\alpha)$ is therefore feasible in some neighborhood of x , as required. ■

The idea of a *linearized* constraint is crucial in both the theory and practice of optimization. Consider the Taylor-series expansion of the i th constraint at a point $\bar{x}$:

$$c_i(x) = c_i(\bar{x}) + \nabla c_i(\bar{x})^T (x - \bar{x}) + O(\|x - \bar{x}\|^2).$$

If we ignore the higher-order terms we obtain the vector of *linear* or *affine* constraints $c_L(x, \bar{x}) = 0$, where

$$c_L(x, \bar{x}) \equiv c(\bar{x}) + J(\bar{x})(x - \bar{x}).$$

If the point at which c is being linearized is understood we shall drop the second argument and use $c_L(x)$ to denote $c_L(x, \bar{x})$.

If $\bar{x}$ is feasible then $c(\bar{x}) = 0$ and the linearized constraint surface is defined by the intersection of the m hyperplanes $\nabla c_i(\bar{x})^T x = \nabla c_i(\bar{x})^T \bar{x}$. This surface is tangent to the surface $c(x) = 0$ at $x = \bar{x}$.

Definition 4.2.4. Let $\bar{x}$ denote any feasible point. Let $T_L(\bar{x})$ denote the set of all tangent vectors associated with the linearized constraints $c_L(x, \bar{x})$. The tangent cone of c_L is then $T_L^0(x) = T_L(x) \cup \{0\}$. ■

We have shown that every $p \in T_L(\bar{x})$ satisfies $J(\bar{x})p = 0$, so that $T(x) \subset T_L(x)$. The tangent cone $T_L^0(x)$ is easily computed. By definition, $T_L^0(\bar{x})$ is given by

$$T_L^0(\bar{x}) = \{p \mid J(\bar{x})p = 0\}$$

Recall that any direction p satisfying $J(\bar{x})p = 0$ lies by definition in the *null space* of $J(\bar{x})$.

Definition 4.2.5. (Equivalent statement of the constraint qualification for equality constraints.) The constraint qualification with respect to the equality constraints $c(x) = 0$ holds at x if x is feasible and $T_L^0(x) = T^0(x)$. ■

4.2.3. The Karush-Kuhn-Tucker (KKT) conditions

At a point where the constraint qualification holds, the equation $J(x)p = 0$ provides a complete characterization of tangents to feasible paths.

We now state the important *first-order Karush-Kuhn-Tucker conditions* for a local solution of NEP. (The “order” of an optimality condition refers to the highest order of the derivatives that it contains.) The proof is an obvious consequence of Lemma 4.2.1 and the constraint qualification.

Theorem 4.2.3. (*First-order necessary optimality conditions for NEP.*) *If the constraint qualification holds at x^* , then x^* is a local solution of NEP only if*

$$g(x^*) = J(x^*)^T \lambda^* = \sum_{i=1}^m \lambda_i^* \nabla c_i(x^*), \quad (4.2.6)$$

for some λ^* . Equivalently, if $Z(x^*)$ denotes a null-space basis for $J(x^*)$, then

$$Z(x^*)^T g(x^*) = 0. \quad \blacksquare \quad (4.2.7)$$

We emphasize the key role of the constraint qualification in this result. Lagrange multiplier result of Lemma 4.2.1 implies that if $g(x^*)$ cannot be written as a linear combination of the constraint gradients, then there exists a nonzero p in $\text{null}(J(x^*))$ such that $g(x^*)^T p < 0$. If the constraint qualification holds, this p is a tangent to a feasible path along which f takes on lower values.

The vector $Z(x)^T g(x)$ is known as the *reduced gradient* of f at x . The first-order condition $Z(x^*)^T g(x^*) = 0$ is analogous to the first-order condition $g(x^*) = 0$ in the unconstrained case.

4.2.4. The Lagrangian

Condition (4.2.6) can also be written as

$$g(x^*) - J(x^*)^T \lambda^* = 0,$$

which can be interpreted as a statement that x^* is a *stationary point* (with respect to x) of the *Lagrangian*

$$L(x, \lambda) \equiv f(x) - \lambda^T c(x) \quad (4.2.8)$$

when $\lambda = \lambda^*$. Note that λ^* is a stationary point of $L(x, \lambda)$ (with respect to λ) at any feasible point x .

We emphasize that in general a solution x^* of NEP is *not* an unconstrained minimizer of the Lagrangian. For example, consider the one-variable problem

$$\underset{x \in \mathbb{R}^1}{\text{minimize}} \quad x^3 \quad \text{subject to} \quad x + 1 = 0, \quad (4.2.9)$$

whose (unique) solution is obviously $x^* = -1$, with Lagrange multiplier $\lambda^* = 3$. However, x^* is an unconstrained *maximizer* of the associated Lagrangian $L(x, \lambda^*) = x^3 - 3(x + 1)$. Fig. 4.2 depicts the level curves of the Lagrangian for this problem.

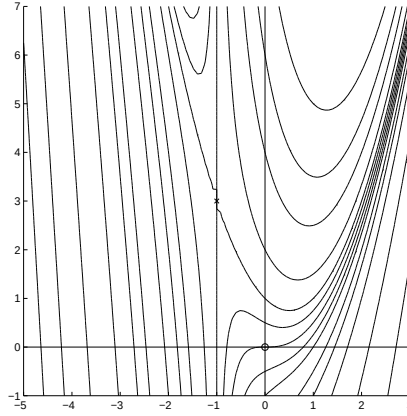


Figure 4.2: Level curves of the Lagrangian $x^3 - \lambda(x + 1)$ for the problem $\min x^3$ subject to $x + 1 = 0$. An “x” marks the position of the stationary point $(x^*, \lambda^*) = (-1, 3)$.

To derive second-order optimality conditions, we examine the second-order behavior of f along feasible paths emanating from x^* . The Hessian of the Lagrangian with respect to x plays a crucial role in the second-order optimality conditions. At the point (x, λ) , the Hessian of the Lagrangian is given by

$$H(x, \lambda) \equiv \nabla_{xx}^2 L(x, \lambda) = H(x) - \sum_{i=1}^m \lambda_i \nabla^2 c_i(x). \quad (4.2.10)$$

Theorem 4.2.4. (Second-order necessary conditions for a local solution of NEP.) *If the constraint qualification holds at x^* , then x^* is a local solution of NEP only if:*

- (a) x^* is feasible, i.e., $c(x^*) = 0$;
- (b) there exists a vector λ^* such that $g(x^*) = J(x^*)^T \lambda^*$; and
- (c) for the λ^* of part (a), $p^T H(x^*, \lambda^*) p \geq 0$ for every vector p satisfying $J(x^*) p = 0$.

Proof. Assume that x^* is a local solution of NEP. Part (b) holds by Theorem 4.2.3. Now consider a nonzero vector p satisfying $J(x^*) p = 0$. (If none exists, the theorem holds vacuously.) Let $x^*(\alpha)$ be the twice-differentiable path whose existence is guaranteed by the constraint qualification, where $x^*(0) = x^*$ and $dx^*(\alpha)/d\alpha|_{\alpha=0} = p$. Let v denote $d^2 x^*(\alpha)/d\alpha^2|_{\alpha=0}$, and assume henceforth that all vector and matrix functions are evaluated at x^* unless otherwise specified. Since each constraint function c_i is identically zero along $x^*(\alpha)$, we have

$$\begin{aligned} \left. \frac{d^2}{d\alpha^2} c_i(x^*(\alpha)) \right|_{\alpha=0} &= \nabla c_i(x^*(\alpha))^T \left. \frac{d^2}{d\alpha^2} x^*(\alpha) \right|_{\alpha=0} + \frac{d}{d\alpha} (\nabla c_i(x^*(\alpha))^T) \left. \frac{d}{d\alpha} x^*(\alpha) \right|_{\alpha=0} \\ &= \nabla c_i(x^*)^T v + p^T \nabla^2 c_i(x^*) p = 0. \end{aligned} \quad (4.2.11)$$

Further, using (4.2.6) and the assumption that $J(x^*)p = 0$,

$$\begin{aligned} \left. \frac{d}{d\alpha} f(x^*(\alpha)) \right|_{\alpha=0} &= g(x^*(\alpha))^T \left. \frac{d}{d\alpha} x^*(\alpha) \right|_{\alpha=0} = g(x^*)^T p \\ &= \lambda^{*T} J(x^*)p = 0. \end{aligned} \quad (4.2.12)$$

Condition (b) implies that x^* is a stationary point of f along the feasible path. In order for x^* to be a local solution, the *curvature* of f along any feasible path must be nonnegative, i.e., it must hold that

$$\left. \frac{d^2}{d\alpha^2} f(x^*(\alpha)) \right|_{\alpha=0} \geq 0. \quad (4.2.13)$$

Using (4.2.12), the definition of v , and (4.2.6), we write (4.2.13) as

$$\begin{aligned} \left. \frac{d^2}{d\alpha^2} f(x^*(\alpha)) \right|_{\alpha=0} &= \left. \frac{d}{d\alpha} \left(g(x^*(\alpha))^T \frac{d}{d\alpha} x^*(\alpha) \right) \right|_{\alpha=0} \\ &= g(x^*)^T \left. \frac{d^2}{d\alpha^2} x^*(\alpha) \right|_{\alpha=0} + p^T H(x^*)p \\ &= \lambda^{*T} J(x^*)v + p^T H(x^*)p \geq 0. \end{aligned} \quad (4.2.14)$$

Rewriting (4.2.11) as $\nabla c_i(x^*)^T v = -p^T H_i(x^*)p$ and substituting this expression into (4.2.14), we obtain

$$\begin{aligned} \left. \frac{d^2}{d\alpha^2} f(x^*(\alpha)) \right|_{\alpha=0} &= -p^T \left(\sum_{i=1}^m \lambda_i^* H_i(x^*) \right) p + p^T H(x^*)p \\ &= p^T H(x^*, \lambda^*)p \geq 0, \end{aligned}$$

which is the desired result. ■

A compact statement of condition (c) of this theorem is that the $(n-m) \times (n-m)$ matrix $Z(x^*)^T H(x^*, \lambda^*) Z(x^*)$ must be positive semi-definite. Given arbitrary x and λ , the matrix $Z(x)^T H(x, \lambda) Z(x)$ is called the *reduced Hessian* of the Lagrangian. The condition that $Z(x^*)^T H(x^*, \lambda^*) Z(x^*)$ is positive semi-definite is analogous to condition that $H(x^*)$ be positive semi-definite in the unconstrained case.

Fig. 4.3 depicts the level curves of the Lagrangian $L(x, \lambda) = -x_1 x_2^2 - \lambda(2 - x_1^2 - x_2^2)$ for problem (4.1.1) with λ fixed at the optimal Lagrange multiplier $\lambda^* = 0.8165$. The stationary point of the Lagrangian lies at $x^* = (0.8165, 1.1547)$, which is a minimizer of the problem. Starting at x^* , the Lagrangian initially has positive curvature along the constraint surface, and negative curvature normal to the constraint surface. Note that the Lagrangian is unbounded below, and therefore has no minimizer.

The next theorem gives *sufficient conditions* for a point x^* to be a solution of NEP.

Theorem 4.2.5. (*Sufficient conditions for a local solution of NEP.*) *A point x^* is a strict local minimizer of NEP if*

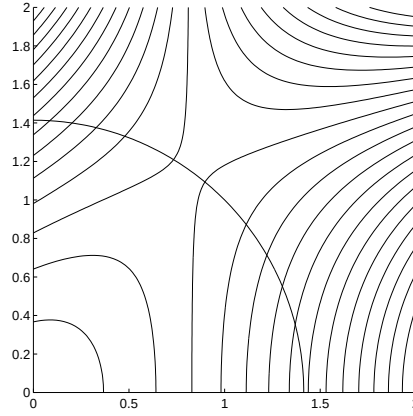


Figure 4.3: This graph depicts the level curves of the Lagrangian $L(x, \lambda^*)$ for the two-dimensional example (4.1.1). The stationary point of the Lagrangian lies at $x^* = (0.8165, 1.1547)$, which is a minimizer of the problem. Starting at x^* , the Lagrangian initially has positive curvature along the constraint surface, and negative curvature normal to the constraint surface.

- (a) x^* is feasible, i.e., $c(x^*) = 0$;
- (b) there exists a vector λ^* such that $g(x^*) = J(x^*)^T \lambda^*$; and
- (c) for the λ^* of part (b), the strict inequality $p^T H(x^*, \lambda^*) p > 0$ holds for all nonzero p such that $J(x^*) p = 0$.

Proof. Consider any feasible sequence $\{x_k\}$ with $x_k \rightarrow x^*$. Let p_k be a unit vector such that $x_k = x^* + \alpha_k p_k$. It follows that $\alpha_k \rightarrow 0$. The sequence $\{p_k\}$ is bounded and must have a convergent subsequence. Without loss of generality, this subsequence and its limit will be denoted by $\{p_k\}$ and p^* . The Taylor-series expansion of c yields

$$c(x_k) = c(x^*) + \alpha_k J(x^*) p_k + \alpha_k \int_0^1 (J(x^* + \xi \alpha_k p_k) - J(x^*)) p_k d\xi.$$

Since $c(x_k) = c(x^*) = 0$, we can write

$$J(x^*) p_k = - \int_0^1 (J(x^* + \xi \alpha_k p_k) - J(x^*)) p_k d\xi.$$

Taking the limit as $k \rightarrow \infty$, it follows that $p_k \rightarrow p^*$, $\alpha_k \rightarrow 0$, and the continuity of J implies that $J(x^*) p^* = 0$.

Now assume that x^* is not a strict minimizer in B , then there must exist a feasible sequence $\{x_k\}$ such that $x_k \rightarrow x^*$ and $f(x^*) \geq f(x_k)$ for sufficiently large k . Expanding the objective function in a Taylor series gives

$$0 \geq f(x_k) - f(x^*) = \alpha_k g(x_k)^T p_k + \alpha_k^2 \int_0^1 p_k^T H(x^* + \xi \alpha_k p_k) p_k (1 - \xi) d\xi. \quad (4.2.15)$$

Similarly, for each constraint function $c_i(x_k)$,

$$0 = \alpha_k \nabla c_i(x^*)^T p_k + \alpha_k^2 \int_0^1 p_k^T H_i(x^* + \xi \alpha_k p_k) p_k (1 - \xi) d\xi.$$

Multiplying each of these identities by λ_i^* , subtracting the resulting product from (4.2.15) and using the condition $g(x^*) = \sum_i \nabla c_i(x^*) \lambda_i^*$ to simplify the final sum gives

$$0 \geq \alpha_k^2 \int_0^1 p_k^T H(x^* + \xi \alpha_k p_k, \lambda^*) p_k (1 - \xi) d\xi,$$

which yields a contradiction as $k \rightarrow \infty$. ■

Note that the sufficient conditions do not require the satisfaction of the constraint qualification. Theorem 4.2.5 is weaker than its unconstrained counterpart Theorem 3.2.6 in the sense that the sufficient conditions imply that x^* is a *strict* local minimizer and not necessarily an isolated local minimizer. We may assert that x^* is isolated only under the additional assumption that the constraint gradients are linearly independent. This is shown in the next result

Theorem 4.2.6. (*Sufficient conditions for an isolated local solution of NEP.*) A point x^* is an isolated local minimizer of NEP if

- (a) x^* is feasible, i.e., $c(x^*) = 0$;
- (b) there exists a vector λ^* such that $g(x^*) = J(x^*)^T \lambda^*$; and
- (c) for the λ^* of part (b), the strict inequality $p^T H(x^*, \lambda^*) p > 0$ holds for all nonzero p such that $J(x^*) p = 0$.
- (d) the constraint gradients at x^* are linearly independent.

Proof. From Theorem 4.2.5, conditions (a)–(c) imply that x^* is a strict local minimizer. It remains to show that x^* is locally unique. Conditions (a)–(b) state that (x^*, λ^*) is a zero of the $(n+m)$ -vector $G(x, \lambda) = \nabla L(x, \lambda)$. The implicit function theorem can be used to establish local uniqueness if the Jacobian of G is nonsingular. Differentiating G with respect to x and λ gives $G'(x, \lambda)$ as

$$G'(x, \lambda) = \begin{pmatrix} H(x) - \sum_{i=1}^m \lambda_i H_i(x) & -J(x)^T \\ -J(x) & 0 \end{pmatrix} = \begin{pmatrix} H(x, \lambda) & -J(x)^T \\ -J(x) & 0 \end{pmatrix},$$

where $H(x)$, $H_i(x)$ and $H(x, \lambda)$ denote $\nabla^2 f(x)$, $\nabla^2 c_i(x)$ and the Hessian of the Lagrangian respectively. For simplicity, we write $H \equiv H(x, \lambda)$ and $J \equiv J(x)$.

Suppose $G(x^*, \lambda^*)$ is singular. Then there must exist vectors u and v , not both zero, such that

$$\begin{pmatrix} H & -J^T \\ -J & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} Hu - J^T v \\ -Ju \end{pmatrix} = 0.$$

If $u \neq 0$ and $v = 0$, then u is a nonzero vector such that $Hu = 0$ and $Ju = 0$, contradicting assumption (c). Alternatively, if $u = 0$ and $v \neq 0$, then v is a nonzero vector such

that $J^T v = 0$, contradicting the assumption that the rows of J are linearly independent. Finally, if $u \neq 0$ and $v \neq 0$, then the equations $Ju = 0$ imply that $u \in \text{null}(J)$, and u can be written as $u = Zw$ for some matrix Z whose columns form a basis for $\text{null}(J)$. The equations $Hu - J^T v = 0$ imply that $HZw = J^T v$, or, equivalently, $Z^T H Z w = Z^T J^T v = (JZ)^T v = 0$, which implies that the reduced Hessian is singular, again contradicting assumption (c). ■

4.3. Quadratic Problems with Equality Constraints

The minimization of a quadratic function subject to a set of linear constraints is termed a *quadratic programming* (QP) problem. Quadratic programming is important not only in its own right, but also because almost all effective methods for linearly and nonlinearly constrained minimization involve the solution of a quadratic program. In this section we discuss the solution of quadratic programs with only *equality* constraints, of the following form

$$\begin{array}{ll} \text{EQP} & \begin{array}{l} \text{minimize} \quad f(x) = d^T x + \frac{1}{2} x^T H x \\ \text{subject to} \quad Ax = b, \end{array} \end{array}$$

where H is symmetric and A has m rows. Note that the gradient $g(x) = \nabla f(x)$ of the quadratic objective function at x is $d + Hx$, and that the Hessian is the constant matrix H .

In order to make the problem sensible, we assume that the set of feasible points $\mathbb{F} = \{x \mid c(x) = Ax - b = 0\}$ is non-empty. This condition is equivalent to the condition that the linear system $Ax = b$ has at least one solution, which is equivalent to the condition that b is a vector that lies in $\text{range}(A)$.

If $\mathbb{F}$ is non-empty and A has rank r , with $r < m$, it can be shown that $\mathbb{F}$ is completely defined by any set of r linearly independent rows of A . It follows that there is no loss of generality in assuming that A has full row rank (so that $m \leq n$). It must be emphasized that this is a property that holds only for *linear* constraints.

4.3.1. Optimality conditions for EQP

The following theorem, which is essentially Theorem 4.2.4 applied to problem EQP, will allow us to derive numerically verifiable conditions for x^* to be optimal for EQP.

Theorem 4.3.1. (*Necessary conditions for a solution of EQP.*) Let Z denote a basis for $\text{null}(A)$. Then a feasible point x^* is a solution of EQP only if

- (a) $Z^T g(x^*) = 0$, or equivalently $g(x^*) = A^T \lambda^*$; for some λ^* ; and
- (b) $Z^T H Z$ is positive semi-definite. ■

The vector $Z^T g(x)$ is known as the *reduced gradient* of f at x . Any point $\bar{x}$ such that $Z^T g(\bar{x}) = 0$ is termed a *constrained stationary point* of f (with respect to the constraints $Ax = b$). The matrix $Z^T H Z$ is known as the *reduced Hessian*.

