Title

I. UNIFIED APPROACH TO UNCONSTRAINED MINIMIZATION II. GENERATION OF CONJUGATE DIRECTIONS FOR UNCONSTRAINED MINIMIZATION WITHOUT DERIVATIVES

Author

Nazareth, Lawrence.

Publication Date

1973-11-01

I. UNIFIED APPROACH TO UNCONSTRAINED MINIMIZATION
II. GENERATION OF CONJUGATE DIRECTIONS FOR UNCONSTRAINED MINIMIZATION WITHOUT DERIVATIVES

Lawrence Nazareth
(Ph.D. thesis)

November 1973

Prepared for the U. S. Atomic Energy Commission
under Contract W-7405-ENG-48

DISCLAIMER

I. UNIFIED APPROACH TO UNCONSTRAINED MINIMIZATION

II. GENERATION OF CONJUGATE DIRECTIONS FOR UNCONSTRAINED MINIMIZATION WITHOUT DERIVATIVES

Lawrence Nazareth

November 1973

This research was supported in part by the Office of Naval Research Contract N00014-69-A-0200-1017 and in part by the Lawrence Berkeley Laboratory under the auspices of the U.S. Atomic Energy Commission.

I. UNIFIED APPROACH TO UNCONSTRAINED MINIMIZATION

II. GENERATION OF CONJUGATE DIRECTIONS FOR UNCONSTRAINED MINIMIZATION WITHOUT DERIVATIVES

Lawrence Nazareth

Abstract

Several important classes of algorithms for unconstrained minimization, when applied to a quadratic function with Hessian A, may be regarded as being alternative ways to effect certain matrix factorizations of or with respect to A. This approach leads to a clear insight into the basic equivalence of many algorithms that are implemented in very different ways and which differ in their informational requirements. It also enables their presentation within a unified framework.

For the case when the Hessian is directly available, we discuss in detail an algorithm, termed Algorithm TC, which effects a particular decomposition of the Hessian. This consists of partially solving the eigenproblem by tridiagonalizing the Hessian using orthogonal similarity transformations, and then obtaining the Cholesky factorization of the tridiagonal matrix. We call this the TC factorization. Successive steps of the two decompositions involved may be interleaved. Expressions for successive approximations to the inverse Hessian are developed. We show also that successive approximations to the inverse Hessian are related through certain recurrence relations.

One of our main reasons for developing this algorithm in detail is to demonstrate that important classes of algorithms

that employ only function values and first derivatives (these include the Conjugate Gradient Algorithm, a large family of Variable Metric Algorithms and certain other Quasi-Newton Methods) effect, implicitly, the same process when applied to a quadratic, but each in a different way. Our main theorem identifies a set of conditions under which different algorithms give identical iterates to the minimum of a quadratic. We also demonstrate that expressions for successive approximations to the inverse Hessian developed in Algorithm TC correspond exactly to expressions for successive approximations to the inverse Hessian given by a family of Variable Metric Methods. Further this family is related to Huang's family of algorithms. This relationship is brought out during the development of recurrence relations for successive approximations to the inverse Hessian in Algorithm TC. Equivalent implementations exist for algorithms that do not require derivatives and we also show equivalences between certain transformed versions of the above algorithms.

We discuss another important class of algorithms which we show correspond to alternative ways of effecting the QR (or Gram-Schmidt) factorization. The viewpoint adopted suggests some new implementations.

In Part II, the second half of this dissertation, we take up the development and analysis of a particular technique for generating conjugate search directions for unconstrained minimization without derivatives. This stems from two theorems proved by Powell. A particular version, which we consider in detail, is related to

the Jacobi eigenvalue process and the two processes, although different, help to illuminate one another. We study convergence of the search directions to mutual conjugacy and cases when cycling occurs. Our main result identifies a broad class of "cyclic patterns" for which convergence of the search directions to mutual conjugacy can be proven. This proof, suitably modified, carries over to the Jacobi process.

Acknowledgements

I am deeply grateful to Dr. B.N. Parlett for his guidance, advice and encouragement. I have benefited very much from his many useful suggestions and comments and thank him especially for bringing the topic in Part II to my attention. My thanks go also to the other two members of the committee, Dr. C.R. Glassey and Dr. I. Adler.

I thank Ruth Suzuki for typing this thesis very expertly.

This thesis is dedicated to Abigail Reeder.

Table of Contents

Chapter 1

1.1 Introduction

Algorithms for minimizing an unconstrained function usually work within the following general framework. An initial point is chosen from which a search is conducted along a suitably chosen direction or set of directions. From the set of all points at which the function is evaluated during such a search, the point where the function value is least is taken to be the new estimate of the minimum. We shall call this the current iterate and denote it by x^c. Fresh search directions are generated or the previous directions revised and the search procedure is repeated. Thus a sequence of successive approximations to the minimum is obtained.

The way in which the search directions are updated and the information used in carrying out the revision, differ from algorithm to algorithm. Most successful algorithms use directions which are a function of, or satisfy certain properties with respect to, one or more of the following:

- values of the function $\phi(x^c)$, at the current iterate x^c, and one or more previous iterates.
- the value and lengths (in some norm, usually the Euclidean) of the gradient vectors $\nabla\phi(x^c)$ at the current iterate x^c, and one or more previous iterates.
- the Hessian $A(x^c)$, i.e., the matrix of second partial derivatives, at the current iterate x^c.
- the distances moved along each search direction during one or more previous iterations of the algorithm.

The gradient $\nabla\phi(x^c)$ and the Hessian $A(x^c)$ are given by the second and third terms of a Taylor series expansion of $\phi(x)$ about x^c, i.e.,

$$\phi(x) = \phi(x^c) + \phi(x^c)^T(x-x^c) + \frac{1}{2}(x-x^c)^T A(x^c)(x-x^c) + \text{higher order terms} .$$

Usually an algorithm is designed to work well when applied to a quadratic function $\psi(x)$, given by

$$\psi(x) = a + b^T x + \frac{1}{2}x^T A x \qquad (1.1a)$$

where

a is a fixed constant

b is a fixed vector

and A is a constant symmetric positive definite matrix .

The algorithm is then stated in terms which allow its application to general functions, satisfying certain conditions, e.g. see Kowalik and Osborne [1].

1.2 Overview of the Thesis

This thesis is presented in two parts.

1.2.1. In Part I we explore the principal approaches to solving the unconstrained minimization problem.

In the first chapter of Part I, certain fundamental relations that underlie most of the algorithms of the field are presented in the form in which we shall use them later.

Next, in Chapter 3, we review briefly some standard matrix

decompositions of Computational Linear Algebra which we shall require and discuss how these correspond to methods for solving systems of equations and the symmetric algebraic eigenproblem.

By identifying which of the fundamental relations of Chapter 2 are employed in a particular unconstrained minimization algorithm and by manipulating these relations, we are then able to present, within a unified framework, several important classes of methods for finding the minimum of an unconstrained function and simultaneously to establish the connection with matrix decompositions and techniques of Computational Linear Algebra. The principal aspects of this chapter (Chapter 4) are as follows. For the case when the Hessian is directly available we discuss, in detail, an algorithm which effects a particular decomposition of the Hessian -- termed the TC factorization. This consists of partially solving the eigenvalue/vector problem by tridiagonalizing the Hessian using orthogonal similarity transformations and then obtaining the Cholesky factorization of the tridiagonal matrix. Successive steps of these two decompositions may be interleaved. Expressions for successive approximations to the inverse Hessian are developed. We show also, that successive approximations to the inverse Hessian are related through certain recurrence relations.