It is important to note that unlike the unconstrained case, it is *not* necessary for the Hessian H to be positive semidefinite at x^* . The following example illustrates this point.

Consider minimizing $f(x) = x_1^2 + x_2^2 - 3x_1x_2 + x_1 + x_2$ subject to the single constraint $x_1 + 2x_2 = 2$. Here

$$d = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{and} \quad H = \begin{pmatrix} 2 & -3 \\ -3 & 2 \end{pmatrix}.$$

The constraint matrix is just the single row $A = (1 \ 2)$, with right-hand side $b = (2)$. The Hessian is indefinite, with eigenvalues 5 and -1 . One possible basis for the null space is the column vector $Z = (2 \ -1)^T$. At the point $x^* = (\frac{7}{22} \ \frac{15}{22})^T$ we have

$$Z^T g(x^*) = (2 \ -1) \begin{pmatrix} \frac{5}{22} \\ \frac{5}{11} \end{pmatrix} = 0.$$

Although the Hessian is indefinite, the reduced Hessian is positive definite, since

$$Z^T H Z = (2 \ -1) \begin{pmatrix} 2 & -3 \\ -3 & 2 \end{pmatrix} \begin{pmatrix} 2 \\ -1 \end{pmatrix} = 22 > 0.$$

Note that the single Lagrange multiplier is defined by the overdetermined equations $A^T \lambda^* = g(x^*)$, or equivalently,

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} \lambda^* = \begin{pmatrix} \frac{5}{22} \\ \frac{5}{11} \end{pmatrix}, \quad \text{from which we obtain} \quad \lambda^* = \frac{5}{22}.$$

The imposition of m linearly independent linear constraints on an n -dimensional problem can be viewed as reducing the dimensionality of the optimization to $n - m$. This reduction in dimensionality can be described formally in terms of the complementary subspaces $\text{range}(A^T)$ and $\text{null}(A)$, with bases Y and Z respectively. Every n -vector x has a unique decomposition

$$x = Yx_Y + Zx_Z. \quad (4.3.1)$$

Consider any feasible point x ; then, since $AZ = 0$,

$$Ax = A(Yx_Y + Zx_Z) = AYx_Y = b. \quad (4.3.2)$$

Because AY is nonsingular, (4.3.2) shows that the linear constraints $Ax = b$ *uniquely determine* the vector x_Y . If we denote the fixed vector Yx_Y as x_0 , every feasible x is of the form

$$x = x_0 + Zx_Z,$$

so that EQP can be viewed as an *unconstrained* problem in the $n - m$ variables x_Z . This gives precisely the expected reduction in dimensionality to $n - m$. For example, consider the single constraint

$$x_1 + x_2 + x_3 + x_4 = 4,$$

for which $A = (1 \ 1 \ 1 \ 1)$. The matrix Y may be taken as A^T , and one possible Z is

$$Z = \begin{pmatrix} -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{5}{6} & -\frac{1}{6} & -\frac{1}{6} \\ -\frac{1}{6} & \frac{5}{6} & -\frac{1}{6} \\ -\frac{1}{6} & -\frac{1}{6} & \frac{5}{6} \end{pmatrix}. \quad (4.3.3)$$

Substituting into (4.3.2), we obtain $4x_y^* = 4$, or $x_y^* = 1$. Therefore, regardless of the objective function, the solution must be of the form

$$x^* = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} + Zx_z^*,$$

where Z is given by (4.3.3) and $x_z^* \in \mathbb{R}^3$.

Since x_y is completely determined by the constraints, we substitute (4.3.1) for x in EQP and ignore constant terms to give an *unconstrained* problem in x_z :

$$\underset{x_z \in \mathbb{R}^{n-m}}{\text{minimize}} \quad x_z^T Z^T (d + HYx_y^*) + \frac{1}{2} x_z^T Z^T H Z x_z.$$

Theorem 4.3.1 shows that, if $Z^T H Z$ is positive semi-definite, x_z^* must satisfy the equations

$$Z^T H Z x_z^* = -Z^T (d + HYx_y^*) = -Z^T \nabla f(x_0). \quad (4.3.4)$$

The above approach, together with the results for the unconstrained case, imply that optimality conditions for an EQP are completely characterized (i.e., there is no “hard case” analogous to both the first and second derivative being zero). The proof of the following lemma follows directly from Theorems 3.2.1 and 4.3.1.

Theorem 4.3.2. *Let x_0 be any feasible point for the constraints $Ax = b$, where A be an $m \times n$ matrix of rank m . Let Z denote a matrix whose columns form a basis for $\text{null}(A)$. If $Z^T H Z$ is positive semi-definite, then the quadratic function $f(x)$ achieves its bounded global minimum at a point x^* such that*

$$x^* = x_0 + Z(Z^T H Z)^{-1} Z^T \nabla f(x_0).$$

If $Z^T H Z$ is positive definite, the minimizer x^* is unique. If $Z^T H Z$ is positive semi-definite and singular, there is an infinite number of minimizers of the form $x^* + Zv$, where $v \in \text{null}(Z^T H Z)$. If $Z^T \nabla f(x_0) \notin \text{range}(Z^T H Z)$, or if $Z^T H Z$ is indefinite, then $f(x)$ is unbounded below. ■

When the reduced Hessian $Z^T H Z$ cannot be computed, an alternative to Theorem 4.3.2 is needed to characterize the solution of EQP. To obtain such a result, we use Theorem 4.3.2 to give the optimality and feasibility conditions for EQP as

$$g(x^*) = d + Hx^* = A^T \lambda^*, \quad (4.3.5a)$$

$$Ax^* = b, \quad (4.3.5b)$$

where λ^* is the vector of Lagrange multipliers. Rewriting (4.3.5) in matrix form, we note that x^* and λ^* solve the symmetric indefinite linear system:

$$\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x^* \\ -\lambda^* \end{pmatrix} \equiv K \begin{pmatrix} x^* \\ -\lambda^* \end{pmatrix} = \begin{pmatrix} -d \\ b \end{pmatrix}. \quad (4.3.6)$$

4.3.2. Solving an equality constrained QP

We have already seen that a quadratic program with equality constraints may be stated as:

$$\begin{array}{ll} \text{EQP} & \underset{x}{\text{minimize}} \quad f(x) = d^T x + \frac{1}{2} x^T H x \\ & \text{subject to} \quad Ax = b, \end{array}$$

where the Hessian H is symmetric, and A is an $m \times n$ matrix with rank m . The gradient $\nabla f(x)$ is the linear function $g(x) = d + Hx$.

First, assume that H is positive definite, in which case EQP is a convex problem with a unique solution. The feasibility and optimality conditions for EQP may be written as two linear relations, repeated here for convenience:

$$\begin{aligned} d + Hx^* &= A^T \lambda^* \\ Ax^* &= b, \end{aligned} \tag{4.3.7}$$

where x^* is a solution and λ^* is the Lagrange multiplier vector. These equations may be rearranged so that x^* is defined as the step p^* from the given point x_0 . Writing $x^* = x_0 + p^*$, (4.3.7) becomes

$$\begin{aligned} Ap^* &= -c, \quad \text{where } c = Ax_0 - b; \\ g + Hp^* &= A^T \lambda^*, \quad \text{with } g = d + Hx_0. \end{aligned}$$

These conditions imply the following system of equations for p^* and λ^* :

$$\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} p^* \\ -\lambda^* \end{pmatrix} = - \begin{pmatrix} g \\ c \end{pmatrix}. \tag{4.3.8}$$

If x_0 is *feasible*, then $c = Ax_0 - b = 0$ and the second block of the right-hand side of the KKT system (4.3.8) is zero.

Linear systems of the type (4.3.8) are of fundamental importance in nonlinear programming. A system with this structure is called a *Karush-Kuhn-Tucker (or just KKT) system*. The matrix K such that

$$K = \begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix}$$

is known as the *KKT matrix*. If A has full rank and the reduced Hessian is nonsingular, then K is nonsingular.

The vectors p^* and λ^* are themselves the solution of a quadratic program

$$\begin{array}{ll} \underset{p \in \mathbb{R}^n}{\text{minimize}} & \varphi(p) \equiv g^T p + \frac{1}{2} p^T H p \\ \text{subject to} & Ap = -c. \end{array} \tag{4.3.9}$$

The quadratic function $\varphi(p)$ defines the *change* in the QP objective corresponding to a step p ; i.e., $\varphi(p) = f(x_0 + p) - f(x_0)$, which implies that *minimizing* $\varphi(p)$ in (4.3.9) is equivalent to *maximizing* $f(x_0) - f(x_0 + p)$, the reduction in f .

Finally, we could just have well used an initial estimate λ_0 for the Lagrange multipliers and solved for the change $s^* = \lambda^* - \lambda_0$. In this case, the optimality conditions are

$$\begin{aligned} Ap^* &= -c, & \text{where } c &= Ax_0 - b; \\ g - A^T \lambda_0 + Hp^* &= A^T s^*, & \text{with } g &= d + Hx_0. \end{aligned}$$

These conditions give analogous equations

$$\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} p^* \\ -s^* \end{pmatrix} = - \begin{pmatrix} g - A^T \lambda_0 \\ c \end{pmatrix}. \quad (4.3.10)$$

We now present three alternative representations of p^* and λ^* whose appropriateness for computation depends on the nature of A and H . Nonetheless, it is important to remember that they are all *equivalent* under certain conditions (for example, when H is positive definite).

4.4. The Method of Multipliers

Now we return to the general *nonlinear* equality constrained problem of the form

$$\begin{aligned} \text{NEP} \quad & \underset{x \in \mathbb{R}^n}{\text{minimize}} && f(x) \\ & \text{subject to} && c(x) = 0. \end{aligned}$$

In the unconstrained case, a standard approach is to use the first-order optimality conditions to define a system of nonlinear equations $g(x) = 0$ whose solution is a first-order optimal point x^* . In the constrained case, the relevant system involves the gradient of the Lagrangian, which expresses the first-order feasibility and optimality conditions satisfied by x^* and λ^* . Theorem 4.2.3 implies that the $n + m$ vector (x^*, λ^*) is a solution of the $n + m$ nonlinear equations $F(x, \lambda) = 0$, where

$$F(x, \lambda) \equiv \nabla L(x, \lambda) = \begin{pmatrix} g(x) - J(x)^T \lambda \\ c(x) \end{pmatrix}. \quad (4.4.1)$$

It follows that one approach to finding (x^*, λ^*) is to apply Newton's method to find a zero of F . Let y denote an $(n + m)$ -vector (x, λ) . Given an initial estimate $y_0 = (x_0, \lambda_0)$ of a zero $y^* = (x^*, \lambda^*)$, Newton's method generates a sequence of iterates y_k such that

$$y_{k+1} = y_k + \alpha_k \Delta y_k, \quad \text{where } F'(y_k) \Delta y_k = -F(y_k).$$

It only remains to compute the $(n + m) \times (n + m)$ Jacobian $F'(y)$. Differentiating (4.4.1) with respect to x and λ gives $F'(y) \equiv F'(x, \lambda)$ as

$$F'(x, \lambda) = \begin{pmatrix} H(x) - \sum_{i=1}^m \lambda_i H_i(x) & -J(x)^T \\ J(x) & 0 \end{pmatrix}.$$

where $H(x)$ and $H_i(x)$ denote $\nabla^2 f(x)$ and $\nabla^2 c_i(x)$, the objective Hessian and Hessian of the i th constraint function. Using this Jacobian in the Newton equations gives

$$\begin{pmatrix} H(x_k, \lambda_k) & -J(x_k)^T \\ J(x_k) & 0 \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta \lambda_k \end{pmatrix} = - \begin{pmatrix} g(x_k, \lambda_k) \\ c(x_k) \end{pmatrix},$$

where $g(x, \lambda)$ and $H(x, \lambda)$ denote the gradient and Hessian of the Lagrangian with respect to x , i.e.,

$$g(x, \lambda) = g(x) - J(x)^T \lambda \quad \text{and} \quad H(x, \lambda) = H(x) - \sum_{i=1}^m \lambda_i H_i(x).$$

The Newton system has two equivalent forms that involve a *symmetric* matrix analogous to the KKT matrix. First, using suitable row and column scalings of ± 1 we obtain the equivalent symmetric system,

$$\begin{pmatrix} H(x_k, \lambda_k) & J(x_k)^T \\ J(x_k) & 0 \end{pmatrix} \begin{pmatrix} \Delta x_k \\ -\Delta \lambda_k \end{pmatrix} = - \begin{pmatrix} g(x_k, \lambda_k) \\ c(x_k) \end{pmatrix} \quad (4.4.2)$$

Note the similarity of this system to the KKT system (4.3.10). It follows that $(\Delta x_k, \Delta \lambda_k)$ may be computed using any of the methods discussed in §4.3 for solving a KKT system. The matrices in (4.4.2) are nonsingular whenever $J(x_k)$ has independent rows and the reduced Hessian $Z(x_k)^T H(x_k, \lambda_k) Z(x_k)$ is nonsingular.

Another equivalent system is obtained by substituting $\eta_k = \lambda_k + \Delta \lambda_k$. In this case, only the right-hand side changes, and the relevant system is

$$\begin{pmatrix} H(x_k, \lambda_k) & J(x_k)^T \\ J(x_k) & 0 \end{pmatrix} \begin{pmatrix} \Delta x_k \\ -\eta_k \end{pmatrix} = - \begin{pmatrix} g(x_k) \\ c(x_k) \end{pmatrix}.$$

Note that this system is similar to the KKT system (4.3.8).

Once Δy_k has been computed, a line search can be used to ensure that an appropriate “merit” or “line search” function is sufficiently reduced at each step. Two common choices are $\|F(x, \lambda)\|_2$ and $\frac{1}{2}\|F(x, \lambda)\|_2^2$. In the latter case, we have the line search function

$$\frac{1}{2}\|g(x) - J(x)^T \lambda\|^2 + \frac{1}{2}\|c(x)\|_2^2,$$

which simultaneously enforces both *feasibility* (by ensuring that $\|c(x)\| \rightarrow 0$) and *optimality* (by forcing $g(x)$ to lie in the range space of $J(x)^T$).

Algorithm 4.4.1 defines an iteration based on the method of multipliers.

The standard analysis of Newton’s method can be used to show that if c and f are sufficiently smooth, and if the KKT matrix is nonsingular, then the sequence $\{y_k\}$ will converge at a q -quadratic rate. The nonsingularity condition holds when second-order sufficient conditions for optimality hold, i.e., the reduced Hessian is positive definite and the smallest singular value of the Jacobian is greater than zero at x^* .

This iteration has an important property when the constraints are linear; i.e., when $c(x) = Ax - b$. In this case, once a unit step $\alpha_k = 1$ is taken, then *every subsequent iterate is feasible*, i.e., $Ax_{k+l} = b$ for all l such that $l \geq 1$. The feasibility of x_{k+1} follows from the equations

$$Ax_{k+1} = A(x_k + \Delta x_k) = Ax_k - c(x_k) = Ax_k - Ax_k + b = b.$$

Thus $c(x_{k+1}) = 0$ and the last m rows of (4.4.2) at the next iteration force Δx_{k+1} to lie entirely in the null space of A . This implies that, for any α ,

$$A(x_{k+1} + \alpha \Delta x_{k+1}) = Ax_{k+1} + \alpha A \Delta x_{k+1} = Ax_{k+1} = b,$$

Algorithm 4.4.1. Lagrangian-based method for NEP.

Fix μ such that $0 < \mu < 1$;

Choose x_0 and λ_0 ;

$x \leftarrow x_0$; $\lambda \leftarrow \lambda_0$;

$k \leftarrow 0$;

while not converged do

 Solve the KKT system

$$\begin{pmatrix} H(x, \lambda) & J(x)^T \\ J(x) & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ -\Delta \lambda \end{pmatrix} = - \begin{pmatrix} g(x, \lambda) \\ c(x) \end{pmatrix}$$

$\alpha \leftarrow 1$;

while $\|g(x + \alpha \Delta x, \lambda + \alpha \Delta \lambda)\| > (1 - \alpha \mu) \|g(x, \lambda)\|$ **do**

$\alpha \leftarrow \alpha/2$;

end do

$x \leftarrow x + \alpha \Delta x$; $\lambda \leftarrow \lambda + \alpha \Delta \lambda$;

$k \leftarrow k + 1$;

end do

and x_{k+2} is feasible. This argument may be continued to show that x_{k+l} is feasible for all l such that $l \geq 1$.

Once x_k is feasible, $\|\nabla L(x_k, \lambda_k)\| = \|g(x_k) - A^T \lambda_k\|$, and the line search attempts to find a point for which the objective gradient is “closest” to $\text{range}(A^T)$.

4.5. Linear Equality Constraints

It has been shown that linear constraints have the special property that once a feasible point is computed, all subsequent search directions satisfy the equation $Ap_k = 0$ and lie in $\text{null}(A)$. Linear constraints have the important property that the feasible region $\mathbb{F}$ is *convex*, i.e., for any two points $x, y \in \mathbb{F}$, and for all $0 \leq \theta \leq 1$, the vector z such that $z = (1 - \theta)x + \theta y$ is also in $\mathbb{F}$. This is the same as saying that, given any $x, y \in \mathbb{F}$,

$$z = x + \theta(y - x) \in \mathbb{F} \quad \text{for all } \theta \in [0, 1].$$

This form clearly shows that z is the result of a step θ along a direction $p = y - x$ emanating from the feasible point x . The property of convexity ensures that if x_k and x_{k+1} are feasible, then all points of the form $x_k + \alpha_k p_k$ are also feasible.

In this section we consider minimizing a general nonlinear function in the special case of linear constraints. An important feature of the methods to be considered is that they combine techniques from *unconstrained optimization* and linear algebra. The problem to be addressed has the form

$$\begin{array}{ll} \text{LEP} & \underset{x}{\text{minimize}} \quad f(x) \\ & \text{subject to} \quad Ax = b, \end{array}$$

where f is a real-valued function defined on $\mathbb{R}^n$. As in the unconstrained case, we usually assume that f is at least twice-continuously differentiable.

4.5.1. A local quadratic model

If x_0 is feasible, $c(x_k) = Ax_k - b = 0$ for all $k \geq 0$ and the method of multipliers is defined by the system

$$\begin{pmatrix} H(x_k) & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} p_k \\ -\eta_k \end{pmatrix} = - \begin{pmatrix} g(x_k) \\ 0 \end{pmatrix}. \quad (4.5.1)$$

The discussion of §4.3 implies that these equations define the necessary conditions for the solution of an equality constrained QP in the n variables p . This problem has the form

$$\begin{aligned} & \underset{p \in \mathbb{R}^n}{\text{minimize}} && g(x_k)^T p + \frac{1}{2} p^T H(x_k) p \\ & \text{subject to} && Ap = 0. \end{aligned} \quad (4.5.2)$$

It follows that the Newton step can be interpreted as the solution of a subproblem in which a local quadratic model of f is minimized subject to the requirement that the solution must be feasible. By analogy with the unconstrained case, the local quadratic model of f is suitable for minimization only when its reduced Hessian is positive definite.

An important feature of this approach is that the step length α_k can be defined using the natural merit function $f(x)$ instead of the norm of the Lagrangian gradient. With this choice of merit function, the Lagrangian is not used directly and the Lagrange multipliers need not be updated with x_k . This observation reflects the fact that when x satisfies $Ax = b$, the Lagrangian satisfies $L(x, \lambda) = f(x) - \lambda^T(Ax - b) = f(x)$ for all x , independent of the value of λ .