Our purpose in introducing such a scheme, termed Algorithm TC, is not primarily to put forward yet another contender into the field (for the case when second derivatives are known). Our principal reason for developing this algorithm in detail is to demonstrate that important classes of algorithms that employ only function values and first derivatives (these include the Conjugate

Gradient algorithm, a large family of Variable Metric algorithms and certain other Quasi-Newton Methods) effect, implicitly, the same process when applied to a quadratic, but each in a different way. This correspondence arises very naturally from a correspondence in the relationships underlying each algorithm and a uniqueness result associated with the tridiagonalization of a matrix. Our main theorem identifies a set of conditions under which different algorithms give identical iterates to the minimum of a quadratic. We also demonstrate that expressions for successive approximations to the inverse Hessian developed in Algorithm TC correspond exactly to expressions for successive approximations to the inverse Hessian given by a family of Variable Metric Methods. Further that this family is related to that of Huang [2], which we show may be developed from the recurrence relations obtained during the discussion on Algorithm TC. We also show equivalences between certain transformed versions of these algorithms.

We discuss in some detail another important class of algorithms which we show correspond to alternative ways of effecting the QR factorization. The viewpoint adopted suggests some new implementations.

We feel that the approach taken in Part I thus leads to several new insights into the basic equivalence of many algorithms that are implemented in very different ways and which differ in their informational requirements. Our approach should help clarify understanding of the numerical behaviour of such algorithms and, in addition, it suggests other possible algorithms for unconstrained optimization.

1.2.2. In Part II of this thesis we take up the analysis of a particular algorithm for unconstrained minimization without use of derivatives. This algorithm is of the conjugate direction type. We recall for the reader, that n search directions $(d_1,\ldots,d_n)$ are said to be conjugate with respect to a positive definite symmetric matrix A if and only if

$$d_i A d_j = 0 \qquad \text{for all} \quad i \neq j \quad .$$

The use of such directions in an unconstrained minimization algorithm stems from the following desirable property. For the quadratic $\psi(x)$ (1.1a), the minimum value will be reached by minimizing in sequence along n search directions that are mutually conjugate with respect to the Hessian A of $\psi(x)$.

Powell's Method [3] seems to be the most widely used algorithm for unconstrained minimization without derivatives. A set of n search directions is maintained. The algorithm carries out a series of minimal line searches used to revise the search directions; if the line searches are exact the algorithm, applied to a quadratic $\psi(x)$, will terminate with a set of conjugate directions, in a bounded number of steps. Since the minimum may thus be found by minimizing successively along each of these directions, we see that Powell's algorithm has what is commonly called a quadratic termination property. However the method has two disadvantages. The first is that there is a danger of linear dependence in the search directions even for a quadratic function, and precautions taken to ensure that the directions span the

complete space of variables often adversely affect the efficiency of the algorithm. The second disadvantage is that generation of conjugate directions is, in general, tied to accuracy of the line searches.

Brodlie's Method [4] gets around these difficulties. The search directions used are always orthogonal and there is therefore no danger of linear dependency. These directions are updated as the algorithm progresses so that they converge for a quadratic to a mutually conjugate set of directions, and this updating process is not dependent upon accuracy of line searches. Clearly the directions in Brodlie's algorithm are converging to the set of eigenvectors of A. He proves that the algorithm, applied to the quadratic $\psi(x)$, exactly parallels a cyclic Jacobi eigenvalue/vector process in the sense that successive approximations to the set of eigenvectors are identical.

Unlike Powell's Method, Brodlie's algorithm does not possess a quadratic termination property, though numerical evidence has shown it to be satisfactory in practice. Seeking the set of eigenvectors of A as Brodlie's algorithm does, would also seem unnecessarily costly. The directions they define are unique when the eigenvalues of A are distinct. All we really want are directions mutually conjugate with respect to A, and such directions are not unique.

In Part II we discuss an algorithm that relaxes the requirement that an essentially unique set of directions be generated, but shares the advantages of Brodlie's Method. It is based upon a technique for generating conjugate directions that stems from two

theorems proved by Powell [20]. These we shall state precisely later, but they may be summarized as follows.

Suppose we are dealing with the quadratic $\psi(x)$, whose Hessian is the positive definite symmetric matrix A. Powell's first theorem states that n linearly independent directions given by the columns of a matrix D, each of unit length in the A-norm, are mutually conjugate with respect to A if and only if the absolute value of the determinant of D (denoted by ΔD) is a maximum. This maximum may easily be shown to be $(\Delta A)^{-1/2}$.

The second theorem states that if D is postmultiplied by an orthogonal matrix Ω, and each column of $D\Omega$ renormalized to be of unit length in the A-norm giving a new matrix D^*, then $\Delta D^* \geq \Delta D$.

The following algorithm is suggested by these two theorems: A set of n search directions is maintained. At any iteration a search may be conducted in sequence along each of these directions and the current estimate of the minimum improved in some way. After normalizing each direction from information gathered during the search (as we shall see in Chapter 5) the set of search directions may be improved by postmultiplying by a suitably chosen orthogonal transformation. This completes an iteration of the algorithm.

Apart from some estimates of second derivatives and these only along the directions of search so as to perform the normalization, the use of explicit derivatives is avoided. An algorithm developed along these lines has the advantages of Brodlie's Method. Thus linear independence of the search directions is preserved. Convergence of the search directions to mutual conjugacy is not linked to

accuracy of the line searches (but, as we shall discuss shortly, is dependent upon using appropriate orthogonal transformations). This algorithm shares with Brodlie's Method the disadvantage of not having a quadratic termination property, but it does not insist upon convergence to an essentially unique set of directions.

Let us denote the normalized search directions at the k^{th} iteration by the columns of $D^{(k)}$ and the corresponding orthogonal transformation by $\Omega^{(k)}$. Our interest is in the choice of $\Omega^{(k)}$ and in the convergence of the columns of $D^{(k)}$ to mutual conjugacy for a quadratic. These are central issues. An algorithm developed along the above lines when applied to non-quadratic functions, must be capable of fast convergence in the neighborhood of a minimum, where the function is well-approximated by a quadratic. Thus we would like the columns of $D^{(k)}$ to converge to mutual conjugacy. This depends upon the choice of $\Omega^{(k)}$. Unless these orthogonal transformations are chosen sensibly it is clear that convergence need not occur. For example, postmultiplying $D^{(k)}$ by a permutation matrix merely interchanges the search directions and $\Delta D^{(k)}$ does not increase. Furthermore although Powell's second theorem implies that $\Delta D^{(k)}$ $(k = 1,2,...)$ is a bounded monotonically non-decreasing sequence, which must therefore necessarily tend to a limit, it does not follow that the limit equals $(\Delta A)^{-1/2}$. A perverse choice of $\Omega^{(k)}$ could cause $D^{(k)}$ to tend to a set of directions that are never arbitrarily close to mutual conjugacy. The off-diagonal elements of $[D^{(k)}]^T A[D^{(k)}]$ which are a measure of how close to mutual conjugacy the directions $D^{(k)}$ are, could possibly cycle. Even if convergence to mutual

conjugacy occurs this need not, as we shall see, imply convergence to fixed directions.

When $\Omega^{(k)}$ is always selected from the set of plane rotation matrices we discuss, in detail, the questions raised in the preceding paragraph. We show that the above difficulties are related to, but distinct from, difficulties encountered in the Jacobi eigenvalue process, and we compare and contrast the two processes at several points. By suitably modifying some of the convergence proofs, we have been also able to use the underlying ideas to prove the convergence of the cyclic Jacobi process for a much larger class of cyclic patterns than has currently appeared in the literature [5]. A detailed overview of the topics discussed in Part II is given in 5.6.

Part I

A Unified Approach to Unconstrained Minimization

Chapter 2

In this Chapter we present some simple properties of quadratics and certain fundamental relations that underlie most algorithms for minimizing an unconstrained function. We discuss some of the consequences of these basic relations and how they suggest algorithms for unconstrained minimization that are in current use.

This enables us to establish the notation and lay the groundwork, for the main results in Chapter 4.