Methods of this type are known as *null-space methods* because all movement occurs in the null space of A . Null-space methods require the definition of the null-space basis Z . This is done either explicitly using the TQ factorization, or implicitly using the variable-reduction form of Z . The first stage of a null-space method requires the calculation of an initial point x_0 such that $Ax_0 = b$. Any point x can be converted into a feasible point using the calculation

$$x_0 = x + Yp_Y, \quad \text{where } p_Y \text{ satisfies } AYp_Y = -c(x).$$

Each subsequent iteration of a null-space method involves the solution of the quadratic subproblem (4.5.1). With the assumption that $Z^T H(x_k) Z$ is positive definite, the solution of (4.5.1) may be written as $p_k = Zp_Z$, where p_Z satisfies

$$Z^T H(x_k) Z p_Z = -Z^T g(x_k).$$

If $Z^T g(x_k) \neq 0$, then Zp_Z must be a descent direction for f since

$$g(x_k)^T p_k = g(x_k)^T Z p_Z = -g(x_k)^T Z (Z^T H(x_k) Z)^{-1} Z^T g_k < 0.$$

It must be emphasized that the assumption of positive-definiteness of $Z^T H(x_k) Z$ is weaker than the assumption of positive-definiteness of $H(x_k)$.

If $Z^T H(x_k) Z$ is sufficiently positive definite, α_k can be defined that give a sufficient decrease in the objective f , and the analysis of convergence carries over directly from the

unconstrained case. In effect, a null-space method transforms LEP into an *unconstrained* problem in a subspace of lower dimension.

Analogous to the unconstrained case, it is possible to define a general class of algorithms that use a convex quadratic model of the objective function. However, in the case of LEP, the model can be defined in terms of an $(n - m) \times (n - m)$ positive-definite reduced matrix B_z .

Algorithm 4.5.1. Generic Null-Space Line Search Method for LEP.

```

Compute  $Z$  such that  $AZ = 0$ ;
Compute  $x_0$  such that  $Ax_0 = b$ ;
Define an  $(n - m) \times (n - m)$  positive-definite matrix  $B_z$ ;
 $k \leftarrow 0$ ;
repeat
    Define  $p_z$  as the solution of  $B_z p_z = -Z^T g(x_k)$ ;
    Set  $p_k = Z p_z$ ;
    Compute an  $\alpha_k$  that gives a sufficient decrease in  $f$ ;
     $x_{k+1} \leftarrow x_k + \alpha_k p_k$ ;
    Redefine  $B_z$  to reflect the curvature at  $x_{k+1}$ ;
     $k \leftarrow k + 1$ ;
until converged;
```

The matrix Z may be computed using any of the methods discussed in §4.3.

4.5.2. Newton-based methods

A Newton-based algorithm for LEP can be developed if second derivatives of f are available. The *Newton direction* for LEP is the direction that solves (4.5.1) with $H_k = H(x_k)$, and will be a direction of descent for f if the reduced Hessian $Z^T H(x_k) Z$ is positive definite.

If the reduced Hessian is not positive definite, the solution of the EQP subproblem (4.5.1) is either unbounded or not unique. As in the unconstrained case, a Newton-based method can be defined in which p_z is the solution of

$$\bar{H}_z p_z = -g_z, \quad (4.5.3)$$

where $\bar{H}_z$ is a positive-definite matrix related in some way to the reduced Hessian and $g_z = Z^T g(x_k)$. In particular, if the reduced Hessian is sufficiently positive definite, $\bar{H}_z$ can be $Z^T H(x_k) Z$.

We may also apply a trust-region method to LEP. Since all changes in x lie in $\text{null}(A)$, the appropriate trust-region constraint is $\|s_z\| \leq \Delta$, where $s = Z s_z$. The constrained subproblem analogous to (3.6.4) is

$$\begin{aligned} & \underset{s_z \in \mathbb{R}^{n-m}}{\text{minimize}} && \varphi(s_z) \equiv g_z^T s_z + \frac{1}{2} s_z^T B_z s_z \\ & \text{subject to} && \|s_z\|_2 \leq \Delta_k, \end{aligned}$$

where $g_z = Z^T g(x_k)$ and $B_z = Z^T H(x_k) Z$. This subproblem may be solved in the same way as (3.6.4).

Under conditions similar to those in the unconstrained case, a Newton-based line search and trust-region methods will converge to x^* from any initial feasible point. Furthermore, if the reduced Hessian is positive definite in a neighborhood of x^* and the starting point is sufficiently close to x^* , the iterates will converge at a q -quadratic rate.

4.5.3. Discrete Newton methods

When first, but not second, derivatives of f are available, a discrete Newton null-space method can be defined, as in the unconstrained case. The major difference is that only the $(n - m)$ -dimensional *reduced* Hessian needs to be approximated, rather than all of $H(x_k)$.

Expanding f in a Taylor series along a basis vector z_i , we have

$$g(x_k + \delta_i z_i) \approx g(x_k) + \delta_i H(x_k) z_i,$$

which implies that a direct finite-difference approximation to $Z^T H(x_k) Z$ can be obtained by using the $n - m$ columns of Z as finite-difference vectors (as opposed to approximating the full Hessian and forming the reduced matrix). If w_i is defined as

$$w_i = \frac{1}{\delta_i} (g(x_k + \delta_i z_i) - g(x_k)), \quad i = 1, \dots, n - m,$$

for some finite-difference interval δ_i , then w_i is an $O(\delta_i)$ approximation to $H(x_k) z_i$. Let W denote the $n \times (n - m)$ matrix whose i -th column is w_i , so that W may be computed with $n - m$ evaluations of the gradient. A symmetric approximation to $Z^T H(x_k) Z$ is then

$$\tilde{H}_Z \equiv \frac{1}{2} (Z^T W + W^T Z).$$

If $\tilde{H}_Z$ is not positive definite, a related positive-definite matrix $\bar{H}_Z$ may be used in (4.5.3) to define p_Z . The global convergence properties of a discrete Newton null-space method for LEP are identical to those of an exact Newton-based method. However, as in the unconstrained case, the limited accuracy of the reduced Hessian approximation can inhibit the quadratic convergence.

4.6. Penalty Function Methods

The basic principle invoked in solving NEP is that of *replacing a difficult problem by an easier problem*. Application of this principle leads to methods that formulate and solve a *sequence of subproblems*, where each subproblem is related in a known way to the original problem. In some of the methods to be discussed, the subproblem involves the *unconstrained* minimization of a model function; in other instances, the subproblem includes bounds and/or linear constraints derived from constraints of the original problem.

For constrained problems, the same general principle applies of creating subproblems based on local models of the objective and constraint functions. Approaches to solving NEP vary in the form of the subproblem to be solved at each iteration, and in the definition of any model functions. Further variation is induced by the use of different numerical methods for solving mathematically equivalent problems.

Although penalty functions in their original form are not widely used today, a thorough understanding of their properties is important background for more recent methods.

Penalty functions have a long history, occur in many forms, and are often called by special names in different applications. Their motivation is always to ensure that the iterates do not deviate “too far” from the constraints.

In general, an unconstrained minimizer of f will occur at an infeasible point, or f may be unbounded below. Therefore, any algorithm for NEP must consider not only the minimization of f , but also the enforcement of feasibility. In solving NEP, a “natural” strategy is to devise a *composite function* whose *unconstrained* minimizer is either x^* itself, or is related to x^* in a known way. The original problem can then be solved by formulating a sequence of unconstrained subproblems (or possibly a single unconstrained subproblem). Intuitively, this can be achieved by minimizing a function that combines f and a term that “penalizes” constraint violations.

4.6.1. The quadratic penalty function

A “classical” penalty term is the squared Euclidean norm of the constraint violations, which leads to the well known *quadratic penalty function*

$$P_2(x, \rho) \equiv f(x) + \frac{\rho}{2} \sum_{i=1}^m c_i(x)^2 = f(x) + \frac{\rho}{2} c(x)^T c(x) = f(x) + \frac{\rho}{2} \|c(x)\|_2^2, \quad (4.6.1)$$

where the nonnegative scalar ρ is called the *penalty parameter*. In the “classical” penalty function method, $P_2(x, \rho)$ is minimized for each of a sequence of increasing values of ρ using an unconstrained minimization method. Let $x(\rho)$ denote an unconstrained minimizer of $P_2(x, \rho)$. The main property idea is that, for a given sequence $\{\rho_k\}$, we have

$$\lim_{k \rightarrow \infty} x(\rho_k) = x^*.$$

In practice, a finite sequence of unconstrained problems are solved, with the approximate minimizer of $P_2(x, \rho_k)$ being used as the initial estimate of the minimizer of $P_2(x, \rho_{k+1})$.

To illustrate this procedure, consider the problem

$$\underset{x \in \mathbb{R}^1}{\text{minimize}} \quad x^2 \quad \text{subject to} \quad x - 1 = 0,$$

whose solution is trivially $x^* = 1$. The quadratic penalty function is given by

$$P_2(x, \rho) = x^2 + \frac{\rho}{2}(x - 1)^2.$$

This function is twice-continuously differentiable, with $\nabla P_2(x, \rho) = (2 + \rho)x - \rho$, and $\nabla^2 P_2(x, \rho) = 2 + \rho$. In this case, as $\nabla^2 P_2(x, \rho) > 0$ for all x , $P_2(x, \rho)$ is convex, with unique minimizer

$$x(\rho) = \frac{\rho}{2 + \rho} = \frac{1}{2/\rho + 1}.$$

This formula gives minimizers for $\rho = 1, 10, 100$ and 1000 as $x(1) = \frac{1}{3}$, $x(10) = \frac{5}{6}$, $x(100) = \frac{100}{102} \approx 0.9804$, and $x(1000) = \frac{1000}{1002} \approx 0.9980$. As $\rho \rightarrow \infty$, we have $x(\rho) \rightarrow 1$, as required. In Figure 4.4 the graphs of $P_2(x, \rho)$ for $\rho = 1, 10$ and 100 are superimposed on the graph of the objective function x^2 .

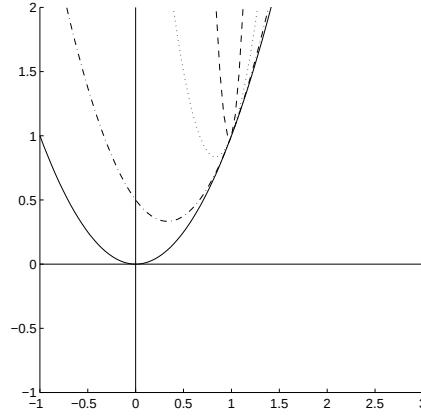


Figure 4.4: This figure depicts the graphs of various penalty functions for the problem $\min x^2$ subject to $x-1=0$. Graphs of the quadratic penalty function $P_2(x, \rho) = x^2 + \frac{1}{2}\rho(x-1)^2$ are superimposed on the graph of the objective $f(x) = x^2$. The dash-dot line is $P_2(x, 1)$, the dotted line is $P_2(x, 10)$, and the dashed line is $P_2(x, 100)$. The solid line is the graph of $x^2 = P_2(x, 0)$.

Next we state the conditions under which we can expect local minimizers of the penalty function to be close to local solutions of NEP. These conditions require that a particular subset of local constrained minimizers be “isolated” within the full set of local constrained minimizers.

Definition 4.6.1. (*Isolated subset.*) Let X and X^* be sets in $\mathbb{R}^n$ such that $X^* \subseteq X$. The set X^* is called an isolated subset of X if there exists a closed set S such that $X^* \subset \text{int}(S)$ and $S \cap X = X^*$.

Broadly speaking, X^* is “separated” by S from any other points of X . The definition is satisfied if $X^* = X$, or if X^* is an isolated point in X .

Theorem 4.6.1. (*Convergence of the quadratic penalty method.*) Let $\{\rho_k\}$ be a strictly increasing unbounded positive sequence. Assume that there exists a nonempty, isolated compact set X^* of local solutions of NEP with the same optimal function value. Then there exists a compact set S such that $X^* \subset S$, and for sufficiently large ρ_k , there exist unconstrained minimizers of $P_2(x, \rho_k)$ in the interior of S . Further, every limit point of any subsequence of the minimizing points is in X^* . ■

In effect, this result ensures that applying the quadratic penalty-function transformation creates a set of *local minimizers* of the penalty function. For ρ sufficiently large, and within a bounded region including x^* , the sequence of local minimizers of $P_2(x, \rho)$ will converge to x^* . It may nonetheless happen that the unconstrained algorithm used to

minimize $P_2(x, \rho)$ will fail to converge to the desired local minimizer. This limitation is illustrated in the following example:

$$\underset{x \in \mathbb{R}^1}{\text{minimize}} \quad x^3 \quad \text{subject to} \quad x + 1 = 0,$$

whose solution is trivially $x^* = -1$. The associated penalty function is

$$P_2(x, \rho) = x^3 + \frac{\rho}{2}(x+1)^2, \quad (4.6.2)$$

which has derivatives $\nabla P_2(x, \rho) = 3x^2 + \rho(x+1)$, $\nabla^2 P_2(x, \rho) = 6x + \rho$. For a given value of ρ , stationary points of $P_2(x, \rho)$ lie at $x(\rho) = -\frac{1}{6}\rho \pm \frac{1}{6}\sqrt{\rho^2 - 12\rho}$. The second derivative of P_2 is $6x + \rho$, which implies that for all $\rho > 12$, the stationary point corresponding to the negative sign is a local maximizer, and the stationary point corresponding to the positive sign is the local minimizer

$$x(\rho) = -\frac{\rho}{6} + \frac{\rho}{6}\sqrt{1 - 12/\rho} = -1 - \frac{3}{\rho} + O\left(\frac{1}{\rho^2}\right).$$

For this local minimizer, $\lim_{\rho \rightarrow \infty} x(\rho) = x^* = -1$. However, the penalty function (4.6.2) is unbounded below for any finite ρ , and hence there is no guarantee that an unconstrained minimization will converge to the minimizer from an arbitrary starting point. Figure 4.5 gives the penalty function $P_2(x, \rho)$ for $\rho = 18$.

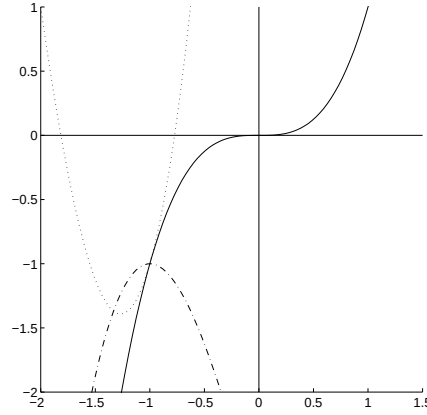


Figure 4.5: Graph of the quadratic penalty function $P_2(x, \rho) = x^3 + \frac{1}{2}\rho(x+1)^2$ for $\rho = 18$. The solid line is the graph of x^3 . The dash-dot line is the graph of the Lagrangian $L(x, \lambda^*)$ with optimal Lagrange multiplier $\lambda^* = 3$.

The solutions of intermediate unconstrained subproblems have several interesting features. Given a strictly increasing positive unbounded sequence $\{\rho_k\}$, let $\{x(\rho_k)\}$ denote the local unconstrained minimizer of $P_2(x, \rho_k)$ nearest to x^* . Then it can be shown that

(i) the sequence $\{P_2(x(\rho_k), \rho_k)\}$ is nondecreasing; (ii) the sequence $\{f_k\}$ is nondecreasing; and (iii) the sequence $\{\|c(x(\rho_k))\|_2\}$ is nonincreasing. Thus, in general, each successive $x(\rho_k)$ displays a decreasing measure of infeasibility and an increasing value of the objective function. Further, each $x(\rho_k)$ is a point at which no direction exists that is simultaneously a descent direction for both f and $\|c\|_2$.

4.6.2. Derivatives of the penalty function

If an unconstrained minimization method is to be applied to minimize the penalty function, it is necessary to know the first and second derivatives of P_2 at every point x . Differentiation of $P_2(x, \rho)$ gives

$$\nabla P_2(x, \rho) = g(x) + \sum_{i=1}^m \rho \nabla c_i(x) c_i(x) = g(x) + \rho J(x)^T c(x) \quad (4.6.3)$$

$$\nabla^2 P_2(x, \rho) = H(x) + \sum_{i=1}^m \rho c_i(x) H_i(x) + \rho J(x)^T J(x). \quad (4.6.4)$$

As P_2 is a smooth function, its gradient must vanish at the unconstrained minimizer $x(\rho)$. It follows from (4.6.3) that

$$g(x(\rho)) = -\rho J(x(\rho))^T c(x(\rho)), \quad (4.6.5)$$

which states that the gradient of the objective is a linear combination of the constraint gradients at a minimizer $x(\rho)$. If the constraint qualification holds, this condition has the same form as the first-order necessary condition for optimality

$$g(x^*) = J(x^*)^T \lambda^* \quad (4.6.6)$$

(see Theorem 4.2.3). A comparison of these two optimality conditions indicates that the quantity $\lambda_i(\rho) \triangleq -\rho c_i(x(\rho))$, $i = 1, \dots, m$, may be viewed as an estimate of the i th Lagrange multiplier at x^* . If $J(x^*)$ has full rank and the sufficient conditions of Theorem 4.2.5 hold at x^* , the convergence of $x(\rho)$ to x^* and the uniqueness of the Lagrange multipliers imply that

$$\lim_{\rho \rightarrow \infty} \lambda_i(\rho) = \lambda_i^*. \quad (4.6.7)$$

At a minimizer of the penalty function, the Hessian $\nabla^2 P_2(x, \rho)$ must be positive semidefinite. Using the identities $\lambda_i(\rho) = -\rho c_i(x(\rho))$ in (4.6.4) gives

$$\begin{aligned} \nabla^2 P_2(x(\rho), \rho) &= H(x(\rho)) - \sum_{i=1}^m \lambda_i(\rho) H_i(x(\rho)) + \rho J(x(\rho))^T J(x(\rho)) \\ &= H(x(\rho), \lambda(\rho)) + \rho J(x(\rho))^T J(x(\rho)), \end{aligned} \quad (4.6.8)$$

where $H(x, \lambda)$ denotes the Hessian of the Lagrangian with respect to x . For large values of ρ it will be the case that $H(x(\rho), \lambda(\rho)) \approx H(x^*, \lambda^*)$. It follows that the term $\rho J(x(\rho))^T J(x(\rho))$ has the effect of inducing a local solution of the penalty function by adding positive curvature to the (usually indefinite) Lagrangian Hessian.

4.6.3. The penalty trajectory

Under suitable assumptions, the set of unconstrained minimizers of the penalty function can be regarded as a function of an independent variable, tracing out a smooth *trajectory* of points converging to x^* .

Theorem 4.6.2. (*Smooth trajectory of the quadratic penalty function.*) Assume that $J(x^*)$ has full rank, and that the sufficient conditions of Theorem 4.2.5 hold at x^* . Then, for ρ sufficiently large, there exist continuously differentiable functions $x(\rho)$ and $\lambda(\rho)$, such that

$$\lim_{\rho \rightarrow \infty} x(\rho) = x^* \quad \text{and} \quad \lim_{\rho \rightarrow \infty} \lambda(\rho) = \lambda^*.$$

Further, $x(\rho)$ is an unconstrained local minimizer of $P_2(x, \rho)$ for any finite ρ .

Proof. It follows from the first-order optimality conditions for NEP, that the equations

$$\begin{aligned} g(x) - J(x)^T \lambda &= 0 \\ c(x) + \mu \lambda &= 0 \end{aligned} \tag{4.6.9}$$

are satisfied at $(x, \lambda) = (x^*, \lambda^*)$ for $\mu = 0$. The Jacobian of this system is

$$J_\mu(x, \lambda) = \begin{pmatrix} H(x, \lambda) & -J(x)^T \\ J(x) & \mu I \end{pmatrix},$$

where $H(x, \lambda)$ denotes the Hessian of the Lagrangian at (x, λ) . Under the given assumptions, $J_0(x^*, \lambda^*)$ is nonsingular, and the implicit function theorem states that in a neighborhood of $\mu = 0$ there exists a unique differentiable vector function $(x(\mu), \lambda(\mu))$ satisfying (4.6.9).