A. Terminology

2.1 Definition of an n-search. Suppose that for a general function $\phi(x)$, a search along each of a set of n linearly independent directions $(d_1,\ldots,d_n)$ is carried out. Let x_1 be the initial point. Suppose that successive points $x_2,x_3,\ldots,x_{n+1}$ are generated with

$$x_{i+1} = x_i + \lambda_i d_i \ , \quad \lambda_i \neq 0 \ \text{ for all } \ i \ . \qquad (2.1a)$$

We shall call such a search procedure an n-search.

If, in addition, the search along each direction d_i seeks the minimum value of the function in that direction, we call the above search procedure a minimal n-search.

Denote the gradient of $\phi(x)$ at x_i by $g_i = g(x_i) = \nabla\phi(x_i)$.

If after conducting an n-search, g_{n+1} is found to be zero, we say the n-search is terminal.

B. Properties of Quadratics

2.2.1. A Fundamental Relation. Consider the quadratic function given by

$$\psi(x) = a + b^T x + \frac{1}{2} x^T A x$$

where

$$\nabla\psi(x) = g(x) = b + Ax \quad .$$

If we conduct an n-search as defined in 2.1, we have, for $i = 1,\ldots,n$,

$$\begin{aligned} g_i &= b + Ax_i \\ g_{i+1} - g_i &= A(x_{i+1} - x_i) \\ &= \lambda_i A d_i \qquad \text{by (2.1a).} \end{aligned} \tag{2.2a}$$

Writing

$$D = (d_1,\ldots,d_n)$$

and

$$Y = (g_2-g_1, g_3-g_2, \ldots, g_{n+1}-g_n)$$

we may then write (2.2a) as

$$AD = Y\lambda^{-1} \tag{2.2b}$$

where $\lambda^{-1} = \text{diag}[\lambda_1^{-1},\ldots,\lambda_n^{-1}]$. We make the convention henceforth that small unsubscripted greek letters denote diagonal matrices. If the n-search is terminal (i.e. if $g_{n+1} = 0$), this may be written as

$$AD = GH \tag{2.2c}$$

where

$$G = (g_1, \ldots, g_n)$$

and

$$H = \begin{pmatrix} -1 & & & & \\ 1 & -1 & & & \\ & 1 & -1 & & \\ & & 1 & \ddots & \\ & & & \ddots & -1 \\ & & & & 1 \end{pmatrix} \lambda^{-1} \quad ,$$

i.e. all elements h_{ij} of the square matrix H, such that $i < j$ or $(i-j) \geq 2$, are zero.

2.2.2. Minimum Value. The minimum value of $\psi(x)$ is attained when $\nabla\psi(x) = 0$, i.e. when

$$x = -A^{-1}b \quad . \tag{2.2d}$$

2.2.3. Invariance of a Quadratic Under Linear Transformation.

Suppose $\bar{Q}$ is a positive definite symmetric matrix (abbreviated henceforth as pds). The quadratic

$$\bar{\psi}(y) = a + (\bar{Q}b)^T y + \tfrac{1}{2} y^T \bar{Q} A \bar{Q} y$$

is obtained from $\psi(x)$ by the following change of variables: $y = \bar{Q}^{-1}x$.

The minimum of $\bar{\psi}(x)$ is attained at $\bar{Q}^{-1}z$, where z is the point where $\psi(x)$ attains its minimum value.

Also we write

$$\bar{d} = \bar{Q}^{-1}d$$

and

$$\begin{aligned} \bar{g} = \nabla\bar{\psi}(y) &= \bar{Q}(b + A\bar{Q}y) \\ &= \bar{Q}(b + Ax) \\ &= \bar{Q}g \quad \text{where} \quad g = \nabla\psi(x) \ . \end{aligned}$$

We employ this notation in Chapter 4.

C. Three Basic Relations

Fundamental to minimization algorithms are the notions of a set of directions being mutually conjugate, of a line search along direction d_i being minimal, and of the directions d_i used in an n-search being linear combinations of the gradients at x_i and at all previous iterates $x_1, x_2, \ldots, x_{i-1}$.

2.3 Conjugate Directions

2.3.1. The non-zero directions $D = (d_1, \ldots, d_n)$ are said to be conjugate with respect to the positive definite, symmetric (pds) matrix A, if and only if

$$d_i^T A d_j = 0 \quad \text{whenever} \quad i \neq j \ .$$

Thus any directions orthogonal in the norm defined by A are conjugate directions. We may write the above relation as

$$\text{(CD)} \qquad D^T A D = \text{diag}[\alpha_i] = \alpha \qquad (2.3a)$$

where α is a non-singular diagonal matrix with all positive elements. Recall the convention from 2.2.1 that small unsubscripted greek letters represent diagonal matrices.

Examples of conjugate directions with respect to the pds matrix A are:

a) If $A = R^T R$ is the Cholesky Decomposition of A, then

$$(R^{-1})^T A (R^{-1}) = I \ .$$

Thus, (R^{-1}) defines a set of mutually conjugate directions.

b) The matrix of column eigenvectors X of A satisfies

$$X^T A X = \text{diag}[\mu_i] = \mu$$

where μ_i $(i = 1,\ldots,n)$ are the eigenvalues of A. Hence the eigenvectors are a set of mutually conjugate directions. Since these eigenvectors may also be chosen so as to satisfy $X^T X = I$ they are also conjugate with respect to the identity matrix I (i.e., orthogonal).

2.3.2. If $D^T A D = \alpha$, then

$$A^{-1} = D\alpha^{-1}D^T = \sum_i (\frac{1}{\alpha_i}) d_i d_i^T \quad . \qquad (2.3b)$$

2.4 Minimal Line Search Relation

If we minimize a general function $\phi(x)$ along direction d_{i-1} and the minimum is attained at x_i, then

$$\text{(MLS)} \qquad g_i^T d_{i-1} = 0 \qquad (2.4a)$$

where

$$g_i = \nabla\phi(x_i) \quad .$$

We call (2.4a) the Minimal Line Search Relation, or the MLS Relation, for short.

2.5 The Direction-Gradient Relation

2.5.1. Suppose each search direction d_i, employed in an n-search, is defined in terms which implicitly or explicitly make it a

linear combination of the gradients at x_i and at all previous points $x_1,\ldots,x_{i-1}$. Then we have

$$D = GU$$

where

$$G = (g_1,\ldots,g_n) \quad \text{(recall 2.1)}$$

and

U is an upper triangular matrix.

We prefer to employ this in the form

$$\text{(DGR)} \qquad G = DR \qquad (2.5a)$$

where $R = U^{-1}$. U is invertible since the search directions d_i are linearly independent, and R is clearly also upper triangular. We call (2.5a) the Direction-Gradient Relation. In the form (2.5a) it may be interpreted as follows: each direction d_i is a linear combination of g_i and all previous directions $d_1,\ldots,d_{i-1}$.

2.5.2. In some cases which we shall consider later, we shall have directions d_i that are implicitly or explicitly linear combinations of $Qg_1,Qg_2,\ldots,Qg_i$, where Q is a pds matrix. We then have

$$QG = DR \quad . \qquad (2.5b)$$

D. Consequences of the Above Relations

Let us now consider the implications of combining two or more of the three basic relations discussed above.

A well known result when the Conjugate Direction relation and the MLS relation are combined is the following.

2.6 Quadratic Termination Property

Suppose that a minimal n-search (2.1) is conducted, for the quadratic function $\psi(x) = a + b^T x + \frac{1}{2}x^T A x$, along directions $d_1,\ldots,d_n$ that are mutually conjugate with respect to A. Then, it is well known that MLS (2.4a) may be strengthened to

$$g_i^T d_j = 0$$

for all i, j such that $1 \leq j < i \leq n+1$

and so

$$g_{n+1} = 0 \quad ,$$

i.e. the minimal n-search is also terminal.

It will be more useful to consider this in the form

$$G^T D = V \qquad (2.6a)$$

where

$$G = (g_1, g_2, \ldots, g_n)$$

and

V is an upper triangular matrix.

This result is well known; see, for example, Kowalik & Osborne [1]. It is used to prove the finite termination of several algorithms, when applied to a quadratic.

2.7 Orthogonality

The conditions that ensure the Quadratic Termination Property, which we have just discussed in 2.6, and the Direction-Gradient

relation, together imply that the columns of G are orthogonal. This follows from:

Lemma 2.1. If a minimal n-search is conducted along directions that are mutually conjugate, and the directions $D = (d_1,\ldots,d_n)$ and gradients $G = (g_1,\ldots,g_n)$ satisfy $G = DR$, then G^TG is a diagonal matrix, i.e. the columns of G are orthogonal.

Proof. From 2.6 the conditions of the Lemma imply that

$$G^TD = V \quad . \tag{2.7a}$$

Since $G = DR$ we have that

$$G^TG = VR \quad .$$

But VR is an upper triangular matrix and G^TG is symmetric. Hence VR is a diagonal matrix, and therefore the columns of G are orthogonal. □

Corollary. If instead the condition $QG = DR$ (cf. (2.5b)) replaces the condition $G = DR$ in Lemma 2.1, where Q is pds, then G^TQG is a diagonal matrix. Thus the columns of G are now orthogonal in the metric defined by Q.

E. Two Further Basic Relations

The idea of developing successive approximations to the inverse Hessian underlies large classes of algorithms known as Variable Metric or Quasi-Newton Methods. In the next two sections we discuss this and develop the fundamental relations that are involved.

2.8 The Variable Metric Relation

2.8.1. Suppose an n-search is conducted over a quadratic function $\psi(x)$, and linearly independent directions $(d_1,\ldots,d_n)$ are employed. Initially this is the only assumption we make about these directions. Suppose further that at the i^{th} step of the n-search, a non-singular matrix H_i is maintained such that

$$H_1 = Q$$

$$\text{(VMR)} \qquad H_i A D_i = D_i \quad (2 \leq i \leq n+1) \qquad (2.8a)$$

where D_i is the matrix $(d_1,\ldots,d_{i-1})$ and Q is some pds matrix.

Since D_{n+1} is a non-singular $n \times n$ matrix, we clearly have that $H_{n+1} = A^{-1}$ and the minimum value of $\psi(x)$ is then given by $-H_{n+1}g_{n+1}$.

(Note that the n-search is <u>not</u>, in general, terminal, i.e. g_{n+1} need not be zero.) We may thus regard H_i, $i = 1,2,\ldots,n+1$ as a series of successive approximations to the inverse Hessian, with H_1 being some initial approximation Q and $H_{n+1} = A^{-1}$. The Hessian A in (2.8a) may not be available explicitly. Since for a quadratic, $A(x_{i+1}-x_i) = g_{i+1} - g_i$, we transform (2.8a) into an expression that does not explicitly involve A as follows:

$$H_1 = Q$$

$$\text{(VMR)} \qquad H_i Y_i = S_i \quad (2 \leq i \leq n+1) \qquad (2.8b)$$

where

$$Y_i = (g_2-g_1,\ldots,g_i-g_{i-1})$$

and

$$S_i = (x_2-x_1,\ldots,x_i-x_{i-1}) \quad .$$

In this form only gradients at successive iterates are used. Using the theory of generalized inverses, see Adachi [6], it is possible to obtain a general expression for H_i from its definition in the form (2.8b). Furthermore it is possible to obtain a recursive expression giving H_{i+1} in terms of quantities derived from H_i, S_i and Y_i. All known methods that use the VMR (2.8a) employ special cases of these expressions.

2.8.2. In the previous section (2.8.1) we have not specified how $d_1,\ldots,d_n$ are obtained, demanding only that they be linearly independent. One way of defining the directions is as follows

$$d_i = -H_i g_i \quad (1 \leq i \leq n) \; . \qquad (2.8c)$$

Thus each search direction looks like a Newton step, but with the approximation H_i to the inverse Hessian replacing the exact inverse A^{-1}.

Several algorithms develop the inverse Hessian using (2.8a or b) and (2.8c) and are able to minimize a quadratic in a finite number of steps, at most $(n+1)$. Note that the use of (2.8c) is not essential for finite termination. Numerical difficulties often plague such algorithms, but their outstanding feature is that a minimal line search along each d_i is not required for finite termination. Also, for some i it is possible that $d_i = 0$, i.e. it is possible for breakdown of the process to occur.

2.8.3. More can be said when we impose the further requirement

that the search along each direction be minimal, i.e. that the n-search of the preceding section be, in fact, a minimal n-search; we also assume that $d_i \neq 0$, for all i; then we can easily show that the n-search is terminal, i.e. $g_{n+1} = 0$. This follows from the next Lemma, a reformulation of a standard result.

Lemma 2.2. Consider a minimal n-search conducted for a quadratic $\psi(x)$ along the directions $(d_1,\ldots,d_n)$ defined by (2.8c) with H_i defined by (2.8a) or equivalently (2.8b). Then the directions $(d_1,\ldots,d_n)$ are mutually conjugate.

Proof. From (2.8a) we have

$$H_i A D_i = D_i \quad .$$

Thus

$$g_i^T H_i A D_i = g_i^T D_i \quad .$$

Then, from (2.8c)

$$-d_i^T A D_i = g_i^T D_i \quad . \tag{2.8d}$$

The proof is by induction. Assume that the first $i-1$ directions $(d_1,\ldots,d_{i-1})$, i.e. the columns of D_i, are mutually conjugate. Since the searches are minimal, it then follows from the Quadratic Termination Property (2.6) applied to the set of directions $(d_i,\ldots,d_{i-1})$ that $g_i^T d_j = 0$ for all $j < i$. Thus from (2.8d)

$$d_i^T A D_i = 0 \quad .$$

Therefore $(d_1,\ldots,d_i)$ are mutually conjugate, i.e. the induction hypothesis holds for D_{i+1}. The result then follows by induction,

since $g_2^T d_1 = 0$, implying that d_1 and d_2 are mutually conjugate. □

That the minimal n-search using directions, obtained from (2.8a or b) and (2.8c) is terminal, i.e. $g_{n+1} = 0$, then follows from the Quadratic Termination Property, since we have just shown that these directions are conjugate. (Note that for termination of this n-search we need both conditions (2.8c) and minimal line searches, in marked contrast to (2.8.1) and (2.8.2)). We shall call (2.8a), or its alternative form (2.8b), the Variable Metric Relation (VMR).

2.9 The Quasi-Newton Relation

Instead of defining H_i by the Variable Metric Relation (2.8a) suppose we use instead only the relation obtained by equating the last column of the left hand side and right hand side of (2.8a), i.e.

$$H_1 = Q$$

$$\text{(QNR)} \qquad H_i A d_{i-1} = d_{i-1} \quad (2 \leq i \leq n+1) \qquad (2.9a)$$

or alternatively

$$\text{(QNR)} \qquad H_i y_{i-1} = s_{i-1} \quad (2 \leq i \leq n+1) \qquad (2.9b)$$

where $y_{i-1} = (g_i - g_{i-1})$ and $s_{i-1} = (x_i - x_{i-1})$. (2.9a or b) is known as the Quasi-Newton Relation (QNR). Again a general expression and a recursive relation may be obtained for H_i which includes, as a subclass, the matrices defined by the Variable Metric Relation. Now, if we conduct a minimal n-search using directions defined by

(2.8c) and the Quasi-Newton Relation it will not necessarily follow that the directions are mutually conjugate. Furthermore, even if H_i, in the Quasi-Newton Relation, is chosen so that the directions d_i defined by (2.8c) are mutually conjugate, it does not appear to be true that H_i must then satisfy the Variable Metric Relation.

Chapter 3

Some Useful Matrix Decompositions

In this Chapter we review briefly some standard matrix decompositions, and how these lead to certain algorithms for solving systems of equations and the symmetric algebraic eigenvalue/vector problem. We shall require these in the next Chapter and they are collected here for easy reference.