Thus, for ρ sufficiently large, we may define $\mu = 1/\rho$ and assert that there exists a point $(x(\rho), \lambda(\rho))$ that satisfies the first-order optimality conditions

$$\begin{aligned} g(x) - J(x)^T \lambda &= 0, \\ \rho c(x) + \lambda &= 0. \end{aligned}$$

The final step of the proof is to show that the stationary point $x(\rho)$ is also a *minimizer* of $P_2(x, \rho)$, i.e., it must be shown that $H + \rho J^T J$ is positive definite for sufficiently large ρ , where H and J denote the Lagrangian Hessian and Jacobian evaluated at $(x(\rho), \lambda(\rho))$ (see expression (4.6.8) for the Hessian of P_2 at a stationary point of P_2). Since $J(x^*)$ has rank m , it follows that J will also have rank m for ρ sufficiently large, and there must exist an $n \times n$ matrix Q (depending on ρ) such that

$$JQ = \begin{pmatrix} 0 & I \end{pmatrix} \quad \text{with } Q \text{ nonsingular.}$$

Let Q be partitioned so that $Q = \begin{pmatrix} Z & Y \end{pmatrix}$, where Z is $n \times (n - m)$ and Y is $n \times m$. Then $JZ = 0$ and $JY = I$, which implies that the columns of Z form a basis for the null space of J , and Y is a left-inverse for J .

Pre- and post-multiplying $H + \rho J^T J$ by Q^T and Q gives

$$Q^T(H + \rho J^T J)Q = Q^T H Q + \rho Q^T J^T J Q = \begin{pmatrix} Z^T H Z & Z^T H Y \\ Y^T H Z & Y^T H Y + \rho I \end{pmatrix}.$$

Note that this matrix is of the form

$$\begin{pmatrix} H_{11} & H_{21}^T \\ H_{21} & H_{22} + \rho I \end{pmatrix},$$

where $H_{11} = Z^T H Z$, $H_{21} = Y^T H Z$ and $H_{22} = Y^T H Y$. Since x^* satisfies the second-order sufficient conditions for a local constrained minimizer, H_{11} is positive definite and we can write the block 2×2 matrix as the product

$$\begin{pmatrix} I & 0 \\ H_{21}H_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} H_{11} & 0 \\ 0 & H_{22} - H_{21}H_{11}^{-1}H_{21}^T + \rho I \end{pmatrix} \begin{pmatrix} I & H_{11}^{-1}H_{21}^T \\ 0 & I \end{pmatrix}.$$

Sylvester's Law of Inertia then implies that the inertia of $H + \rho J^T J$ is the same as the inertia of $\text{diag}(H_{11}, H_{22} - H_{21}H_{11}^{-1}H_{21}^T + \rho I)$. Clearly, this matrix is positive definite for all $\rho > \bar{\rho}$, where $\bar{\rho} = \max\{-\lambda_{\min}, 0\}$, with $\lambda_{\min}$ the smallest eigenvalue of $H_{22} - H_{21}H_{11}^{-1}H_{21}^T$. $\blacksquare$

The trajectory of minimizers of the quadratic penalty function has several interesting properties. In order to discuss conditions at the limit point, we introduce the variable $\mu = 1/\rho$, use the notation $x(\mu) \triangleq x(\rho)$, and take limits as μ approaches zero. Since Theorem 4.6.2 implies the existence of a smooth function $x(\mu)$, we expand about $\mu = 0$:

$$x(\mu) = x^* + \mu y + O(\mu^2),$$

where y is the tangent to the trajectory at $\mu = 0$, i.e.,

$$y \triangleq x'(\mu) = \lim_{\mu \rightarrow 0} \frac{1}{\mu} (x(\mu) - x^*) = \left. \frac{dx(\mu)}{d\mu} \right|_{\mu=0}.$$

Because $x(\mu)$ is an unconstrained minimizer of the quadratic penalty function, for $\mu > 0$ we have the identity

$$\mu \nabla P_2(x(\mu), \mu) \equiv \mu g(\mu) + J(\mu)^T c(\mu) = 0, \quad (4.6.10)$$

where the notation “ (μ) ” denotes evaluation at $x(\mu)$. Differentiating (4.6.10) with respect to μ at $x(\mu)$ and using the chain rule, we obtain:

$$\frac{d}{d\mu}(\mu g + J^T c) = g + \mu H x'(\mu) + J^T J x'(\mu) + \sum_{i=1}^m c_i H_i x'(\mu) = 0, \quad (4.6.11)$$

where H denotes $\nabla^2 f$, H_i denotes $\nabla^2 c_i$, and all functions are evaluated at $x(\mu)$. As $\mu \rightarrow 0$, we know that $c_i \rightarrow 0$, and hence, using (4.6.6), (4.6.11) becomes in the limit:

$$g(x^*) + J(x^*)^T J(x^*) y = J(x^*)^T \lambda^* + J(x^*)^T J(x^*) y = 0. \quad (4.6.12)$$

Equation (4.6.12) implies that $J(x^*)^T (\lambda^* + J(x^*) y) = 0$ and since $J(x^*)$ has full row rank, we may assert that

$$J(x^*) y = -\lambda^*. \quad (4.6.13)$$

If $\lambda_i^* \neq 0$, (4.6.13) implies that the trajectory of minimizers of the quadratic penalty function generates a *nontangential* approach to x^* . This property of the trajectory is a crucial factor in the construction of efficient algorithms based on minimizing P_2 .

Next we show that at any point $(x(\mu), \lambda(\mu))$ on the trajectory, the tangent vector $x'(\mu)$ can be found by solving a linear system involving the first and second derivatives of f and c . Points on the trajectory satisfy the identities

$$\begin{aligned} g - J^T \lambda &= 0, \\ c + \mu \lambda &= 0, \end{aligned}$$

where $x = x(\mu)$, $\lambda = \lambda(\mu)$, $g = g(x(\mu))$ and $J = J(x(\mu))$. Differentiating these identities with respect to μ gives

$$\begin{aligned} H(x, \lambda)x'(\mu) - J^T \lambda'(\mu) &= 0, \\ Jx'(\mu) + \lambda'(\mu) &= -\lambda(\mu), \end{aligned}$$

where $\lambda'(\mu) = d\lambda(\mu)/d\mu$ and $x'(\mu) = dx(\mu)/d\mu$. These equations imply that $(x'(\mu), \lambda'(\mu))$ satisfies the system

$$\begin{pmatrix} H(x, \lambda) & -J^T \\ J & \mu I \end{pmatrix} \begin{pmatrix} x'(\mu) \\ \lambda'(\mu) \end{pmatrix} = \begin{pmatrix} 0 \\ -\lambda(\mu) \end{pmatrix}. \quad (4.6.14)$$

or, equivalently,

$$(H + \frac{1}{\mu} J^T J)x'(\mu) = -\frac{1}{\mu} J^T \lambda, \quad \text{and} \quad \lambda'(\mu) = -\frac{1}{\mu}(\lambda + Jx'(\mu)).$$

This form of the tangent to the trajectory will be considered further in the discussion of the classical penalty method.

4.6.4. The “classical” penalty method

Based on the theory developed thus far, it might appear that the solution to NEP could be found simply by setting ρ to a very large value and applying a standard unconstrained minimization method to $P_2(x, \rho)$. Unfortunately, the Hessians $\nabla^2 P(x(\rho), \rho)$ become increasingly ill-conditioned matrices as ρ increases. To see why, observe from (4.6.8) that the Hessian of P_2 can be written in the form $\nabla^2 P_2 = H + \rho J^T J$. As $\rho \rightarrow \infty$, $\lambda \rightarrow \lambda^*$ and the first term in this expression approaches the bounded matrix $H(x^*, \lambda^*)$. Thus, if $m < n$, the matrix $\nabla^2 P_2(x, \rho)$ evaluated at $x(\rho)$ is dominated by the unbounded rank-deficient matrix $\rho J^T J$.

Analysis of the eigenvalues and eigenvectors of $\nabla^2 P_2(x(\rho), \rho)$ as $\rho \rightarrow \infty$ reveals that $(n - m)$ eigenvalues are bounded, with associated eigenvectors that in the limit lie in the null space of $J(x^*)$. However, the remaining m eigenvalues are of order ρ , i.e., unbounded in the limit, and the corresponding eigenvectors lie in the range of $J(x^*)^T$.

The effect of this ill-conditioning on the unconstrained minimization problem is illustrated in Figure 4.6. This figure shows the level curves of the quadratic penalty function $P_2(x, \rho)$ for the two-dimensional example (4.1.1). The two figures correspond to ρ -values of 10 and 100. The larger the value of ρ , the more the level curves follow the curvature

of the nonlinear constraint. Moreover, the sides of the curving valley become steeper as ρ increases. This implies that if an unconstrained minimization method is started at an arbitrary point, the iterates quickly move to the base of the valley close to the constraint boundary. Subsequent iterates must follow the (possibly highly nonlinear) constraint surface by negotiating a path along the base of the curved valley.

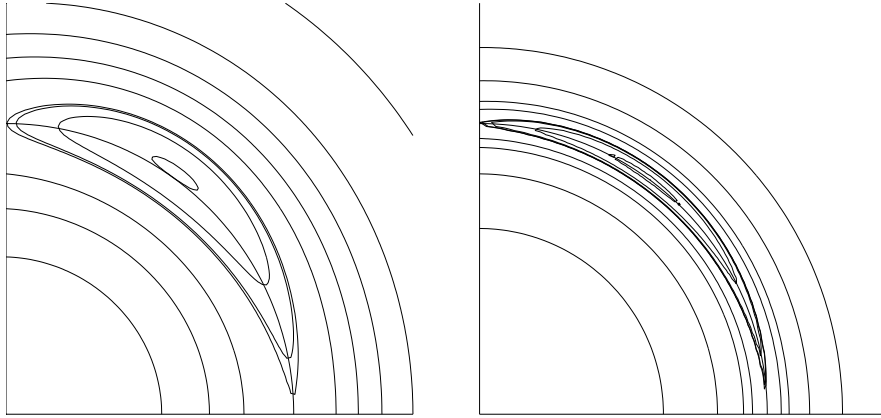


Figure 4.6: These graphs depict the level curves of the quadratic penalty function $P_2(x, \rho)$ for the two-dimensional example (4.1.1). The left and right figures correspond to the ρ -values of 10 and 100 respectively.

This difficulty can be alleviated by exploiting the properties of the penalty trajectory. If the Lagrange multipliers are nonzero, the trajectory makes a non-tangential approach to the constraint surface. If the iterates can be made to approach the solution along the path of the trajectory, then the difficulties associated with approaching the solution along the constraint surface are avoided. The solution can be approached along the trajectory by approximately minimizing $P_2(x, \rho)$ for each of a sequence of increasing values of ρ —say, 10, 10^2 , 10^3 , etc. This scheme for increasing ρ is the basis for the so-called “classical” penalty method. The first minimization, which has the effect of moving the iterates to a neighborhood of the trajectory, is likely to be the most difficult subproblem (given an arbitrary starting point). However, the problem will not be ill-conditioned if the initial value of ρ is not too large. Inevitably, the problems will become ill-conditioned as ρ is increased, but the hope is that the change in variables will lie in the “well-defined” direction defined by the range-space of the constraint gradients.

Algorithm 4.6.1 is a typical first-order method based on the quadratic penalty function. It uses a modified Newton method in conjunction with a backtracking line search.

Unfortunately, the classical penalty method can still require a significant number of iterations when solving the later subproblems. Until quite recently, this inefficiency was thought to be caused by rounding errors associated with the calculation of the search direction from the ill-conditioned Newton equations. However, the classical penalty method can behave poorly even if the search direction is computed with no rounding errors!

Algorithm 4.6.1. Classical Newton Penalty Method for NEP.
 Choose x_0 , ρ ($\rho > 0$), τ ($0 < \tau < \frac{1}{2}$), γ ($\gamma > 1$), and ϵ ($0 < \epsilon \ll 1$);
 $x \leftarrow x_0$;
 $k \leftarrow 0$;
while not converged do
 while $\|\nabla P_2(x, \rho)\| > \epsilon$ **do**
 Compute $P(x, \rho)$, $\nabla P_2(x, \rho)$;
 Define E such that $\nabla^2 P_2(x, \rho) + E$ is positive definite;
 Solve $(\nabla^2 P_2(x, \rho) + E)\Delta x = -\nabla P_2(x, \rho)$;
 $\alpha \leftarrow 1$;
 while $P_2(x + \alpha\Delta x, \rho) > P_2(x, \rho) + \tau\alpha\nabla P_2(x, \rho)^T \Delta x$ **do**
 $\alpha \leftarrow \alpha/2$;
 end do
 $x \leftarrow x + \alpha\Delta x$;
 $k \leftarrow k + 1$;
 end do
 $\rho \leftarrow \gamma * \rho$;
end do

To understand the behavior of the classical penalty method, it is helpful to write the Newton equations in terms of the m -vector $\pi(x)$ with components

$$\pi_i(x) \triangleq -\rho c_i(x). \quad (4.6.15)$$

This vector has some important properties and is known as the vector of *primal multipliers*. From (4.6.6) we observe that at any point $x(\rho)$ on the trajectory, we can define $\lambda_i(\rho) = -\rho c_i(x(\rho))$, which approximates the Lagrange multiplier λ_i^* . This definition implies that for a given value of ρ , and for any sequence $\{x_k\}$ converging to $x(\rho)$,

$$\lim_{k \rightarrow \infty} \pi(x_k) = \lambda(\rho).$$

Written in terms of x_k and $\pi(x_k)$, the Newton equations are

$$(H(x_k, \pi_k) + \rho J(x_k)^T J(x_k))\Delta x_k = -(g(x_k) - J(x_k)^T \pi_k), \quad (4.6.16)$$

where π_k denotes the vector $\pi(x)$ evaluated at x_k .

It may be verified by direct multiplication that if $E = 0$, the classical penalty method search direction satisfies the system

$$\begin{pmatrix} H(x_k, \pi_k) & -J(x_k)^T \\ J(x_k) & \frac{1}{\rho}I \end{pmatrix} \begin{pmatrix} \Delta x_k \\ -\eta_k \end{pmatrix} = - \begin{pmatrix} g(x_k) - J(x_k)^T \pi_k \\ c(x_k) + \frac{1}{\rho} \pi_k \end{pmatrix}.$$

As $(x_k, \pi_k) \rightarrow (x^*, \lambda^*)$, the matrix of this system approaches the matrix

$$\begin{pmatrix} H(x^*, \lambda^*) & -J(x^*)^T \\ J(x^*) & 0 \end{pmatrix},$$

which is well conditioned as long as problem NEP is well-conditioned. It follows that large values of ρ do not introduce unnecessary numerical error in the calculation of Δx_k using this system. It can be shown that the absolute error in Δx_k is reasonable even when the Newton equations $\nabla^2 P_2(x_k, \rho) \Delta x_k = -\nabla P_2(x_k, \rho)$ are solved directly.

If numerical error is not the cause of the inefficiency, what is? The difficulty occurs when the value of ρ is increased at the start of the new problem. In practice, each unconstrained minimization is done only approximately, with a termination condition that determines the accuracy of the unconstrained minimizers. An important feature of any penalty method is that, in the limit, this termination condition should allow only *one* inner iteration for each value of ρ . If this is to be the case, the first direction associated with $\bar{\rho}$ must provide an increasingly good estimate of $(x(\bar{\rho}), \lambda(\bar{\rho}))$.

Suppose that we have just obtained the point $(x(\mu), \lambda(\mu))$ on the trajectory by solving a penalty subproblem with penalty parameter $\rho = 1/\mu$. The next step is to define $\bar{\mu}$ ($\bar{\mu} = 1/\bar{\rho}$) and solve the Newton equations

$$\begin{pmatrix} H(x, \bar{\pi}) & -J(x)^T \\ J & \bar{\mu}I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \lambda \end{pmatrix} = - \begin{pmatrix} g - J^T \bar{\pi} \\ c + \bar{\mu} \bar{\pi} \end{pmatrix}, \quad (4.6.17)$$

where $\bar{\pi} = -c(x(\bar{\mu}))/\bar{\mu}$. If the direction Δx is to be satisfactory, the new iterate $x + \Delta x$ must be close to $x(\bar{\mu})$, the point on the trajectory corresponding to the new penalty parameter. This implies that Δx must be approximately parallel to the tangent to the trajectory at $x(\mu)$. If this is not the case, Δx will have a relatively significant component in the null space of $J(x(\mu))$ and the unit step is unlikely to be accepted by the line search. Unfortunately, the “classical” Newton direction Δx_k is not usually parallel to the tangent $x'(\mu)$ to the trajectory at $x(\mu)$. To show this, it is only necessary to compare the solutions of the two systems (4.6.17) and (4.6.14). Since $\bar{\pi}$ is substantially different from $\pi(\mu)$, the right-hand-side vector $g - J^T \bar{\pi}$ is not close to zero. Hence $x + \Delta x$ is generally a poor estimate of $x(\bar{\mu})$ and the potential quadratic convergence rate is inhibited by the line search.

To illustrate this phenomenon, consider the application of the classical penalty method to the problem

$$\begin{array}{ll} \underset{x_1, x_2}{\text{minimize}} & x_1 x_2 - \frac{1}{2} x_1^2 - x_2 \\ \text{subject to} & 2 - x_1^2 - x_2^2 = 0. \end{array} \quad (4.6.18)$$

Assume that the initial point x_0 is a minimizer of $P_2(x, \rho)$ with $\rho = 10$. Since x_0 lies on the trajectory, $\|\nabla P_2(x, \rho)\|_2 = 0$. If the new penalty value is $\bar{\rho} = 100\rho = 1000$, the initial value of penalty gradient is $\|\nabla P_2(x_0, 1000)\| \approx 1.75 \times 10^2$, and the penalty search direction and tangent to the trajectory are given by

$$\Delta x_0 \approx \begin{pmatrix} -2.03544 \\ 0.48303 \end{pmatrix}, \quad \text{and} \quad x'(\mu) \approx \begin{pmatrix} 0.18966 \\ -0.13658 \end{pmatrix}.$$

These vectors are not even close to being parallel! (The sign is irrelevant since $x'(\mu)$ always points away from the solution.) The classical penalty algorithm requires a further six iterations to reduce $\|\nabla P_2(x, 1000)\|_2$ to $O(10^{-14})$.

4.6.5. The primal-dual penalty method

For any value of ρ , the point $(x(\rho), \lambda(\rho))$ on the penalty trajectory satisfies the $n + m$ equations

$$g(x) - J(x)^T \lambda = 0, \quad (4.6.19a)$$

$$c_i(x) + \frac{1}{\rho} \lambda_i = 0. \quad (4.6.19b)$$

These relations imply that $(x(\rho), \lambda(\rho))$ can be determined by solving $n + m$ nonlinear equations in the $n + m$ unknowns (x, λ) using an iterative method, which is usually some variant of Newton's method.

Conditions (4.6.19) imply that $(x(\rho), \lambda(\rho))$ is a zero of the nonlinear function $F(x, \lambda)$, where $F(x, \lambda)$ and its Jacobian $F'(x, \lambda)$ are given by

$$F(x, \lambda) = \begin{pmatrix} g(x) - J(x)^T \lambda \\ c(x) + \frac{1}{\rho} \lambda \end{pmatrix}, \quad F'(x, \lambda) = \begin{pmatrix} H(x, \lambda) & -J(x)^T \\ J(x) & \frac{1}{\rho} I \end{pmatrix}.$$

Suppose that (x, λ) is an estimate of $(x(\rho), \lambda(\rho))$. Let π, g, c, J and H denote the quantities $\pi(x), g(x), c(x), J(x)$ and $H(x, \lambda)$. Given (x, λ) , the next iterate of Newton's method for a solution of $F = 0$ is $(x + \Delta x, \lambda + \Delta \lambda)$, where $(\Delta x, \Delta \lambda)$ satisfies the Newton equations

$$\begin{pmatrix} H & -J^T \\ J & \frac{1}{\rho} I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \lambda \end{pmatrix} = - \begin{pmatrix} g - J^T \lambda \\ c + \frac{1}{\rho} \lambda \end{pmatrix}. \quad (4.6.20)$$

These equations are known as the *primal-dual Newton equations*,

For computational purposes, it is best to reformulate these equations as a symmetric system so that the symmetry can be exploited by the linear system solver. Our aim is to produce symmetric equations analogous to those associated with the method of multipliers (see (4.4.2)). This may be achieved by multiplying both sides of (4.6.20) by -1 and changing the sign of the λ variables. This gives the symmetric primal-dual Newton equations

$$K \begin{pmatrix} \Delta x \\ -\Delta \lambda \end{pmatrix} = - \begin{pmatrix} g - J^T \lambda \\ c + \frac{1}{\rho} \lambda \end{pmatrix}, \quad \text{where} \quad K = \begin{pmatrix} H & J^T \\ J & -\frac{1}{\rho} I \end{pmatrix}. \quad (4.6.21)$$

The symmetric matrix K is known as the *primal-dual KKT matrix*. This symmetric system can be solved in many ways. For example, block elimination can be used to write the $\Delta \lambda$ variables in terms of the Δx variables, giving the two smaller systems

$$\begin{aligned} (H + \rho J^T J) \Delta x &= -(g - J^T \pi) \\ \Delta \lambda &= \pi - \lambda - \rho J \Delta x. \end{aligned}$$

If the second-order sufficient conditions for optimality hold, the Jacobian $F'(x, \lambda)$ is nonsingular in a neighborhood of (x^*, λ^*) for ρ sufficiently large (see §4.6.3). This result allows us to apply the standard convergence theory for Newton's method, which states that the Newton iterates will converge quadratically in a sufficiently small neighborhood of $(x(\rho), \lambda(\rho))$.

4.6.6. The ℓ_1 penalty function

With the quadratic penalty function, the penalty parameter ρ must become infinite in order to achieve convergence to x^* . In contrast, we can devise a *non-differentiable* penalty function of which x^* is the unconstrained minimizer. (Such a function is called an *exact* penalty function.)

The most widely used exact penalty function is the ℓ_1 *penalty function*, or *absolute-value penalty function*:

$$P_1(x, \rho) \equiv f(x) + \rho \sum_{i=1}^m |c_i(x)| = f(x) + \rho \|c(x)\|_1, \quad (4.6.22)$$

where $\rho \geq 0$. The ℓ_1 penalty function has been used for many years (under different names) in structural and mechanical design problems.

For example, consider the problem

$$\underset{x \in \mathbb{R}^1}{\text{minimize}} \quad x^2 \quad \text{subject to} \quad x - 1 = 0,$$

with solution $x^* = 1$ and multiplier $\lambda^* = 2$. The associated ℓ_1 penalty function is

$$P_1(x, \rho) = x^2 + \rho|x - 1| = \begin{cases} x^2 + \rho(x - 1) & \text{if } x \geq 1; \\ x^2 - \rho(x - 1) & \text{if } x < 1, \end{cases}$$

of which x^* is an unconstrained minimizer if $\rho > 2$. The derivative of $P_1(x, \rho)$ is the piecewise continuous function,

$$P'_1(x, \rho) = \begin{cases} 2x + \rho & \text{if } x \geq 1; \\ 2x - \rho & \text{if } x < 1, \end{cases}$$

which is undefined at the solution $x^* = 1$.

The function $P_1(x, \rho)$ has discontinuous derivatives at any point where a constraint function vanishes, and hence x^* will be a point of discontinuity in the gradient of P_1 . An important difference between P_1 and P_2 (4.6.1) is that, under mild conditions, ρ in (4.6.22) need not become arbitrarily large in order for x^* to be an unconstrained minimizer of P_1 . Rather, there is a threshold value ρ^* such that x^* is an unconstrained minimizer of P_1 for *all* $\rho > \rho^*$. Thus, whenever ρ in (4.6.22) is “sufficiently large”, an unconstrained minimizer of P_1 will be a solution of the original problem NEP.

Theorem 4.6.3. *If x^* is a solution of problem NEP, then there exists a parameter ρ^* ($\rho^* \geq 0$) such that x^* is an unconstrained minimizer of $P_1(x, \rho)$ for all $\rho \geq \rho^*$. Moreover, if NEP satisfies the constraint qualification at x^* , then $\rho^* = \|\lambda^*\|_\infty$, where λ^* is a Lagrange multiplier vector at x^* . ■*

Methods based on non-differentiable penalty functions avoid the large values of ρ associated with P_2 , since the penalty parameter can remain finite. However, the crucial value ρ^* depends on quantities evaluated at x^* (which is, of course, unknown), and therefore the value of ρ may need to be adjusted as the algorithm proceeds. Difficulties can arise

if an unsuitable value of the penalty parameter is chosen. If ρ is too small, the penalty function may be unbounded below. On the other hand, the unconstrained subproblem will be ill-conditioned if ρ is too large, and special techniques are then required to obtain an accurate solution. Figure 4.7 shows the level curves of the ℓ_1 penalty function $P_1(x, \rho)$ for the two-dimensional example (4.1.1). The two figures correspond to the ρ -values 10 and 100.

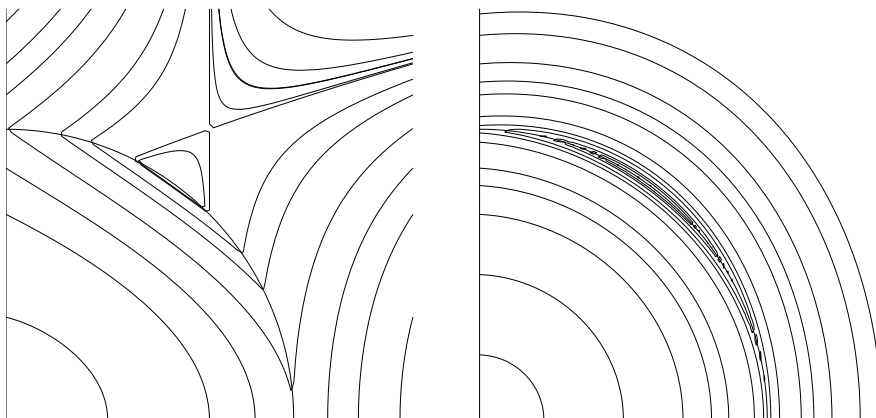


Figure 4.7: These graphs depict the level curves of the ℓ_1 penalty function $P_1(x, \rho)$ for the two-dimensional example (4.1.1). The values of ρ are $\rho = 1$ and $\rho = 10$. The feasible region consists of points lying on the circle $x_1^2 + x_2^2 = 2$.

Since P_1 is non-differentiable even at the solution, standard unconstrained methods designed for smooth problems cannot be applied directly. However, special algorithms have been designed for minimizing (4.6.22) that utilize information about the original nonlinearly constrained problem. Such methods are not discussed here.

Chapter 5

Inequality Constraints

5.1. Introduction

The problems considered in this chapter have variables that are restricted by *inequality* constraints. The general statement of the problem is:

$$\begin{array}{ll} \text{NIP} & \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) \\ & \text{subject to} \quad c(x) \geq 0, \end{array}$$

where $c(x)$ is a vector-valued function with m components $c_i(x)$, and $f(x)$ and $\{c_i(x)\}$ are smooth functions. For the inequality constraint case, the feasible region is given by

$$\mathbb{F} = \{x \mid c_i(x) \geq 0, i = 1, \dots, m\}.$$

The definition of NIP makes no assumptions about the properties of f and c other than that they are “smooth” (which generally means that f and c are twice-continuously differentiable). However, in some cases it is possible to derive methods that exploit special properties of f and c . A *linear programming problem* is an optimization problem in which the objective function and constraint functions are linear. An optimization problem is called a *nonlinear program* if the objective or any constraint function is nonlinear. A *quadratic program* is a nonlinear program with quadratic objective function and linear constraints.

For illustrative purposes, we will adapt the equality-constraint problem (4.1.1) of Chapter 4. Fig. 5.1 depicts the level curves of the two-variable problem

$$\begin{array}{ll} \underset{x_1, x_2}{\text{minimize}} & -x_1 x_2^2 \\ \text{subject to} & 2 - x_1^2 - x_2^2 \geq 0, \\ & x_1, x_2 \geq 0. \end{array} \tag{5.1.1}$$

In this case, there are three inequality constraints, one nonlinear: $c_1(x) = 2 - x_1^2 - x_2^2$, and two linear: $c_2(x) = x_1$ and $c_3(x) = x_2$. The feasible region $\mathbb{F}$ consists of all points in the positive quadrant lying inside the circle $x_1^2 + x_2^2 = 2$. If this problem is compared to the example (4.1.1) of NEP, it can be seen that the (unique) solution is $x^* = (0.8165, 1.1547)$. At this point, the function $c_1(x)$ associated with the first constraint is exactly

zero, while the second and third constraints have positive values of $c_2(x)$ and $c_3(x)$. The constraint $c_1(x) \geq 0$ is said to be *active* or *satisfied exactly* at $x = x^*$, and the other constraints are said to be *inactive* or *satisfied*. If it could be determined in advance that $c_1(x) \geq 0$ is the only constraint exactly satisfied at the solution, then the constraints $c_2(x) \geq 0$ and $c_3(x) \geq 0$ could be ignored, and x^* could be found by minimizing $f(x)$ subject to the single constraint $c_1(x) \geq 0$ *maintained as an equality*. This is more or less how many successful algorithms for solving NIP proceed—the objective is minimized with a subset of the inequality constraints held at equality. In practice, of course, the set of constraints active at a solution is not known in advance. The additional problem of identifying the subset of active constraints makes minimization subject to inequality constraints considerably more difficult than minimization subject to equality constraints.

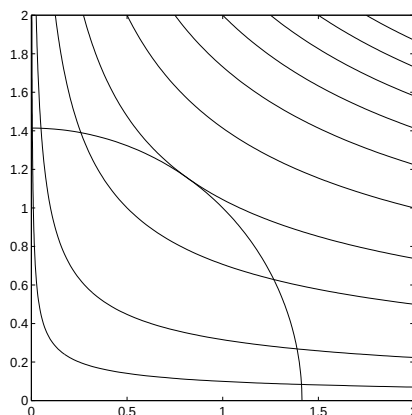


Figure 5.1: This graph depicts the level curves and constraint functions for the two-dimensional example (5.1.1). The feasible region consists of points that lie inside the circle $x_1^2 + x_2^2 = 2$ between the vertical and horizontal axes.

5.2. Optimality Conditions

We now derive optimality conditions for problem NIP. As before, the matrix $J(x)$ will denote the Jacobian of the constraint vector $c(x)$. We start with definitions that formalize some of the observations made in the introduction.

Definition 5.2.1. *The constraint $c_i(x) \geq 0$ is said to be satisfied at $\bar{x}$ if $c_i(\bar{x}) \geq 0$, active (binding, satisfied exactly) if $c_i(\bar{x}) = 0$, inactive if $c_i(\bar{x}) > 0$, and violated if $c_i(\bar{x}) < 0$. ■*

The feasible region $\mathbb{F}$ for the constraints $c(x) \geq 0$ is said to have a *strict interior* if it contains at least one strictly feasible point $\bar{x}$ for which no constraints are active, i.e., such that $c(\bar{x}) > 0$. It is easy to see that the existence of one strictly feasible point $\bar{x}$ implies the existence of a neighborhood of *infinitely many strictly feasible points* surrounding $\bar{x}$.

Practical optimality conditions for the inequality constraint case are developed using arguments similar to those used for the equality-constraint case. The local behavior of the objective function is investigated along a feasible path that emanates from an alleged solution x . However, the derivation of optimality conditions for inequality constraints is more complicated than for equalities for two reasons. First, only a subset of the constraints are usually involved in some of the optimality conditions; and second, the set of feasible paths is much larger in the inequality case.

Given the set of inequality constraints $c(x) \geq 0$, the *active set* at a point $\bar{x}$ refers to the set of indices of the constraints active at $\bar{x}$. We shall use $\mathcal{A}(x)$ to denote the active set at x , with the argument suppressed if the meaning is clear. We use $c_a(x)$ to denote the subset of constraints that are active at x , and $J_a(x)$ the Jacobian of $c_a(x)$. The matrix $J_a(x)$ is called the matrix of *active-constraint gradients* or *active-constraint matrix* (sometimes called simply the “active set” if there is no danger of confusion with the index set $\mathcal{A}$). We emphasize that $J_a(x)$ includes only the gradients of the *active* constraints, whereas $J(x)$ is the full Jacobian of all the constraints. We shall use the notation $\nabla c_i(x)$ for the gradient of an active or inactive constraint, where the meaning should always be clear from the context.

We shall sometimes use $m_a(x)$ to denote the number of constraints active at x . The order of the indices in $\mathcal{A}(x)$ usually indicates the *order of the rows* of J_a , i.e., if

$$\mathcal{A}(x) = \{i_1, i_2, \dots, i_{m_a}\}, \quad \text{then} \quad J_a(x) = \begin{pmatrix} \nabla c_{i_1}(x)^T \\ \nabla c_{i_2}(x)^T \\ \vdots \\ \nabla c_{i_{m_a}}(x)^T \end{pmatrix}.$$

The order of the indices in $\mathcal{A}(x)$ is critical in some contexts and unimportant in others. We emphasize that both $J_a(x)$ and $\mathcal{A}(x)$ *depend on* x , although this dependence is not always indicated explicitly.

For problem NIP, the active constraints at an alleged solution have special importance because they restrict the directions of feasible paths. If x is any feasible point, a feasible path for NIP is a directed differentiable path $x(\alpha)$ starting at x such that, for some positive σ , $c(x(\alpha)) \geq 0$ for all $0 \leq \alpha \leq \sigma$. If x is *strictly feasible* with respect to a constraint, i.e., $c_i(x) > 0$ for some i , then for *any* path $x(\alpha)$, the constraint satisfies $c_i(x(\alpha)) > 0$ for $|\alpha|$ sufficiently small. Hence, if a constraint is inactive at x , then it will remain inactive for *any* path in a sufficiently small neighborhood of x . On the other hand, an active constraint restricts the direction of a feasible path in every neighborhood of a feasible point. Suppose that $c_i(x) = 0$. If $x(\alpha)$ is a feasible path such that $c_i(x(\alpha)) = 0$ for all $0 \leq \alpha \leq \sigma$, then $x(\alpha)$ is known as a *binding* feasible path at x . If p is a tangent to a binding feasible path, then

$$\nabla c_i(x)^T p = 0. \quad (5.2.1)$$

A move along a binding feasible path is said to remain “on” the constraint. In contrast, if $x(\alpha)$ is a feasible path such that $c(x(\alpha)) > 0$, for $\alpha > 0$, then $x(\alpha)$ is known as a *nonbinding* feasible path at x . If p is a tangent to a nonbinding feasible path, then

$$\nabla c_i(x)^T p > 0. \quad (5.2.2)$$

This relation implies that the constraint $c_i(x)$ becomes inactive at $x(\alpha)$ if $\alpha > 0$. A nonzero move along a non-binding feasible path is said to move “off” the constraint.

As in the equality constraint case, at any feasible point x , it is of interest to characterize the set of tangents for all feasible paths starting at x .

Definition 5.2.2. Let $T(x)$ denote the set of tangent vectors associated with every feasible path starting at a feasible point x , i.e.,

$$T(x) = \{p \mid p \text{ a nonzero tangent to a feasible path starting at } x.\}$$

Then $T^0(x) = T(x) \cup \{0\}$ is known as the tangent cone of $c(x)$ at x . ■

Consideration of the first-order necessary conditions for a minimizer of the *univariate* function $f(x(\alpha))$ at $\alpha = 0$ leads to the general first-order necessary optimality conditions for NIP.

5.2.1. First-order conditions

As in the equality constraint case, first-order conditions are derived by investigating the properties of the univariate function $f(x(\alpha))$ at $\alpha = 0$ for all feasible paths $x(\alpha)$. The first-order necessary conditions for $\alpha = 0$ to be a local minimizer of this function lead to the following theorem, stated without proof.

Theorem 5.2.1. Assume that f and c are differentiable. Then a feasible point x^* is a local solution of NIP only if $g(x^*)^T p \geq 0$ for every $p \in T^0(x)$. ■

It follows from (5.2.1) and (5.2.2) that every vector p in the tangent cone must satisfy the set of inequalities

$$J_a(x)p \geq 0. \quad (5.2.3)$$

As in the equality constraint case, we would like to derive an *algebraic* version of Theorem 5.2.1, i.e., we would like to assert that a necessary condition for a feasible point x^* to be optimal is that

$$g(x^*)^T p \geq 0 \quad \text{for all } p \text{ such that } J_a(x)p \geq 0.$$

Unfortunately, the set of vectors that satisfy (5.2.3) is not necessarily the same as the set of vectors in the tangent cone. It is not difficult to show that the set $\{p \mid J_a(x)p \geq 0\}$ is equal to the tangent cone $T_L^0(x)$ associated with the *linearized* active constraints $c_L(y) \geq 0$, where $c_L(y) = c_a(x) + J_a(x)(y - x) \geq 0$. Unfortunately, $T^0(x)$ and $T_L^0(x)$ are not necessarily identical, as the following example illustrates.

Consider the inequality constraints $c_1(x) \geq 0$ and $c_2(x) \geq 0$, where $c_1(x) = x_1^3 - x_2$ and $c_2(x) = x_2$. At the point $\bar{x} = 0$, both constraints are active, with

$$J_a(\bar{x}) = \begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix}.$$

It follows that $T_L^0(\bar{x})$ consists of all vectors of the form $p = (\gamma, 0)^T$. However, although $p = (-1, 0)^T$ is member of $T_L^0(\bar{x})$, it is not a tangent to a feasible path at $\bar{x}$, and hence $p \notin T^0(\bar{x})$. It follows that $T^0(\bar{x}) \subset T_L^0(\bar{x})$, but that $T_L^0(\bar{x}) \not\subset T^0(\bar{x})$.

As in the equality-constraint case, we now consider conditions under which we can define the tangent cone algebraically. To this end we define the following first-order constraint qualification.

Definition 5.2.3. (First-order constraint qualification for inequality constraints.) The first-order constraint qualification with respect to the set of inequality constraints $c(x) \geq 0$ holds at the feasible point x if, for any nonzero vector p such that $J_a(x)p \geq 0$, p is tangent to a differentiable feasible path starting at x .

If this constraint qualification holds, the regular and linearized tangent cones are identical sets. We are now free to consider the conditions under which there may or may not exist vectors p satisfying both $J_a(x)p \geq 0$ and $g(x)^T p < 0$. The existence of such a vector implies that x^* is not optimal, since every sufficiently small positive step along p retains feasibility and strictly reduces f . The following lemma formally states this result.

Lemma 5.2.1. Let x^* be a feasible point at which the first-order constraint qualification holds. Then x^* is a minimizer of NIP only if

$$g(x^*)^T p \geq 0 \quad \text{for all } p \text{ such that } J_a(x^*)p \geq 0, \quad (5.2.4)$$

where $J_a(x^*)$ denotes the Jacobian of active constraints at x^* .

This lemma provides necessary conditions for the optimality of x^* . The logical next question is: what properties of f and J_a imply that the crucial relationship (5.2.4) does (or does not) hold? For the case of equality constraints $c(x) = 0$, we were able to derive a simple test involving whether or not the gradient vector g is in the range of J^T . The situation for inequality constraints is less straightforward because of the wider class of feasible directions. We shall see that the analogous test for optimality with inequality constraints involves whether or not g can be written as a *nonnegative* linear combination of the *active* constraint gradients.

Conditions for existence of a feasible descent direction follow directly from *Farkas' Lemma*, a classical result published in 1902.