A. Decomposition Related to the Solution of Sets of Linear Equations

3.1 Triangular Factorization

The fundamental fact behind direct methods for solving $Ax = b$ is that under certain conditions (nonsingular principal submatrices) A can be uniquely factored as

$$A = \hat{L}\hat{D}\hat{U}$$

where

$\hat{L}$ is unit lower triangular

$\hat{D}$ is diagonal

$\hat{U}$ is unit upper triangular.

The usual Gaussian Elimination algorithm implicitly yields

$$A = LU$$

where $L = \hat{L}$ and $U = \hat{D}\hat{U}$. See Wilkinson [7].

When A is pds the necessary conditions are fulfilled and

$$A = LL^T$$

where $L = \hat{L}\hat{D}^{1/2}$. This is the Cholesky factorization.

3.2 QR Factorization

Any non-singular matrix G can be decomposed as follows

$$G = DR \tag{3.2a}$$

where

$$D^TAD = \alpha \quad .$$

R is a unit upper triangular matrix, A is positive definite symmetric, and $\alpha = \text{diag}[\alpha_i]$, using our convention that small unsubscripted greek letters denote diagonal matrices. Thus the columns of D are orthogonal in the metric defined by a pds matrix A. The decomposition is essentially unique, Wilkinson [7]. (This is better known as the QR factorization but we have changed the notation for later convenience).

(i) The above decomposition is implicitly what is obtained when carrying out the Gram-Schmidt Orthogonalization Process on the vectors defined by the columns of G. The standard relations of the Gram-Schmidt Process may be obtained by equating columns of the left hand side and right hand side of $G = DR$. If we denote the i^{th} column of G by g_i, i.e. $G = (g_1,\ldots,g_n)$ we have

$$\begin{aligned} d_1 &= g_1 \\ d_j &= g_j - \sum_{i=1}^{j-1} r_{ij} d_j \end{aligned} \tag{3.2b}$$

where

$$r_{ij} = d_i^TAg_j/(d_i^TAd_i)$$

being chosen to satisfy the relations $d_i^TAd_j = 0$ for $i \neq j$.

(ii) Alternative ways of obtaining the QR factorization of a matrix A avoid the numerical inaccuracies of standard Gram-Schmidt. They are:

a) Premultiplication by a series of plane rotations, Given's Method.

b) Premultiplication by a series of elementary Hermitian matrices, Householder's Method.

For details see Wilkinson [7], Chapter 4. We stress that these are alternative realizations of the same basic relations.

(iii) The Gram-Schmidt Process may be viewed as projecting at step i, the vector g_i into a space orthogonal to that spanned by $(Ad_1,\ldots,Ad_{i-1})$. Thus d_i is obtained by premultiplying g_i by an orthogonal projection matrix, i.e. a symmetric matrix that satisfies the relation $X^2 = X$. This orthogonal projection matrix may be written as

$$P_i = I - (AD_i)(AD_i)^+$$

where D_i is the $n \times (i-1)$ matrix $(d_1,\ldots,d_{i-1})$ and $(AD_i)^+$ is the generalized inverse of (AD_i), see Boullion & Odell [8]. Now, since (AD_i) is of full column rank

$$(AD_i)^+ = [(AD_i)^T(AD_i)]^{-1}(AD_i)^T \quad .$$

See, again, Boullion & Odell [8], page 11. Thus

$$P_i = I - (AD_i)[(AD_i)^T(AD_i)]^{-1}(AD_i)^T \quad . \qquad (3.2c)$$

In recursive form this yields

$$P_{i+1} = P_i - P_i A d_i d_i^T A P_i / (d_i^T A P_i A d_i) \quad . \qquad (3.2d)$$

The orthogonalizing process is thus given very simply by

$$d_1 = g_1$$

$$d_j = P_j g_j \quad \text{for all} \quad j \geq 2 \qquad (3.2e)$$

where P_j is updated using (3.2c) or (3.2d).

B. Decompositions Related to the Algebraic Eigenproblem

3.3 Any symmetric matrix A can be decomposed into

$$A = X \mu X^T$$

where

$$X^T X = I$$

and

$$\mu = \text{diag}[\mu_i] \text{ is a diagonal matrix .}$$

X consists of the set of eigenvectors, the i^{th} column being the eigenvector corresponding to the eigenvalue μ_i.

The Jacobi Method is one way of obtaining this decomposition. In it X is accumulated as a project of elementary rotations, the process being defined as

$$A^{(1)} = A$$

$$A^{(k+1)} = \Omega_k^T A^{(k)} \Omega_k$$

where the elementary rotation Ω_k is obtained as follows: Some rule is used to select a pair (p,q) of indices. Then

$$\Omega_k = (w_{st}^{(k)}) \qquad (3.3a.1)$$

such that

$$w_{pp}^{(k)} = w_{qq}^{(k)} = \cos\theta$$
$$w_{pq}^{(k)} = -w_{qp}^{(k)} = \sin\theta$$
$$w_{ss}^{(k)} = 1\,, \quad s \neq p \neq q$$
$$w_{st}^{(k)} = 0\,, \quad \text{otherwise.}$$

θ is chosen to satisfy

$$\tan 2\theta = \frac{2a_{pq}^{(k)}}{(a_{pp}^{(k)} - a_{qq}^{(k)})} \qquad (3.3a)$$

and $a_{ij}^{(k)}$ is the $(i,j)^{th}$ element of $A^{(k)}$.

If the rule is:

(i) choose (p,q) such that $a_{pq}^{(k)}$ is the largest off-diagonal element we obtain the Classical Jacobi Process. Convergence is assured.

(ii) choose (p,q) according to some cyclic pattern, we obtain the Cyclic Jacobi Process.

(iii) choose (p,q) by rows or by columns we obtain the Special Cyclic Jacobi Process. Convergence for this case has been proven, see Forsythe & Henrici [9].

For more details of the Jacobi Process see Wilkinson [7].

3.4 Reduction to Upper Hessenberg Form and Tridiagonal Form

3.4.1. Any matrix A can be decomposed as follows

$$AG = GH^u$$
$$G^TG = \beta \qquad (3.4a)$$

where H^u is unit upper Hessenberg and $\beta = \text{diag}[\beta_i]$. These may also be written

$$\begin{aligned} G^T A G &= \beta H^u \\ G^T G &= \beta \quad . \end{aligned} \tag{3.4b}$$

If A is symmetric then βH^u must be symmetric as well as Hessenberg. Hence tridiagonal, which we shall denote by T,

$$T = \beta H^u \tag{3.4c}$$

and

$$\begin{aligned} G^T A G &= T \\ G^T G &= \beta \end{aligned} \tag{3.4d}$$

(TF)

and if A is positive definite then so is T. We shall make extensive use of the tridiagonal factorization TF (3.4d) in Chapter 4.

In subsequent sections we shall also need the following uniqueness result:

Lemma 3.1. If $G_1^T A G_1 = T_1$ and $G_2^T A G_2 = T_2$ where G_1 and G_2 are orthogonal matrices which have the same first column, say g_1, and T_1 and T_2 are tridiagonal, then

$$G_2 = G_1 \delta \quad \text{and} \quad T_2 = \delta T_1 \delta$$

where $\delta = \text{diag}[\delta_i]$, with $\delta_i = \pm 1$ for all i.

Proof. The proof is given in Wilkinson [7], page 352. (The proof breaks down if any $t_{i+1,i} = 0$, in which case (in 3.4d) g_{i+1} can be chosen to be an arbitrary vector orthogonal to $g_1, \ldots, g_i$. For

our needs this will not present a difficulty, since when applying this result in Chapter 4 this case does not arise). We shall use the above uniqueness result to prove the equivalence of various methods of unconstrained minimization later in Part I. □

3.4.2. We shall also require the decomposition (3.4b) in the following form: reduce QAQ to unit upper Hessenberg form using transformations G whose columns are orthogonal in the Q-metric, where Q is some pds matrix

$$\begin{aligned} G^TQAQG &= \bar{\beta}\bar{H}^u \\ G^TQG &= \bar{\beta} \quad . \end{aligned} \tag{3.4e}$$

If A is symmetric then $\bar{T} = \bar{\beta}\bar{H}^u$ is tridiagonal (since $Q^T = Q$) and again G is unique up to column signs.

Writing $\bar{G} = \bar{Q}G$ where $Q = \bar{Q}^2$ this may be interpreted as reducing $\bar{Q}A\bar{Q}$ to unit upper Hessenberg (or tridiagonal form) using transformations $\bar{G}$ whose columns are orthogonal in the Euclidean metric.

3.5 Methods for obtaining G and H^u (or T) are as follows:

(i) Writing out (3.4a) as a set of recurrence relations, and assuming that the first column of G is specified to be g_1, we obtain

$$Ag_r = g_{r+1} + \sum_{i=1}^{r} h_{ir}g_i \tag{3.5a}$$

where h_{ir}, the $(i,r)^{th}$ element of H^u, is chosen so that (3.5a) satisfies

$$g_i^T g_{r+1} = 0 \quad \text{for all} \quad i \leq r \ , \qquad (3.5a.1)$$

i.e. $$h_{ir} = g_i^T A g_r / (g_i^T g_i) \ .$$

This is known as Arnoldi's Method. Given a symmetric matrix we can tridiagonalize it using relations (3.5a), since $t_{ir} = \beta_i h_{ir}$ from (3.4c).

We shall also need the recurrence relations for the decomposition given by (3.4e) and this is a convenient place to give them. These are obtained directly from (3.5a) by substituting $\bar{Q}A\bar{Q}$ for A and $\bar{Q}g_r$ for g_r, i.e.

$$AQg_r = g_{r+1} + \sum_{i=1}^{r} \bar{h}_{ir} g_i \qquad (3.5b)$$

where

$$\bar{h}_{ir} = g_i^T QAQg_r / (g_i^T Q g_i)$$

and if A is symmetric, $\bar{T}$ is given by $\bar{t}_{ir} = \bar{\beta}_i \bar{h}_{ir}$.

(ii) (3.4b) may also be written as

$$[G\beta^{-1/2}]^T A [G\beta^{-1/2}] = \beta^{-1/2} H^u \beta^{-1/2} \qquad (3.5c)$$
$$[G\beta^{-1/2}]^T [G\beta^{-1/2}] = I$$

using our convention that $\beta = \text{diag}[\beta_i]$. Then $G\beta^{-1/2}$ may be obtained as a product of plane rotations (Given's Method) or as a product of elementary Hermitian matrices (Householder's Method) by carrying out a series of elementary orthogonal similarity transformations on A, that finally reduce it to upper Hessenberg form, or tridiagonal form if A is symmetric. For details see Wilkinson [7], Chapter 6.

Chapter 4

A Unified Approach to Unconstrained Minimization

As noted in Chapter 1, the various algorithms of unconstrained minimization differ principally in the way the search directions are generated or updated. Many algorithms, when applied to quadratics, have the same underlying relations and, as we shall show, implicitly effect the same decomposition of or with respect to the Hessian. Such algorithms differ in the way the decomposition is carried out, in much the same way as the Givens, Householder or Gram-Schmidt methods all effect the QR factorization, but each in a different way. From a numerical standpoint, of course, these differences are significant but, in exact arithmetic, with certain initial conditions the same, they would all give identical results. The other principal factor that distinguishes one minimization algorithm from another is the informational requirements of an algorithm, e.g. some algorithms use no derivatives, others do.

Thus in our discussion, each algorithm that we consider here will be placed at a "point" determined by two sets of "co-ordinates" -- a) the information used and b) the underlying relations and matrix decomposition involved.

We refer the reader back to 1.2.1 for a detailed overview of this chapter.

A. Methods That Use Second Derivatives

We include in this section both methods that have available the Hessian $A(x^c)$ at the current iterate x^c, through access

to second partial derivatives, and methods that estimate the complete Hessian at x^c, for example, by estimating second partial derivatives in the directions of the coordinate axes e_i for all i and along (e_i+e_j) for all pairs $i < j$. From this the Hessian at x^c may be deduced.

After briefly reviewing some of the important methods in this area, we discuss in detail an algorithm that effects a particular decomposition of the Hessian, namely tridiagonalization followed by Cholesky decomposition. Prior to introducing this algorithm we state our motivation for presenting it.

4.1 Methods in this class are usually variants of Newton's Method, which employs the direction

$$d^c = -[A(x^c)]^{-1}g^c \qquad (4.1a)$$

where x^c is the current iterate and g^c the gradient at x^c. The direction d^c may be obtained by inverting $A(x^c)$ or by solving the set of equations

$$A(x^c)d^c = -g^c \quad .$$

This solution is obtained in two principal ways.

The first and most natural class of methods uses the triangular factorizations of Chapter 3. Thus Fiaccio and McCormick [10] factorize $A(x^c)$ into $\hat{L}\hat{D}\hat{L}^T$ and solve $(\hat{L}\hat{D}\hat{L}^T)d^c = -g^c$ (note that the columns of $\hat{L}^{-1}$ are conjugate directions). Matthew and Davies [11] factorize $A(x^c)$ into the LU factorization and solve. Since

for general functions $A(x^c)$ may be indefinite, precautions to prevent numerical instability are also built into these algorithms.

The second class of methods employs the Orthogonal Decompositions of Chapter 3, 3.3. The method of Greenstadt [12] carries out the decomposition

$$X^T A(x^c) X = \mu$$

$$X^T X = I$$

where

$$[A(x^c)]^{-1} = X\mu^{-1}X^T = \sum_i \frac{1}{\mu_i} x_i x_i^T \quad . \qquad (4.1b)$$

The Jacobi Method 3.3 may be used to obtain this. Alternatively this decomposition may be carried out by first reducing $A(x^c)$, which is always symmetric, to tridiagonal form T as in 3.4 and then obtain the eigenvalues and eigenvectors of T and hence those of $A(x^c)$. Again precautions are built in to prevent numerical instability when any μ_i is small, since it is entirely possible that the Hessian at a particular iterate x^c may be very ill-conditioned, although at the minimum itself, say z, $A(z)$ is well conditioned.

4.2 Algorithm TC. In this section we propose a method intermediate between the above two classes of methods. Our motivation for introducing it is twofold. It does not appear in the literature and this variant should, in most cases, be much faster than Greenstadt's algorithm. However our principal reason for developing and exploring this algorithm in detail is for reasons of exposition.

We claim that the decomposition involved, tridiagonalization followed by Cholesky factorization is, in some sense, fundamental. Many algorithms, as we shall show later (in particular the conjugate gradient and a class of variable metric methods implicitly effect the same decomposition of the Hessian, each in a different way. We refer the reader also to our overview in 1.2.1.

In describing this algorithm we shall, for simplicity, denote the Hessian at the current iterate by the symmetric matrix A and henceforth we drop the superfix c denoting the current ierate.

The Algorithm:

(i) Reduce A to tridiagonal form T, see (3.4d) of Chapter 3. We shall insist that the first column of G be g_1, the gradient at the current iterate, and use equations (3.5a) to complete the decomposition. This decomposition is essentially unique. At the $(j-1)^{th}$ stage of this reduction we shall therefore have obtained the first j columns of G, i.e., $(g_1,\ldots,g_j)$ and the first $(j-1)$ columns of T.

(ii) Next carry out the Cholesky factorization of the symmetric tridiagonal matrix T

$$T = R^T \alpha R$$

where R is unit upper triangular and all elements r_{ij} such that $(j-i) \geq 2$ are zero; $\alpha = \text{diag}[\alpha_i]$ is a diagonal matrix.

Now it is not necessary to perform step (ii) only after completing step (i). As the elements of T are generated in step

(i) we can also carry out successive stages of the Cholesky Decomposition. We enlarge on this in the next section 4.3.