Lemma 5.2.2. (Farkas' Lemma.) Let A be an $m \times n$ matrix and g an n -vector. Then

$$g^T p \geq 0 \quad \text{for all } p \text{ such that } Ap \geq 0$$

if and only if

$$g = A^T \lambda \quad \text{for some } \lambda \geq 0.$$

Equivalently, a vector p satisfying $g^T p < 0$ and $Ap \geq 0$ exists if and only if g cannot be written as a nonnegative linear combination of the columns of A^T . ■

Proof. Here we provide a simple proof for the case where the rows of A are linearly independent. A proof for the general case is considerably more complicated and is given in the classnotes for Math 171a.

The “if” direction can be verified as follows. Since $Ap \geq 0$, $g = A^T \lambda$ and $\lambda \geq 0$, we have

$$g^T p = \lambda^T Ap = \sum_{i=1}^m \lambda_i (Ap)_i \geq 0.$$

Thus, if g is known to be a nonnegative linear combination of the columns of A^T , no vector p can exist that satisfies both relations $Ap \geq 0$ and $g^T p < 0$.

To prove the result in the “only if” direction, it is sufficient to show that there is a p satisfying $g^T p < 0$ and $Ap \geq 0$ whenever g is *not* a nonnegative linear combination of the columns of A^T .

First, we consider the case when g does not even lie in the range of A^T . In this case g can be written uniquely as

$$g = g_R + g_N, \quad g_N \neq 0,$$

where $g_R \in \text{range}(A^T)$, $g_N \in \text{null}(A)$, and $g_N^T g_R = 0$. The vector $p = -g_N$ satisfies $Ap = 0$, with

$$g^T p = (g_R + g_N)^T p = -(g_R + g_N)^T g_N = -\|g_N\|_2^2 < 0.$$

Hence we have shown that there exists a vector p such that $g^T p < 0$ and $Ap \geq 0$ as required.

Now assume that g lies in the range of A^T , but that there is at least one negative component of λ (say, λ_j). Since the rows of A are independent, λ is unique and *no* nonnegative λ exists such that $g = A^T \lambda$.

Suppose that the s th component of λ is negative. If A has independent rows, any m -vector can be expressed as a linear combination of the n columns of A . This implies that there exists at least one direction p such that

$$Ap = e_s, \tag{5.2.5}$$

where e_s is the s th coordinate vector. For the moment, assume that A comprises the *first* m rows of A , so that the s th row of A is a_s^T . For any p that satisfies (5.2.5), the relation $g = A^T \lambda$ implies that we can write

$$g(x^*)^T p = \lambda_s a_s^T p = \lambda_s. \tag{5.2.6}$$

Since $\lambda_s < 0$, (5.2.5) and (5.2.6) imply that p satisfies $g^T p < 0$ and $Ap \geq 0$. ■

In the context of NIP, this result was proved by making the assumption that at any feasible point x , there exists a feasible path with tangent p satisfying $J_a(x)p = e_s$, i.e.,

$$\nabla c_i(x)^T p = \begin{cases} 1 & \text{if } i = s; \\ 0 & \text{otherwise.} \end{cases}$$

This feasible path can be regarded as initially “staying on” all of the constraints in the active set except the s th, which must initially increase (since $\nabla c_s(x)^T p > 0$). Directions of this form play a fundamental role in the construction of algorithms for linear constraints.

The main first-order necessary condition for a solution of NIP can now be stated.

Theorem 5.2.2. (*First-order necessary conditions at a minimizer of NIP.*) Let x^* be a point such that $c(x^*) \geq 0$, with $c_a(x) = 0$. If the first-order constraint qualification holds at x^* , then x^* is a local minimizer of NIP only if there exists a vector λ_a^* such that

$$g(x^*) = J_a(x^*)^T \lambda_a^*, \quad \text{with } \lambda_a^* \geq 0. \tag{5.2.7}$$

Proof. Assume that x^* is a solution of NIP, and consider any nonzero vector p satisfying

$$J_a(x^*)p \geq 0. \tag{5.2.8}$$

The first-order constraint qualification implies that p is tangent to a feasible path emanating from x^* and contained in the feasible region. If $g(x^*)^T p \geq 0$, then Farkas' Lemma immediately implies the existence of $\lambda_a^* \geq 0$ satisfying (5.2.7). Therefore, we now suppose that there exists a nonzero p satisfying (5.2.8), but such that

$$g(x^*)^T p < 0. \quad (5.2.9)$$

The rate of change of f along the associated feasible path is $g(x^*)^T p$, so that (5.2.9) implies that f is strictly less than $f(x^*)$ at feasible points in every neighborhood of x^* , thereby contradicting the assumption that x^* is a local minimizer. Thus, there can be no vector p satisfying (5.2.8) and (5.2.9), and Farkas' Lemma implies that (5.2.7) holds. ■

The crucial difference from the analogous theorem for equality constraints is that *the Lagrange multipliers corresponding to active inequality constraints must be nonnegative*.

Definition 5.2.4. Any point x such that $g(x) = J_a(x)^T \lambda$ for some $\lambda \geq 0$, is known as a first-order KKT point of NIP. ■

It is easy to see that checking for optimality at a point x is the same as checking if the gradient $g(x)$ can be written as a nonnegative linear combination of the active-constraint gradients $\{\nabla c_i(x), i \in \mathcal{A}(x)\}$. An equivalent question is whether $g(x)$ lies in the set $\mathcal{C}_N(x)$ such that

$$\mathcal{C}_N(x) = \{y \in \mathbb{R}^n \mid y = J_a(x)^T \lambda, \quad \lambda \geq 0\}. \quad (5.2.10)$$

An important property of $\mathcal{C}_N(x)$ is that, given a point $y \in \mathcal{C}_N(x)$ and any nonnegative number γ , the point γy must also lie in $\mathcal{C}_N(x)$. This is the defining property of a type of set known as a *cone*, and is the reason why the set $\mathcal{C}_N(x)$ is called the *normal cone*. The normal cone at x can easily be shown to be closed and convex.

The main properties of the first-order necessary conditions are illustrated with a two-dimensional example. Consider the problem

$$\begin{aligned} & \underset{x_1, x_2}{\text{minimize}} && 2x_1^2 + x_2^2 + x_1x_2 + x_1 - x_2 \\ & \text{subject to} && (x_1 + 1)^2 + x_2^2 - 2 \geq 0, \\ & && x_1x_2 + 3x_2 - 3 \geq 0, \\ & && x_1 \geq 0, \quad x_2 \geq 0, \end{aligned} \quad (5.2.11)$$

which has derivatives

$$g(x) = \begin{pmatrix} 4x_1 + x_2 + 1 \\ 2x_2 + x_1 - 1 \end{pmatrix}, \quad \text{and} \quad J(x) = \begin{pmatrix} 2(x_1 + 1) & 2x_2 \\ x_2 & x_1 + 3 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

At the point $\bar{x} = (0, 1)^T$, we have $\mathcal{A}(\bar{x}) = \{1, 2, 3\}$, and derivatives

$$g(\bar{x}) = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \quad J_a(\bar{x}) = \begin{pmatrix} 2 & 2 \\ 1 & 3 \\ 1 & 0 \end{pmatrix}.$$

The first-order optimality conditions hold with $\bar{\lambda} = (\frac{1}{2}, 0, 1)^T$.

Any point at which $g \in \text{range}(A_a^T)$ and the active-constraint gradients are linearly dependent is known as a *degenerate point* of $\mathbb{F}$. At a degenerate point, Lagrange multipliers exist but are not unique. For example, the vector $\bar{\lambda} = (\frac{5}{4}, -\frac{1}{2}, 0)^T$ is also a solution of the system $g(\bar{x}) = J_a(\bar{x})^T \bar{\lambda}$ above. This example shows that if $J_a(x)$ has dependent rows at a first-order KKT point, then alternative multiplier vectors exist that have negative components. This does not contradict Theorem 5.2.2, which requires $\bar{\lambda}$ to be nonnegative for *some* vector $\bar{\lambda}$.

The point $\bar{x} = (0, 1)^T$ considered above is said to be a *vertex* of $\mathbb{F}$. A vertex of $\mathbb{F}$ is a feasible point at which there is at least one subset of n *linearly independent* constraint gradients, i.e., $J_a(x)$ has rank n . A vertex can exist only if $m \geq n$, and $\bar{x}$ can be a vertex only if at least n constraints are active at $\bar{x}$, i.e., if $m_a(\bar{x}) \geq n$. If *exactly* n constraints are active at a vertex, so that the matrix of active-constraint gradients is square and nonsingular, the vertex is said to be *nondegenerate*. If *more than* n constraints are active, the vertex is *degenerate*. In the plane, a degenerate vertex is the point of intersection of three or more lines.

When there are an infinite number of possible Lagrange multiplier vectors at a degenerate point, how can a nonnegative λ_a be found (if one exists)? One answer to this question can be found by examining the general proof of Farkas' lemma. Consider the following problem, which has the same constraints as problem (5.2.11),

$$\begin{aligned} \underset{x_1, x_2}{\text{minimize}} \quad & -(x_1 + 1)^2 x_2 \\ \text{subject to} \quad & (x_1 + 1)^2 + x_2^2 - 2 \geq 0, \\ & x_1 x_2 + 3x_2 - 3 \geq 0, \\ & x_1 \geq 0, \quad x_2 \geq 0. \end{aligned} \tag{5.2.12}$$

At the point $\bar{x} = (0, 1)^T$, the gradient $g(\bar{x}) = (-2, -1)^T$ lies in the column space of $J_a(\bar{x})^T$, but it can be verified by sketching the normal cone $\mathcal{C}_N(\bar{x})$ that $\bar{x}$ is not a KKT point. Unfortunately, this method for checking optimality cannot be used when there are more than three variables! Under these circumstances, a *computational* procedure is needed for checking optimality. Unfortunately, $J_a(x)$ has dependent rows at $\bar{x}$, so it is not possible to use the method for computing p described in the simplified proof of Farkas lemma (see Theorem 5.2.2). For example, the Lagrange multiplier vector $\bar{\lambda} = (-\frac{5}{4}, \frac{1}{2}, 0)^T$ has $\bar{\lambda}_s < 0$ for $s = 1$, yet there is no tangent p such that $J_a(\bar{x})p = e_s$. If such a vector existed, the system

$$\begin{pmatrix} 2 & 2 \\ 1 & 3 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

would require $p_1 = 0$, $p_1 = -3p_2$, and $p_1 + p_2 = \frac{1}{2}$, which is clearly impossible.

Next we describe a method for computing p based on the properties of the general proof of Farkas' lemma. This proof uses the result that if $g(x) \notin \mathcal{C}_N(x)$, then p can be defined as the normal vector of a particular hyperplane that separates $g(x)$ from $\mathcal{C}_N(x)$. This normal vector is computed in two steps. First, a point y_0 in $\mathcal{C}_N(x)$ *closest* to $g(x)$ is calculated by minimizing the function $\varphi(y) = \|y - g(x)\|_2^2$ over all $y \in \mathcal{C}_N(x)$. Then, p is defined such that $p = y_0 - g(x)$. The vector p defines the equation of a hyperplane

$p^T y = 0$ that splits $\mathbb{R}^n$ into two half spaces

$$H_1 = \{y \mid p^T y \geq 0\} \quad \text{and} \quad H_2 = \{y \mid p^T y < 0\},$$

with $\mathcal{C}_N(x) \subset H_1$ and $g(x) \in H_2$. Since $\mathcal{C}_N(x) \subset H_1$ and $\nabla c_i(x) \in \mathcal{C}_N(x)$ for all $i \in \mathcal{A}(x)$, it follows that $\nabla c_i(x) \in H_1$ and the definition of H_1 means that

$$\nabla c_i(x)^T p \geq 0, \quad i \in \mathcal{A}(x), \quad \text{or, equivalently,} \quad J_a(x)p \geq 0.$$

Similarly, since $g(x) \in H_2$, we have $g(x)^T p < 0$ and it follows that the hyperplane normal p is the required vector such that

$$g(x)^T p < 0, \quad \text{and} \quad J_a(x)p \geq 0.$$

This discussion implies that p may be defined as the vector $-g_J(x)$, where $g_J(x) = g(x) - J_a(x)^T \lambda_a^*$ with λ_a^* the solution of the bound-constrained linear least-squares problem

$$\underset{\lambda_a \in \mathbb{R}^{m_a}}{\text{minimize}} \quad \|g(x) - J_a(x)^T \lambda_a\|_2^2, \quad \text{subject to} \quad \lambda_a \geq 0. \quad (5.2.13)$$

The vector $g_J(x)$ is known as the *projected gradient* at x .

Theorem 5.2.3. *Let x^* be a point such that $c(x^*) \geq 0$, with $c_a(x) = 0$. If the first-order constraint qualification holds at x^* , then x^* is a local minimizer of NIP only if the projected gradient $g_J(x^*)$ is zero.*

The positive components of the optimal solution λ_a^* of problem (5.2.13) implicitly define a subset of the active constraints with strictly positive Lagrange multipliers.

The projected gradient for problem (5.2.12) is the vector $g_J = (-2, -1)^T$, which corresponds to the optimal solution $\lambda_a^* = 0$ in (5.2.13). It follows that since $g_J \neq 0$, the point $\bar{x}$ does not satisfy the first-order conditions for optimality. Moreover, the vector $p = (2, 1)^T$ is the direction of a feasible path along which f is initially decreasing. Note that in this case, $p = -g_J$ is an interior vector, i.e., $J_a(\bar{x})p > 0$.

Problem (5.2.13) is an example of a *convex program* (see §5.2.5). As the constraint gradients e_i comprise the m_a independent columns of the identity matrix, every KKT point is nondegenerate. Methods for solving least-squares problems of the form (5.2.13) will be considered later in this chapter.

5.2.2. Second-order conditions

To a large extent, the first-order necessary conditions can be derived by considering the equality-constraint problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad c_a(x) = 0. \quad (5.2.14)$$

The crucial difference is that additional feasible paths must be considered that permit movement off the constraints $c_a(x) = 0$. This implies that the set of feasible paths for the inequality constraints is larger than the set of paths for (5.2.14). For inequalities, the Lagrange multipliers provide more than just a stationarity condition for the Lagrangian, they also provide the sign of the initial derivative of $f(x(\alpha))$ along an interior feasible

path. In particular, if all the Lagrange multipliers are positive, then for any tangent p to an interior path, we have

$$g(x^*)^T p = \sum_{i=1}^m \lambda_i^* \nabla c_i(x^*)^T p > 0.$$

Since f is not stationary at $x(0)$ for such paths, the curvature of f does not play a role in the second-order conditions. This is not the case for feasible paths that stay *on* the constraint surface, i.e., paths such that $c_a(x(\alpha)) = 0$. In this case the sign of the curvature is crucial, and the “second-order part” of the second-order conditions for NIP are essentially the second-order necessary conditions for the equality-constraint problem (5.2.14).

However, the application of the second-order conditions for (5.2.14) requires the assumption of the constraint qualification for the equality constraints $c_a(x) = 0$. This amounts to requiring the following condition:

Definition 5.2.5. (*Second-order constraint qualification for inequality constraints.*) *The second-order constraint qualification with respect to the inequalities $c(x) \geq 0$ holds at the feasible point x if, for any nonzero vector p such that $J_a(x)p = 0$, p is tangent to a twice-differentiable path along which c_a is identically zero.*

It may be thought that the first-order constraint qualification for $c(x) \geq 0$ must imply the equality constraint qualification for $c_a(x) = 0$ (and hence the second-order constraint qualification for $c(x) \geq 0$). However, this is not necessarily the case. For example, suppose that the constraint qualification for equality constraints does *not* hold for problem (5.2.14) (see Definition 4.2.3). This implies that a vector p exists satisfying $J_a(x)p = 0$ that is *not* a tangent to a path $x(\alpha)$ such that $c_a(x(\alpha)) = 0$. However, this vector p may be a suitable tangent for an *interior* path $x(\alpha)$ such that $c_a(x(\alpha)) > 0$. This implies that the first-order constraint qualification may hold for $c(x) \geq 0$ when the constraint qualification for equality constraints does not hold for $c_a(x) = 0$. To illustrate how this might happen, consider the constraints

$$c_1(x) = x_1^2 + x_2 \geq 0, \quad c_2(x) = x_1^2 - x_2 \geq 0, \quad \text{and} \quad c_3(x) = x_3 \geq 0,$$

at the point $x^* = (0, 0)^T$. The Jacobian is

$$J(x) = \begin{pmatrix} 2x_1 & 1 & 0 \\ 2x_1 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The active set is $\mathcal{A}(x^*) = \{1, 2, 3\}$ and

$$J_a(x^*) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

If p is any tangent to a feasible path at x^* . Then the requirement that $J_a(x^*)p \geq 0$ imposes the restrictions $\nabla c_1(x^*)^T p = p_2 \geq 0$, $\nabla c_2(x^*)^T p = -p_2 \geq 0$, and $\nabla c_3(x^*)^T p = p_3 \geq 0$.

It follows that the set of tangent vectors consists of all vectors of the form $(p_1, 0, p_3)^T$. Each of these vectors is tangent to a path $x^*(\alpha)$ that points into the feasible region and it follows that the first-order constraint qualification is satisfied. However, the second-order constraint qualification does *not* hold. There are no feasible paths at x^* that satisfy $c_1(x) = 0$, $c_2(x) = 0$ and $c_3(x) = 0$, despite the fact that every vector of the form $p = (\gamma, 0, 0)^T$ satisfies $J_a(x^*)p = 0$.

Other examples indicate that the first- and second-order constraint qualifications are distinct, and neither implies the other. If both the first- and second-order constraint qualifications hold, second-order conditions can be derived directly from problem (5.2.14). As would be expected, these conditions involve the curvature of the Lagrangian restricted to the active constraints:

$$L_a(x, \lambda_a) \equiv f(x) - \lambda_a^T c_a(x).$$

In what follows, $H_a(x, \lambda)$ is used to denote the Hessian with respect to x of the Lagrangian $L_a(x, \lambda_a)$.

Theorem 5.2.4. *(Second-order necessary conditions for a local solution of NIP.) Let x^* be a point such that $c(x^*) \geq 0$, with $c_a(x) = 0$. If the first- and second-order constraint qualifications hold at x^* , then x^* is a local solution of NIP only if*

- (a) *there exists a vector λ_a^* such that $g(x^*) = J_a(x^*)^T \lambda_a^*$ with $\lambda_a^* \geq 0$; and*
- (b) *for the vector λ_a^* of part (a), the inequality $p^T H_a(x^*, \lambda_a^*) p \geq 0$ holds for all vectors p such that $J_a(x^*)p = 0$.*

Proof. Consider any nonzero vector p satisfying $J_a(x^*)p = 0$. Because of the second-order constraint qualification, p is tangent to a feasible path along which the constraints active at x^* remain identically zero. Exactly as in the proof of Theorem 4.2.4, analysis of the curvature of f along the path implies the desired result. ■

Each of the following conditions on $J_a(x)$ is sufficient to allow both the first- and second-order constraint qualification to hold:

- (i) The active constraints are linear, i.e., $c_a(x) = A_a x - b_a$ for some constant matrix A_a and constant vector b_a .
- (ii) The matrix $J_a(x)$ has full row rank, i.e., the active constraint gradients are linearly independent.
- (iii) There exists an “interior” tangent vector s , i.e., there exists a vector s such that $J_a(x)s > 0$. This condition is known as the Mangasarian-Fromovitz constraint qualification.

Sufficient conditions for x^* to be a strict local minimizer of NIP are given in the following theorem:

Theorem 5.2.5. *(Sufficient conditions for a local solution of NIP.) Let x^* be a point such that $c(x^*) \geq 0$, with $c_a(x) = 0$. Then x^* is a strict local minimizer of NIP if there exists a vector λ^* such that*

- (a) *$g(x^*) = J_a(x^*)^T \lambda_a^*$ with $\lambda_a^* > 0$; and*

(b) for the λ_a^* of part (a), the strict inequality $p^T H_a(x^*, \lambda_a^*) p > 0$ holds for all p such that $J_a(x^*)p = 0$. ■

Condition (a) of Theorem 5.2.5, which requires that the Lagrange multipliers corresponding to active constraints are strictly positive, is usually termed *strict complementarity*. The possibility of zero multipliers causes significant difficulties when analyzing the theoretical properties of algorithms for NIP. However, their effect on practical computation is usually less serious than the theory predicts, which is fortunate since zero multipliers are very common in practice. Zero multipliers are discussed further in the next section.

5.2.3. *Zero Lagrange multipliers

Zero Lagrange multipliers can occur in various situations. If x^* is a degenerate KKT point, the Lagrange multipliers are not unique, and at least one set of multipliers exist with zero values. Similarly, zero multipliers occur if a redundant constraint passes through a point satisfying the first-order necessary conditions. For example, consider minimizing $f(x) = x_1^2 - x_2^2$, subject to $x_2 \geq 0$. At $x^* = (0, 0)^T$, the gradient vanishes. Thus, $\lambda_a^* = 0$, $J_a(x^*) = \begin{pmatrix} 0 & 1 \end{pmatrix}$, one possible null space basis is $Z = (1, 0)^T$, and the reduced Hessian is positive definite. However, x^* is not a local minimizer, since any positive step along $(0, \delta)^T$, for $\delta > 0$, is a feasible direction that strictly reduces f (the problem has no bounded solution).

The second-order sufficient conditions of Theorem 5.2.5 principally involve the two conditions $\lambda_a^* > 0$ and $p^T H_a(x^*, \lambda_a^*) p > 0$. These conditions ensure that the objective function f initially increases along all feasible paths starting at x^* . The condition $(\lambda_a^*)_i > 0$ ensures that f initially increases along every *non-binding* feasible path. The condition $p^T H_a(x^*, \lambda_a^*) p > 0$ ensures that the curvature of f is initially positive for every *binding* feasible path.

In practice, many problems have zero Lagrange multipliers, and the requirement that every multiplier be positive is too restrictive. The derivation of more realistic conditions involves limiting the class of feasible paths that start at x^* . Consider the class of potential feasible paths in the presence of zero multipliers. It is unnecessary to impose conditions on the curvature of f along non-binding feasible paths for constraints with strictly positive multipliers. In this situation, the fact that the multiplier is positive already implies that f is initially increasing. However, it *is* necessary to impose a second-order condition along non-binding feasible paths for constraints with *zero* multipliers since the relation $g(x^*)^T p = 0$ is not sufficient to guarantee that f initially increases off the constraint.

These arguments imply that it is necessary to impose positive curvature only along feasible paths that are simultaneously *binding* for active constraints with positive multipliers, and *non-binding* for active constraints with zero multipliers. At any point x , let $\mathcal{A}_+(x)$ denote the set of indices of active constraints with *positive* Lagrange multipliers and let $J_+(x)$ denote the corresponding matrix of constraint gradients. Similarly, let $\mathcal{A}_0(x)$ denote the set of indices of active constraints with zero multipliers, and let $J_0(x)$ denote the associated matrix of constraint gradients. At any point satisfying the first-order necessary conditions of Theorem 5.2.2, every active constraint index must be in either $\mathcal{A}_+$ or $\mathcal{A}_0$. Given these quantities, we define the tangents associated with the set of feasible paths as:

$$T_+(x) = \{p \mid p \neq 0 \text{ is tangent to a feasible path with } J_+(x)p = 0 \text{ and } J_0(x)p \geq 0\}.$$

In effect, the sufficient conditions given in the next theorem require that f has positive curvature along every feasible path in this set.

Theorem 5.2.6. (*Sufficient Conditions with Zero Multipliers.*) Let x^* be a point such that $c(x^*) \geq 0$, with $c_a(x) = 0$. Then x^* is an isolated minimizer of NIP if there exists a vector λ^* such that

- (a) $g(x^*) = J_a(x^*)^T \lambda_a^*$ with $\lambda_a^* \geq 0$; and
- (b) for the λ_a^* of part (a), the strict inequality $p^T H_a(x^*, \lambda_a^*) p > 0$ holds for all vectors p such that $p \in T_+(x^*)$. ■

The difficulty with this theorem is that the set $T_+(x)$ cannot be defined computationally, which makes the result of only theoretical interest. However, it is possible to give slightly weaker conditions that can be verified using a simple linear algebraic procedure. Note that the set $T_+(x)$ is the intersection of the sets $\{p \mid J_+(x)p = 0\}$ and $\{p \mid J_0(x)p \geq 0\}$, which implies that $T_+(x)$ must be contained in each of them. By working with the set $\{p \mid J_+(x)p = 0\}$ (which is larger and hence more restrictive than $T_+(x)$), we can derive conditions that are computationally verifiable.

Theorem 5.2.7. (*Sufficient conditions for a solution of NIP.*) A feasible point x^* is an isolated minimizer of NIP if there exists a vector λ^* such that

- (a) there exists a vector λ_a^* such that $g(x^*) = J_a^T \lambda_a^*$ with $\lambda_a^* \geq 0$; and
- (b) for the λ_a^* of part (a), the strict inequality $p^T H_a(x^*, \lambda_a^*) p > 0$ holds for all nonzero p such that $J_+(x^*)p = 0$;

Condition (a) is equivalent to the condition that $Z_+(x^*)^T g(x^*) = 0$, where $Z_+(x^*)$ denotes a basis for the null space of $J_+(x^*)$. Similarly, condition (b) is equivalent to positive-definiteness of the reduced Hessian $Z_+(x^*)^T H_a(x^*, \lambda_a^*) Z_+(x^*)$. Note that when these sufficient conditions hold, x^* is not only optimal, but is also *locally unique*, i.e., $f(x^*) < f(x)$ for all feasible x in a neighborhood of x^* ($x \neq x^*$).

5.2.4. Optimality conditions in complementary slackness form

In some circumstances, it is useful to define a Lagrange multiplier for *every* constraint, with the convention that the multiplier corresponding to an inactive constraint is zero. Let λ^* denote the “extended” multiplier vector. The necessary condition (5.2.7) then becomes

$$g(x^*) = J(x^*)^T \lambda^*, \quad \lambda^* \geq 0,$$

where

$$\lambda_i^* c_i(x^*) = 0, \quad i = 1, 2, \dots, m.$$

The conditions $\lambda_i^* c_i(x^*) = 0$, which force at least one of $c_i(x^*)$ and λ_i^* to be zero for every i , is called a *complementarity condition*. Thus, whenever a given component of λ is *positive* (“slack”), the corresponding component of c must be *zero* (and vice versa). This condition can be written conveniently as $\Lambda^* C(x^*) = 0$, where Λ denotes the diagonal matrix with entries $(\lambda_1^*, \lambda_2^*, \dots, \lambda_m^*)$ and $C(x) = \text{diag}(c_1(x), c_2(x), \dots, c_m(x))$.

We say that *strict complementarity* holds between two nonnegative m -vectors λ and c if, for every i , either $\lambda_i > 0$ and $c_i = 0$, or else $\lambda_i = 0$ and $c_i > 0$. Strict complementarity means that *both λ_i and c_i cannot be zero*. Without strict complementarity, the condition $\lambda_i = c_i = 0$ is possible.

Let $Z(x^*)$ denote a matrix whose columns form a basis for the null-space of $A^*J(x^*)$. If a suitable constraint qualification holds at x^* , the following conditions can be shown to be necessary for the point x^* to be a local solution of NIP:

$$C(x^*) \geq 0; \quad (5.2.15a)$$

$$g(x^*) = J(x^*)^T \lambda^* \quad \text{for some } \lambda^*; \quad (5.2.15b)$$

$$A^*C(x^*) = 0, \quad (5.2.15c)$$

$$A^* \geq 0, \quad (5.2.15d)$$

$$Z(x^*)^T H(x^*, \lambda^*) Z(x^*) \text{ positive semi-definite.} \quad (5.2.15e)$$

Condition (5.2.15c), which forces at least one of $c_i(x^*)$ and λ_i^* to be zero for every i , is another form of the *complementarity* condition discussed earlier. This condition implies that if $c_i(x^*) > 0$, i.e., constraint i is *inactive* at x^* , then λ_i^* must be zero.

Second-order sufficient conditions for x^* to be an isolated local solution of NIP are: (i) a suitable constraint qualification applies at x^* (for example, the gradients of the active constraints at x^* are linearly independent); (ii) conditions (5.2.15) are satisfied; and (iii) the following strengthened versions of (5.2.15d–5.2.15e) hold:

$$\lambda_i^* > 0 \text{ if } c_i(x^*) = 0; \quad \text{and} \quad Z(x^*)^T H(x^*, \lambda^*) Z(x^*) \text{ positive definite.} \quad (5.2.16)$$

The strict inequality on λ_i^* is called *strict complementarity*, and holds when all Lagrange multipliers associated with active constraints are positive. The property of strict complementarity is often assumed for the theoretical analysis of algorithms.

Consider the nonlinear function $F : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^{n+m}$ such that

$$F(x, \lambda) = \begin{pmatrix} g(x) - J(x)^T \lambda \\ AC(x)e \end{pmatrix},$$

where $A = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$, $C(x) = \text{diag}(c_1(x), c_2(x), \dots, c_m(x))$ and e is the vector of ones $(1, 1, \dots, 1)^T$. Note (x^*, λ^*) is a zero of the nonlinear system

$$F(x, \lambda) = 0.$$

This system is analogous to the system $\nabla L(x, \lambda) = 0$ in the equality constrained case.

Theorem 5.2.8. *If the second-order sufficiency conditions (5.2.16) hold, and the active-constraint gradients are linearly independent at x^* , then the Jacobian $F'(x, \lambda)$ is nonsingular at (x^*, λ^*) .*

Proof. The Jacobian of F is given by

$$F'(x, \lambda) = \begin{pmatrix} H(x) - \sum_{i=1}^m \lambda_i H_i(x) & -J(x)^T \\ AJ(x) & C(x) \end{pmatrix}.$$

Hence we must show that if $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^m$, then $(u, v) = 0$ is the only solution of the system

$$\begin{pmatrix} H(x^*, \lambda^*) & -J(x^*)^T \\ \Lambda^* J(x^*) & C(x^*) \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = 0. \quad (5.2.17)$$

This system may be written as

$$H(x^*, \lambda^*)u - J(x^*)^T v = 0 \quad (5.2.18a)$$

$$\Lambda^* J(x^*)u + C(x^*)v = 0. \quad (5.2.18b)$$

Consider the i th equation of (5.2.18b). If $i \in \mathcal{A}(x^*)$, then strict complementarity implies that $c_i(x^*)v_i = 0$, and hence that $\lambda_i^* \nabla c_i(x^*)^T u = 0$. Since $\lambda_i^* > 0$ for $i \in \mathcal{A}(x^*)$, we have

$$\nabla c_i(x^*)^T u = 0, \quad \text{for } i \in \mathcal{A}(x^*). \quad (5.2.19)$$

If $i \notin \mathcal{A}(x^*)$, then strict complementarity implies that $\lambda_i^* = 0$, and again we have $c_i(x^*)v_i = 0$. In this case, however, we must have $v_i = 0$ since $c_i(x^*) > 0$.

The arguments above imply that $u^T J(x^*)^T v = 0$, $v^T \Lambda^* J(x^*)u = 0$, and $v^T C(x^*)v = 0$. Hence, pre-multiplying (5.2.18a) by u^T gives

$$u^T H(x^*, \lambda^*)u = u^T (H(x^*) - \sum_{i \in \mathcal{A}} \lambda_i^* H_i(x^*))u = 0.$$

The assumption that $u^T H(x^*, \lambda^*)u > 0$ for all nonzero u such that $\nabla c_i(x^*)^T p = u$, $i \in \mathcal{A}(x^*)$, implies that $u = 0$, from (5.2.19).

It only remains to be shown that the elements of v corresponding to the active constraints are zero. From (5.2.18b), we have

$$J(x^*)^T v = \sum_{i \in \mathcal{A}} \nabla c_i(x^*)v_i = 0.$$

The assumption that the active constraint gradients are linearly independent implies $v_i = 0$, for $i \in \mathcal{A}(x^*)$. ■

5.2.5. Convex problems

Convex programming problems form an important class of optimization problem, of which linear programming is the most obvious instance. As we shall see, many complications that arise for general optimization problems disappear in the presence of convexity.

Definition 5.2.6. (*Convex set.*) The set S is convex if, for every x_1 and x_2 in S , and for all θ satisfying $0 \leq \theta \leq 1$, the point $z = (1 - \theta)x_1 + \theta x_2$ is also in S .

Definition 5.2.7. (*Convex and concave functions.*) The function $f(x)$, defined for x in a nonempty open convex set S , is convex if, for every two points x_1 and x_2 in S and every θ satisfying $0 \leq \theta \leq 1$,

$$f((1 - \theta)x_1 + \theta x_2) \leq (1 - \theta)f(x_1) + \theta f(x_2). \quad (5.2.20)$$

(If the set S is not specified, it is assumed to be $\mathbb{R}^n$.) The function f is concave if $-f$ is convex. The function f is strictly convex if the inequality in (5.2.20) is strict when $x_1 \neq x_2$ and $0 < \theta < 1$. ■

Several useful results associated with convexity are:

1. The intersection of a finite number of convex sets is convex;
2. All level sets of a convex function are convex;
3. Given a set of convex functions $\{\varphi_i(x)\}$, $i = 1, \dots, m$, the set of points satisfying $\varphi_i(x) \leq 0$ is convex;
4. The smooth function $f(x)$, defined for x in an open convex set S , is convex if its Hessian matrix $H(x)$ is positive semi-definite for all $x \in S$, and strictly convex if $H(x)$ is positive definite for all $x \in S$.

Convex programs are constrained optimization problems with important special properties. It should be stressed that the only *equality* constraints permitted in a convex program are linear constraints.

Definition 5.2.8. (*Convex program.*) The problem of minimizing $f(x)$ subject to the linear equality constraints $Ax = b$ and the inequality constraints $c_i(x) \geq 0$, $i = 1, \dots, m$, is a convex program if $f(x)$ is convex and $-c_i(x)$ is convex for $i = 1, \dots, m$. ■

A slight complication is that our generic form for inequality constraints involves a “greater than” relation ($c_i(x) \geq 0$) for expositional convenience. Unfortunately, an optimization problem with constraints in this form is a convex program only if each *negative* constraint function $-c_i(x)$ is convex, i.e., if $c_i(x)$ itself is concave. Hence minus signs appear throughout our discussion of the constraints in convex programs.

Using these definitions, it is easy to see that a linear function is convex (and also concave), and that a linear programming problem is a convex program. Two properties that are important in interior methods for linear programming are stated formally in the following theorems; see Fiacco and McCormick (1968) or Fletcher (1987) for details.

Theorem 5.2.9. If x^* is a local constrained minimizer of a convex programming problem, it is also a global constrained minimizer. Further, the set of minimizers of a convex program is convex.

Theorem 5.2.10. If the optimization problem NIP is a convex program, and if x^* satisfies the feasibility and first-order necessary conditions of Theorem 5.2.2, then x^* is a global minimizer of NIP.

5.3. General Nonlinear Constraints

We now consider methods for problems with nonlinear constraints:

$$\begin{array}{ll} \text{NIP} & \begin{array}{l} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) \\ \text{subject to} \quad c(x) \geq 0, \end{array} \end{array}$$

where $c(x)$ has m components $c_i(x)$, and f and $\{c_i(x)\}$ are smooth functions. The matrix $J(x)$ will denote the Jacobian of the constraint vector $c(x)$. Unless stated otherwise,

we assume throughout that a Karush-Kuhn-Tucker (KKT) point (x^*, λ^*) exists for NIP, satisfying the first-order optimality conditions:

$$c(x^*) \geq 0, \quad \lambda^* \geq 0, \quad c(x^*)^T \lambda^* = 0, \quad J(x^*)^T \lambda^* = g(x^*). \quad (5.3.1)$$

Let $v_i(x)$ denote the magnitude of the violation of the i th constraint at the point x , i.e.,

$$v_i(x) = \max\{-c_i(x), 0\} = -\min\{c_i(x), 0\}.$$

Then the vector $v(x)$ with components $v_i(x)$ is known as the vector of violations at x . By definition, $v(x) = 0$ for all $x \in \mathbb{F}$. The vector-valued function $v(x)$ is continuous, but it is not differentiable at any point such that $c_i(x) = 0$ (this is why it is not possible to derive the optimality conditions for inequality constraints by treating the equality constraint problem $\min f(x)$ subject to $v(x) = 0$).

Many methods for inequality constraints may be motivated by considering a form of NIP in which the nonlinear inequality constraints are converted to equalities using a set of nonnegative *slack variables*. A slack variable s_i can be used to convert the inequality constraint $c_i(x) \geq 0$ to an equality constraint by means of the transformation:

$$c_i(x) \geq 0 \quad \text{if and only if} \quad c_i(x) - s_i = 0, \quad s_i \geq 0.$$

This gives the following problem, which is equivalent to NIP:

$$\begin{array}{ll} \text{NIP}' & \begin{array}{l} \text{minimize} \quad f(x) \\ x \in \mathbb{R}^n, s \in \mathbb{R}^m \\ \text{subject to} \quad c(x) - s = 0, \quad s \geq 0. \end{array} \end{array}$$

If (x^*, s^*) denotes an optimal solution of NIP', then the optimality conditions are

$$c(x^*) - s^* = 0, \quad \lambda^* \geq 0, \quad s^*{}^T \lambda^* = 0, \quad J(x^*)^T \lambda^* = g(x^*). \quad (5.3.2)$$

These conditions imply that a nonlinear constraint $c_i(x) \geq 0$ being active for NIP is equivalent to its associated bound constraint being active in NIP'.

Note that problem NIP' has two types of constraint: the nonlinear equality constraints $c(x) - s = 0$ and the nonnegativity constraints $s \geq 0$. It will be shown that the equalities may be treated using methods already discussed for equality constraints.

Problems of the form NIP' are especially convenient when the x -variables are already subject to upper and lower bounds, which is a very common situation in practice. In this case, it is best to treat the bounds specially and not include them in the general inequalities $c(x) \geq 0$.

The introduction of slack variables allows the use of a smooth variant of the vector of violations. Consider the vector-valued function $v(x, s)$ such that

$$v(x, s) = c(x) - s.$$

Then $v(x^*, s^*) = 0$ and $v(x, s)$ has the same degree of differentiability as $c(x)$. In particular

$$v'(x, s) = \begin{pmatrix} J(x) & -I \end{pmatrix} \quad \text{and} \quad \nabla^2 v_i(x, s) = \begin{pmatrix} \nabla^2 c_i(x) & 0 \\ 0 & 0 \end{pmatrix}. \quad (5.3.3)$$

In many cases the slack variable transformation can be done *implicitly*. Many algorithms are derived by considering problem NIP', yet no explicit values of the slack variables are computed.

5.4. Penalty-Function Methods

In applying a penalty-function approach to problem NIP, the motivation is identical to that in the equality-constrained case, namely to add a weighted penalty for infeasibility. Thus, the quadratic or ℓ_1 penalty function may be used as in §4.6.6 and §4.6.1, but only the *violated* constraints are included in the penalty term.

The quadratic and ℓ_1 penalty functions for NIP may be written as

$$P_2(x, \rho) = f(x) + \frac{1}{2}\rho\|v(x)\|_2^2 \quad \text{and} \quad P_1(x, \rho) = f(x) + \rho\|v(x)\|_1,$$

where $v(x)$ is the vector of constraint violations at x . Note that, unlike the equality-constraint case, $P_2(x, \rho)$ has a discontinuous second derivative at any point such that $c_i(x) = 0$.