(iii) If at any stage, the Cholesky factorization breaks down (which could happen if A is not positive definite) then we only carry (i) to completion omitting further stages of (ii). Then, once the tridiagonal form T has been found, the standard Greenstadt procedure [12] is followed by obtaining the eigenvalue and vectors of T. The search direction for a Newton step is then given by (4.1a) and (4.1b).

(iv) If, however, the Cholesky factorization is successful (and if A is positive definite it will be) then the decomposition effected is

$$G^TAG = R^T\alpha R \qquad (4.2a)$$
$$G^TG = \beta$$

which we call the TC factorization.

The inverse Hessian is given by

$$A^{-1} = G(R^{-1})\alpha^{-1}(R^{-1})^TG^T \quad . \qquad (4.2b)$$

The search direction $d = -A^{-1}g_1$ for a Newton step is thus

$$d = -G(R^{-1})\alpha^{-1}(R^{-1})^TG^Tg_1 \qquad (4.2c)$$
$$= -GR^{-1}\alpha^{-1}(R^{-1})^T(g_1^Tg_1)e_1$$

where e_1 is the first column of the unit matrix. This is equivalent to solving the equations

$$R^T \alpha R y = G^T g_1 = (g_1^T g_1) e_1$$
$$d = -Gy \quad . \tag{4.2d}$$

Clearly we could have a substantial saving over Greenstadt's algorithm which must compute the eigenvalues and vectors, at every stage.

Definition. For later reference we define $D = GR^{-1}$, from which it follows that $A^{-1} = D\alpha^{-1}D^T$.

4.3 Development of the Inverse

We discuss in detail the development of the inverse Hessian (4.2b) for Algorithm TC and in particular we develop expressions for successive approximations to A^{-1}. Our motivation for doing this is that we wish to demonstrate later the close tie-in between these expressions and expressions for successive approximations to A^{-1} obtained by Variable Metric Algorithms.

The development is in four stages: a) we obtain expressions for successive approximations to R in (4.2a), b) we deduce expressions for successive approximations to R^{-1}, c) we develop expressions for successive approximations H_i to A^{-1} and d) we develop recurrence relations for H_i.

4.3.1. Successive Approximations to R

As noted in step (ii) of Algorithm TC we need not wait until the tridiagonalization and Cholesky factorization are complete to develop R. Intermediate stages of R may be developed in parallel with the determination of the matrix T.

At stage $(j-1)$ of Algorithm TC we have $(g_1,\ldots,g_j)$, the first j columns of G, and $(t_1,\ldots,t_{j-1})$ the first $(j-1)$ columns of T. Since the tridiagonal matrix T is symmetric we therefore also know the value of the element $t_{(j-1),j}$ (in the j^{th} column of T).

We write out the elements of the j^{th} principal leading submatrix T_j of T as follows:

$$T_j = \begin{pmatrix} t_{11} & t_{12} & & & & & \\ t_{21} & t_{22} & t_{23} & & & & \\ & t_{32} & t_{33} & t_{34} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & t_{(j-1),(j-2)} & t_{(j-1),(j-1)} & t_{(j-1),j} & \\ & & & & t_{j,(j-1)} & \tau^{(j)} & \end{pmatrix}$$

where $\tau^{(j)}$ is an arbitrary unknown parameter and all the other elements are known. The Cholesky factorization of this is of the form

$$R_j^T \, \mathrm{diag}[\alpha_1,\ldots,\alpha_{j-1},\nu^{(j)}]R_j \qquad (4.3a)$$

where R_j is a unit upper triangular $(j \times j)$ matrix with all elements r_{ij}, $(j-i) \geq 2$, being zero, and $\nu^{(j)}$ is another arbitrary parameter. In the Cholesky factorization R_j is entirely dependent upon the known elements of T_j, i.e., $\tau^{(j)}$, the (n,n) element does not affect any of R_j's elements. The unknown parameter $\tau^{(j)}$ only affects the last element of the diagonal matrix

$\mathrm{diag}(\alpha_1,\ldots,\alpha_{j-1},\nu^{(j)})$. This is therefore given by another arbitrary non-zero parameter, denoted by $\nu^{(j)}$. Clearly R_j is the $(j \times j)$ leading principal submatrix of R.

At the next iteration we know $(g_1,\ldots,g_{j+1})$ and $(t_1,\ldots,t_j)$. Now we deduce the Cholesky factorization of T_{j+1} from that of T_j, where

$$T_{j+1} = \begin{pmatrix} t_{11} & t_{12} & & & & & \\ t_{21} & t_{22} & t_{23} & & & & \\ & t_{32} & t_{33} & t_{34} & & & \\ & & \cdot & \cdot & & & \\ & & & \cdot & \cdot & & \\ & & & & t_{j,(j-1)} & t_{jj} & t_{j,(j+1)} \\ & & & & & t_{(j+1),j} & \tau^{(j+1)} \end{pmatrix}$$

T_{j+1} must be factored into

$$T_{j+1} = R_{j+1}^T \mathrm{diag}[\alpha_1,\ldots,\alpha_{j-1},\alpha_j,\nu^{(j+1)}]R_{j+1} \quad .$$

Noting that R is tridiagonal and employing (4.3a) we may write this as

$$T_{j+1} = \left(\begin{array}{c|c|c} R_j^T & \underset{\sim}{0} & \\ \hline \underset{\sim}{0} & r_{j+1,j} & 1 \end{array}\right)\left(\begin{array}{c|c|c} \alpha_1 & & \\ & \ddots\ \alpha_{j-1} & \\ \hline & & \alpha_j \\ \hline & & \nu^{(j+1)} \end{array}\right)\left(\begin{array}{c|c|c} R_j & & \underset{\sim}{0} \\ \hline & & r_{j,j+1} \\ \hline \underset{\sim}{0} & & 1 \end{array}\right) \quad .$$

Equating the (j,j) element

$$r_{j,j-1}^2\alpha_{j-1} + \alpha_j = t_{jj}$$

yielding

$$\alpha_j = t_{jj} - \alpha_{j-1} r_{j,j-1}^2 \quad . \tag{4.3b}$$

Also

$$r_{j+1,j}\alpha_j = t_{j+1,j}$$

yielding

$$r_{j+1,j} = t_{j+1,j}/(t_{jj} - \alpha_{j-1} r_{j,j-1}^2) \quad . \tag{4.3c}$$

4.3.2. Successive Approximations to R^{-1}

Bcause R_{j+1} is unit upper triangular the inverse is given by

$$R_{j+1}^{-1} = \left(\begin{array}{c|c} R_j^{-1} & -R_j^{-1}\rho_j \\ \hline 0 & 1 \end{array}\right) \tag{4.3d}$$

where

$$\rho_j = (0,0,\ldots,0,r_{j,j+1})^T = (r_{j,j+1})e_j$$

and e_j is the j^{th} column of the identity matrix, I.

Thus R_{j+1}^{-1} may thus also be updated as the iteration progresses.

4.3.3. Successive Approximations to A^{-1}

Using the results of the previous section let us now develop an expression for successive approximations to the inverse Hessian A^{-1}. Later in Section 4.10 we show an exact correspondence between these expressions developed here and expressions for successive approximations to A^{-1} given by a class of variable metric algorithms.