The convergence of the quadratic and ℓ_1 penalty function methods applied to NIP can be proved as for the equality-constrained problem. (See §4.6.1 and §4.6.6.) For $P_2(x, \rho)$, the existence of a trajectory $x(\rho)$, with a multiplier function $\lambda(\rho)$, can be demonstrated as in the equality constraint case, subject to the assumption of strict complementarity (see §4.6.1 and §5.2). An important property of penalty-function methods can be derived from the behavior of the Lagrange multiplier estimates obtainable at $x(\rho)$. (See (4.6.7).) Since

$$-\rho c_i(x(\rho)) \rightarrow \lambda_i^* \quad \text{and} \quad \lambda_i^* \geq 0,$$

it is clear that a constraint active at x^* with a positive multiplier will be strictly violated at $x(\rho)$ for sufficiently large ρ . It follows that for increasing ρ , the quadratic penalty function method converges to x^* with $c_i(x(\rho)) < 0$.

This property has two implications in practice. First, the set of violated constraints can be used as a prediction of the active set at x^* . Second, the discontinuity in the Hessian is unlikely to affect the unconstrained minimizations. In practice, however, a small tolerance is usually included in the definition of a “violated” constraint to avoid difficulties when strict complementarity does not hold. For example, we may define

$$v_i(x) = -\min\{c_i(x), \epsilon\}.$$

where ϵ is some small positive tolerance.

The discontinuity in the derivatives can be removed altogether if the penalty subproblem is changed to be

$$\underset{x \in \mathbb{R}^n, s \in \mathbb{R}^m}{\text{minimize}} \quad f(x) + \frac{1}{2}\rho\|v(x, s)\|_2^2 \quad \text{subject to} \quad s \geq 0.$$

The price that must be paid for smoothness is an increase in the complication of the subproblem. However, as we have already remarked, this increase is marginal if there are already upper and lower bounds on the x variables.

5.5. Barrier-Function Methods

Suppose that we wish to minimize $f(x)$ subject to a set of inequality constraints $c_i(x) \geq 0$, $i = 1, \dots, m$. If the constraints affect the solution, either an unconstrained minimizer of $f(x)$ is infeasible (for example, when minimizing x^2 subject to $x \geq 1$), or else $f(x)$ is

unbounded below when the constraints are removed (for example, when minimizing x^3 subject to $x \geq 1$). Consequently, if an optimization method tries to achieve a “large” reduction in the objective function from its value at a feasible point, the iterates tend to move *outside* the feasible region. In fact, many popular algorithms for nonlinearly constrained optimization typically produce infeasible iterates that approach feasibility only in the limit.

When feasibility at intermediate points is essential—for example, in practical problems where the objective function is meaningless unless the constraints are satisfied—it seems desirable for iterates to approach the constrained solution from the *interior* of the feasible region. *Barrier methods* constitute a well known class of methods with this property.

For the feasible region $\mathbb{F}$ associated with our generic optimization problem, the subset of points in $\mathbb{F}$ for which all the constraint functions are *strictly positive* is denoted by $\text{Sint}(\mathbb{F})$ and defined as

$$\text{Sint}(\mathbb{F}) = \{x \mid c_i(x) > 0, \quad i = 1, \dots, m\}. \quad (5.5.1)$$

A point x in $\text{Sint}(\mathbb{F})$ is said to be *strictly feasible*.

Although the sets $\text{Sint}(\mathbb{F})$ and $\text{int}(\mathbb{F})$ are identical in many instances, they can be different. For example, consider the single constraint $x_1^2 + x_2^2 \geq 0$ in $\mathbb{R}^2$. The corresponding feasible region $\mathbb{F}$ includes all of $\mathbb{R}^2$; consequently, every point in $\mathbb{R}^2$ is an interior point, and $\text{int}(\mathbb{F}) = \mathbb{R}^2$. In contrast, the set $\text{Sint}(\mathbb{F})$ includes all points in $\mathbb{R}^2$ *except* the origin.

Barrier methods may be applied only to inequality constraints for which strictly feasible points exist. This property does not hold for all inequality constraints, even if the feasible region is nonempty; for example, consider the constraints $x_1 + x_2 \geq 0$ and $-x_1 - x_2 \geq 0$, for which the feasible region consists of the line $\{x_1 + x_2 = 0\}$.

Given an initial strictly feasible point and mild assumptions about the feasible region, strict feasibility can be retained by minimizing a composite function consisting of the original objective $f(x)$ plus a positive multiple of an infinite “barrier” at the boundary of $\text{Sint}(\mathbb{F})$. The most effective methods for unconstrained optimization (such as Newton’s method) require differentiability. A suitable barrier term is therefore composed of functions that are smooth at strictly feasible points, but contain a positive singularity if any constraint is zero. Under these conditions, a minimizer of the composite function must occur at a strictly feasible point.

When the barrier term is heavily weighted, a minimizer of the composite function will lie, informally speaking, “far away” from the boundary. If the coefficient of the barrier term is reduced, the singularity becomes less influential, except at points near the boundary; minimizers of the composite function can then move closer (but not “too close”) to the boundary. The weight on the barrier term thus tends to regulate the distance from the iterates to the boundary. In the parlance of modern interior methods, the barrier term forces the iterates to remain *centered* in the strictly feasible region.

As the factor multiplying the barrier term decreases to zero, intuition suggests that minimizers of the composite function will converge to a constrained solution x^* that lies on the boundary of $\text{Sint}(\mathbb{F})$. We shall see later that this intuition can be verified rigorously under reasonably mild conditions.

We stress that there is ample room for many formulations of a “barrier function”, as indicated by the range of definitions in Fiacco and McCormick (1968) and in Nesterov and Nemirovsky (1989). Other varieties of composite functions—called “potential” and

“centering” functions—have also been proposed for use in interior methods; see, for example, Sonnevend (1986) and Gonzaga (1992). Karmarkar’s original (1984) LP algorithm included a logarithmic potential function. The method of centers of Huard (1967) imposes an additional constraint at each iteration based on the current value of the objective function; see Renegar (1988) for an LP method based on this idea.

In all cases, the composite functions display a common motivation of simultaneously reflecting the objective function (thereby encouraging its reduction) as well as forcing iterates to stay “nicely centered” in the feasible region. They differ, however, in the balance of these sometimes conflicting aims.

5.5.1. The logarithmic barrier function

For simplicity, we discuss only the simplest barrier function based on a logarithmic singularity, which was not only the most popular in the 1960s, but also has received substantial attention since 1984. The logarithmic barrier function was first defined by Frisch in 1955, and was extensively studied and analyzed during the 1960s. Detailed theoretical discussions of classical barrier methods, along with historical background, are given in Fiacco and McCormick (1968) and Fiacco (1979).

The *logarithmic barrier function* associated with minimizing $f(x)$ subject to $c(x) \geq 0$ is

$$B(x, \mu) = f(x) - \mu \sum_{i=1}^m \ln c_i(x), \quad (5.5.2)$$

where the *barrier parameter* μ is strictly positive. (When the meaning is clear, we may write B with a single argument μ or without arguments.) Since the logarithm is undefined for nonpositive arguments, the logarithmic barrier function is defined only in $\text{Sint}(\mathbb{F})$.

The intuitive motivation for a barrier method is that we seek unconstrained minimizers of $B(x, \mu)$ for values of μ decreasing to zero. If the solution x^* of the constrained problem lies on the boundary and exact arithmetic is used, a barrier method can never produce the exact solution. Barrier methods consequently terminate when the current iterate satisfies some approximation to the desired optimality conditions. “Classical” barrier algorithms as well as many recent interior methods have the following form:

Algorithm 5.5.1. Classical Barrier Algorithm.

Choose x_0 so that $c(x_0) > 0$; Choose μ ($\mu > 0$); γ ($0 \leq \gamma \leq 1$);

Set $k = 0$;

while not converged do

 Compute an unconstrained minimizer $x(\mu_k)$ of $B(x, \mu_k)$;

$x_{k+1} \leftarrow x(\mu_k)$;

$\mu \leftarrow \gamma \mu$;

$k \leftarrow k + 1$;

end do

In practice, the calculation of $x(\mu_k)$ is carried out approximately, and only a few iterations of an unconstrained method may be performed before the barrier parameter is updated. In the theoretical results given here, we assume that $x(\mu_k)$ is an exact unconstrained minimizer.

The next two sections provide a rigorous foundation for the generic approach, including the assumptions necessary to make it succeed in converging to a solution x^* of the original constrained problem. After establishing local convergence properties, we return in §5.5.2 to a more detailed analysis of the sequence of barrier minimizers.

5.5.2. The barrier trajectory

In this section, we describe conditions under which a sequence $x(\mu)$ of barrier minimizers not only converges to x^* , but also defines a smooth path (the “barrier trajectory”) that is nontangential to the active constraint gradients.

Discussions of the logarithmic barrier function involve special diagonal matrices related to vectors, for which the following notation has become popular. When a lower-case letter refers to a vector, its upper-case version means the diagonal matrix of comparable dimension whose (i, i) element is the i th component of the vector. For example, $C(x)$ denotes the $m \times m$ diagonal matrix of constraint values $\{c_i(x)\}$:

$$C(x) = \text{diag}(c_i(x)) = \begin{pmatrix} c_1(x) & & & \\ & c_2(x) & & \\ & & \ddots & \\ & & & c_m(x) \end{pmatrix},$$

and $C(x)^{-1}$ is the diagonal matrix whose i th element is $1/c_i(x)$. Using this convention, we have the general relation $C(x)e = c(x)$, where e denotes the vector of appropriate dimension whose components are all equal to one, $e = (1, 1, \dots, 1)^T$.

The gradient of $B(x, \mu)$ with respect to x is

$$\nabla B(x, \mu) = g(x) - \sum_{i=1}^m \frac{\mu}{c_i(x)} \nabla c_i(x) = g(x) - \mu J(x)^T C(x)^{-1} e, \quad (5.5.3)$$

where $\nabla c_i(x)$ is the gradient of $c_i(x)$ and $J(x)$ is the Jacobian of $c(x)$. (Recall that the constraint gradients are the transposed rows of $J(x)$ and hence the *columns* of $J(x)^T$.) The Hessian of $B(x, \mu)$ is

$$\begin{aligned} \nabla^2 B(x, \mu) &= H(x) - \sum_{i=1}^m \frac{\mu}{c_i(x)} H_i(x) + \sum_{i=1}^m \frac{\mu}{c_i(x)^2} \nabla c_i(x) \nabla c_i(x)^T \\ &= H(x) - \sum_{i=1}^m \frac{\mu}{c_i(x)} H_i(x) + \mu J(x)^T C(x)^{-2} J(x). \end{aligned}$$

The point $x(\mu)$ is an unconstrained minimizer of $B(x, \mu)$ only if the gradient vanishes at $x(\mu)$. Substituting from (5.5.3), the following relation must hold at $x(\mu)$:

$$g(\mu) = \sum_{i=1}^m \nabla c_i(\mu) \frac{\mu}{c_i(\mu)}, \quad (5.5.4)$$

where the argument μ denotes evaluation at $x(\mu)$. Since $\mu > 0$ and $c_i(\mu) > 0$, it follows that the gradient of f at $x(\mu)$ is a *positive* linear combination of the gradients of *all* the constraints.

From §5.2.4 the first-order optimality conditions for nonlinear constraints are

$$g(x^*) = J(x^*)^T \lambda^* = \sum_{i=1}^m \nabla c_i(x^*) \lambda_i^*, \quad (5.5.5)$$

where $\lambda_i^* \geq 0$ and $\lambda_i^* c_i(x^*) = 0$, $i = 1, \dots, m$. At x^* , the gradient of f is thus a nonnegative linear combination of all the constraint gradients, where inactive constraints have zero multipliers.

The similar forms of (5.5.4) and (5.5.5) reveal that the i th coefficient $\mu/c_i(\mu)$ in the linear combination (5.5.4) is directly analogous to the i th Lagrange multiplier λ_i^* . When standard sufficient optimality conditions hold at x^* and the gradients of the active constraints are linearly independent, the multiplier estimates $\mu/c_i(\mu)$ do indeed converge to λ_i^* . In fact, under these conditions a differentiable curve $x(\mu)$ of barrier minimizers, parametrized by μ , exists near $\mu = 0$ and converges to x^* . This curve of minimizers is called the *barrier trajectory*; in linear programming, it is usually known as the *central path*. Its existence and properties define the broad class of “path-following” algorithms that attempt to follow the trajectory to the solution.

Theorem 5.5.1. *Consider the problem of minimizing $f(x)$ subject to $c_i(x) \geq 0$, $i = 1, \dots, m$, where $m \geq 1$. Let $\mathbb{F}$ denote the feasible region, and assume that $\text{Sint}(\mathbb{F})$ is nonempty. Assume further that x^* is a local constrained minimizer at which*

- (a) $g(x^*) = J(x^*)^T \lambda^*$, with $\lambda_i^* c_i(x^*) = 0$;
- (b) $\lambda_i^* > 0$ if $c_i(x^*) = 0$;
- (c) there exists $\alpha > 0$ such that $p^T H(x^*, \lambda^*) p \geq \alpha \|p\|^2$ for all p satisfying $J_a(x^*) p = 0$, where $J_a(x^*)$ denotes the Jacobian of the active constraints at x^* and $H(x, \lambda)$ is the Hessian of the Lagrangian function (see (4.2.10));
- (d) the gradients of the active constraints at x^* are linearly independent.

Consider a logarithmic barrier method in which $B(x, \mu_k)$ is minimized for a sequence of positive values $\{\mu_k\}$ converging monotonically to zero as $k \rightarrow \infty$. Then

- (i) there is at least one subsequence of unconstrained minimizers of the barrier function $B(x, \mu_k)$ converging to x^* ;
- (ii) For such a convergent subsequence $\{x_k\}$,

$$\lim_{k \rightarrow \infty} \mu_k / c_i(x_k) = \lambda_i^*;$$

- (iii) for sufficiently large k , the Hessian matrix $\nabla^2 B(x_k, \mu_k)$ is positive definite;
- (iv) a unique, continuously differentiable vector function $x(\mu)$ of unconstrained minimizers of $B(x, \mu)$ exists in a neighborhood of $\mu = 0$;
- (v) $\lim_{\mu \rightarrow 0} x(\mu) = x^*$. ■

We have now verified the existence of both the barrier trajectory $x(\mu)$ and the associated multiplier estimate $\lambda(\mu)$. A remaining question involves existence and differentiability of the trajectory at x^* itself. For sufficiently small μ , the following $n + m$ equations are satisfied identically at every pair $(x(\mu), \lambda(\mu))$ on the trajectory:

$$g(x) - J(x)^T \lambda = 0; \quad (5.5.6a)$$

$$\lambda_i c_i(x) = \mu, \quad i = 1, \dots, m. \quad (5.5.6b)$$

If we treat the multipliers $\lambda(\mu)$ as *separate variables*, (5.5.6) can be viewed as a system of nonlinear equations in the $n + m$ variables (x, λ) . The Jacobian matrix of this system is given by

$$K = \begin{pmatrix} H(\mu) - \sum \lambda_i(\mu) H_i(\mu) & -J(\mu)^T \\ A(\mu)J(\mu) & C(\mu) \end{pmatrix}, \quad (5.5.7)$$

where A and C are diagonal matrices corresponding to λ and c .

We can again apply the implicit function theorem to deduce the existence of a differentiable trajectory $(x(\mu), \lambda(\mu))$ at (x^*, λ^*) if the matrix (5.5.7) is nonsingular at $\mu = 0$. Let k^* denote the limiting version of (5.5.7):

$$k^* = \begin{pmatrix} H(x^*, \lambda^*) & -J(x^*)^T \\ \lambda^* J(x^*) & C(x^*) \end{pmatrix}.$$

This matrix is nonsingular (see Theorem 5.2.8) and the implicit function theorem applied to (5.5.6) thus implies that the trajectory $x(\mu)$ exists and is differentiable at x^* .

The approach of the trajectory $x(\mu)$ to x^* can be analyzed as follows. Let $\dot{x}$ denote $dx(\mu)/d\mu$, with a similar meaning for $\dot{\lambda}$. Differentiating (5.5.6) with respect to μ , we see that, for sufficiently small μ , $x(\mu)$ and $\lambda(\mu)$ satisfy the system of differential equations

$$\begin{pmatrix} H - \sum \lambda_i H_i & -J^T \\ \lambda J & C \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{\lambda} \end{pmatrix} = \begin{pmatrix} 0 \\ e \end{pmatrix}, \quad (5.5.8)$$

with initial conditions $x(0) = x^*$ and $\lambda(0) = \lambda^*$.

Let y denote the vector $\dot{x}$ evaluated at $\mu = 0$, i.e., y is the tangent to the barrier trajectory at x^* . For an *active* constraint i , the second set of equations in (5.5.8) reveals that

$$\lambda_i^* \nabla c_i(x^*)^T y = 1, \quad \text{so that} \quad \nabla c_i(x^*)^T y = \frac{1}{\lambda_i^*} \quad \text{and} \quad J_a y = (\hat{A}^*)^{-1} e. \quad (5.5.9)$$

The assumption of strict complementarity means that $\lambda_i^* \neq 0$ for any active constraint. Hence relation (5.5.9) shows that the barrier trajectory approaches x^* *nontangentially* with respect to the active constraints, i.e., the iterates do not converge “along” the boundary.

Assume that constraint i is active, and let θ_i denote the angle between y and the normal to constraint i at x^* . It follows from (5.5.9) that

$$\cos \theta_i \sim \frac{1}{\|\nabla c_i(x^*)\| \lambda_i^*}. \quad (5.5.10)$$

If all active constraint gradients are approximately equal in norm, relation (5.5.10) shows that the approach of barrier trajectory to x^* is “closer to tangential” for active constraints with larger multipliers.

The nontangential property fails to hold without strict complementarity, even if the active constraint gradients are linearly independent. A complete analysis of this case is given in Jittorntrum (1978).

5.5.3. Properties of the barrier Hessian

The Hessian matrices H_b of the barrier function display a special structure as $\mu \rightarrow 0$. The barrier Hessian is given by

$$H_b(\mu) = H - \sum_{i=1}^m \frac{\mu}{c_i} H_i + \mu J^T C^{-2} J.$$

As $x(\mu)$ converges to x^* , the first two terms of the Hessian approach the Hessian of the Lagrangian function at x^* ; the third matrix is given by

$$\mu J^T C^{-2} J = J^T D^2 J, \quad \text{where} \quad D^2 = \mu C^{-2} = \frac{1}{\mu} \Lambda^2. \quad (5.5.11)$$

We have already shown that the elements of D corresponding to inactive constraints are converging to zero, and the elements corresponding to active constraints are becoming unbounded.

Let m_a denote the number of active constraints whose gradients are linearly independent. The Hessian of the barrier function can be characterized in three ways, depending on m_a .

If no constraints are active at x^* , then the Hessian of the barrier function converges (as we would expect) to the Hessian of f itself.

At the other extreme, suppose that $m_a = n$, so that the Jacobian of the active constraints has rank n . If $\hat{\lambda}^* > 0$, the barrier Hessian along the trajectory approaches a large multiple of the *nonsingular* matrix $J_a^T (\hat{\Lambda}^*)^2 J_a$. In this case, the condition of the limiting Hessian depends on the condition of J_a and the condition of $\hat{\Lambda}^*$, but is not necessarily large. (This situation holds for linear programs in which there are no zero Lagrange multipliers.)

Finally, assume that $0 < m_a < n$. The limiting matrix D^2 then contains m_a unbounded elements and $n - m_a$ zero elements, which means that asymptotically $J^T D^2 J$ of (5.5.11) becomes not only unbounded, but also *rank-deficient*. Murray (1971) showed that $H_b(\mu)$ has m_a unbounded eigenvalues, corresponding to eigenvectors in the range space of J_a^T , and $n - m_a$ bounded eigenvalues, corresponding to eigenvectors in the null space of J_a . The barrier Hessian accordingly becomes increasingly ill-conditioned for “small” μ , and is *singular* in the limit. This property is one of the reasons that barrier function methods fell into disfavor in the 1970s, since standard unconstrained methods (such as Newton-based or quasi-Newton methods) tend to experience numerical difficulties when the Hessian is ill-conditioned. Various linear algebraic approaches have been proposed for dealing with this inherent ill-conditioning of the Hessian.