At stage (j-1) of Algorithm TC we have partially achieved the decomposition (4.2a). The known elements of $R^T \alpha A$ are given by (4.3a). It is therefore quite natural to consider the following approximation to the inverse.

$$H_j = G \left(\begin{array}{c|c} R_j^{-1} \begin{bmatrix} \alpha_1^{-1} & & & \\ & \ddots & & \\ & & \alpha_{j-1}^{-1} & \\ & & & \delta^{(j)} \end{bmatrix} R_j^{T^{-1}} & \underset{\sim}{0} \\ \hline \underset{\sim}{0} & L_{j+1} \end{array} \right) G^T \qquad (4.3e)$$

where L_{j+1} is an arbitrary pds matrix of appropriate dimension and $\delta^{(j)}$, the inverse of $\nu^{(j)}$ ($\nu^{(j)} \neq 0$), is another arbitrary parameter. From (4.3d)

$$H_{j+1} = G \left(\begin{array}{c|c} \begin{bmatrix} R_j^{-1} & -R_j^{-1}\rho_j \\ \hline 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1^{-1} & & \\ & \ddots & \\ & & \alpha_j^{-1} \\ \hline & & \delta^{(j+1)} \end{bmatrix} \begin{bmatrix} R_j^{T^{-1}} & \underset{\sim}{0} \\ \hline -\rho_j^T R_j^{T^{-1}} & 1 \end{bmatrix} & \underset{\sim}{0} \\ \hline \underset{\sim}{0} & L_{j+2} \end{array} \right) G^T$$

In order to facilitate comparisons we partition G as $G = (G_j, F_{j+1})$, $G_j = (g_1, \ldots, g_j)$, $F_{j+1} = (g_{j+1}, \ldots, g_n)$. Then

$$H_j = G_j R_j^{-1} \mathrm{diag}[\alpha_1^{-1}, \ldots, \alpha_{j-1}^{-1}, \delta^{(j)}] R_j^{T^{-1}} G_j^T + F_{j+1} L_{j+1} F_{j+1}^T \qquad (4.3e.1)$$

and

$$H_{j+1} = (G_j | g_{j+1}) \left[\begin{array}{c|c} R_j^{-1} & -R_j^{-1}\rho_j \\ \hline \underset{\sim}{0} & 1 \end{array}\right] \left[\begin{array}{c|c} \begin{matrix} \alpha_1^{-1} & & \\ & \ddots & \\ & & \alpha_j^{-1} \end{matrix} & \underset{\sim}{0} \\ \hline \underset{\sim}{0} & \delta^{(j+1)} \end{array}\right] \left[\begin{array}{c|c} R_j^{T^{-1}} & 0 \\ \hline -\rho_j^T R_j^{T^{-1}} & 1 \end{array}\right] \left[\begin{array}{c} G_j^T \\ \hline g_{j+1}^T \end{array}\right]$$

$$+ F_{j+2} L_{j+2} F_{j+2}^T \quad . \tag{4.3f}$$

4.3.4. Recurrence Relations for H_j

We wish to develop a recurrence relation relating H_{j+1} and H_j. There are several ways of doing this. We discuss one of them. To this end let us require that, for all j, L_{j+1} be the diagonal matrix

$$L_{j+1} = \text{diag}(\gamma^{(j+1)},\ldots,\gamma^{(n)}) \tag{4.3f.1}$$

where $\gamma^{(i)}$ for all i is an arbitrary parameter and $\gamma^{(i)} > 0$. Then we obtain:

$$H_{j+1} - H_j = G_j R_j^{-1} \left[\begin{array}{c|c} \begin{matrix} 0 & & \\ & \ddots & \\ & & 0 \end{matrix} & \\ \hline & \alpha_j^{-1} - \delta^{(j)} \end{array}\right] R_j^{T^{-1}} G_j^T$$

$$- \delta^{(j+1)} [G_j R_j^{-1} p_j g_{j+1}^T + g_{j+1} p_j^T R_j^{T^{-1}} G_j^T]$$

$$+ (\delta^{(j+1)} - \gamma^{(j+1)}) g_{j+1} g_{j+1}^T \quad . \tag{4.3g}$$

The recurrence relations for $j = 1,2,\ldots,n$ have $2n$ arbitrary

parameters given by $\delta^{(1)},\ldots,\delta^{(n)}$ and $\gamma^{(1)},\ldots,\gamma^{(n)}$. The relations may be manipulated further by introducing H_j into the RHS as follows: From (4.3e.1) and (4.3f.1)

$$H_j g_{j+1} = \gamma^{(j+1)} g_{j+1} \quad .$$

Substituting into (4.3g) we obtain

$$H_{j+1} - H_j = G_j R_j^{-1} \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \alpha_j^{-1} - \delta^{(j)} \end{array}\right] R_j^{T^{-1}} G_j^T$$

$$+ q^{(j+1)} [H_j g_{j+1} g_{j+1}^T H_j] \qquad (4.3h)$$

$$- \frac{\delta^{(j+1)}}{\gamma^{(j+1)}} [G_j R_j^{-1} p_j g_{j+1}^T H_j + H_j g_{j+1} p_j^T R_j^{T^{-1}} G_j^T]$$

where

$$q^{(j+1)} = (\delta^{(j+1)} - \gamma^{(j+1)})/(\gamma^{(j+1)})^2 \quad .$$

Using the definition for D at the end of the previous section, i.e., $D = GR^{-1}$ we have:

$$d_j = G_j R_j^{-1} e_j$$

where e_j is the j^{th} column of the unit matrix. Then writing

$$r^{(j+1)} = -(\delta^{(j+1)}/\gamma^{(j+1)}) r_{j+1,j} \quad ,$$

(4.3h) becomes

$$H_{j+1} - H_j = \frac{d_j d_j^T}{\alpha_j} - \delta^{(j)} d_j d_j^T + q^{(j+1)}[H_j g_{j+1} g_{j+1}^T H_j]$$

$$+ r^{(j+1)}[d_j g_{j+1}^T H_j + H_j g_{j+1} d_j^T] \quad . \qquad (4.3i)$$

From (4.3e.1) we also get

$$H_j g_j = \delta^{(j)} d_j \quad . \qquad (4.3i.1)$$

Therefore we see that (4.3i) may be put into the form

$$H_{j+1} = H_j + p^{(j+1)} d_j d_j^T + q^{(j+1)} H_j (g_{j+1}-g_j)(g_{j+1}-g_j)^T H_j$$

$$+ r^{(j+1)}[d_j (g_{j+1}-g_j)^T H_j + H_j (g_{j+1}-g_j) d_j^T] \qquad (4.3j)$$

where

$$p^{(j+1)} = [\alpha_j^{-1} - \delta^{(j)} - q^{(j+1)} - r^{(j+1)}] \quad .$$

$p^{(j+1)}$, $q^{(j+1)}$ and $r^{(j+1)}$ are scalar parameters such that (4.3j) for $j = 1,2,\ldots,n$ has altogether $2n$ degrees of freedom. This is by virtue of the $2n$ parameters $\delta^{(1)},\ldots,\delta^{(n)}$ and $\gamma^{(1)},\ldots,\gamma^{(n)}$ being arbitrary. Expression (4.3j) is related to the updating formulae for the variable metric family defined by Huang [2] in the form stated by Powell [13]. We thus see a connection between Algorithm TC and Variable Metric Algorithms, and we discuss this again later.

4.4 Algorithm T-TC (a transformed version of Algorithm TC)

For purposes of comparison later in this chapter, we consider here a slightly different version of Algorithm TC.

Given the Hessian A and a pds matrix Q, consider Algorithm TC to be carried out using QAQ instead of A, and transformations G whose columns are now orthogonal in the Q-metric. Thus at Step (i) of Algorithm TC, instead of relations (3.4d), namely

$$G^TAG = T$$

$$G^TG = \beta = \text{diag}[\beta_1,\ldots,\beta_n]$$

we use relations (3.4e), namely

$$G^TQAQG = \bar{T}$$

$$G^TQG = \bar{\beta} \quad .$$

We insist as before that the first column of G be g_1 and continue the decomposition using relations (3.5b) in place of (3.5a). The remaining steps of Algorithm T-TC are as before.

Since Q is pds we can always find a symmetric matrix $\bar{Q}$ such that $Q = \bar{Q}^2$. We may then write the above relations as

$$(\bar{Q}G)^T(\bar{Q}A\bar{Q})(\bar{Q}G) = \bar{T} = \bar{R}^T\bar{\alpha}\bar{R}$$

$$(\bar{Q}G)^T(\bar{Q}G) = \bar{\beta} \quad .$$

Suppose we are working with the quadratic $\psi(x) = a + b^Tx + \frac{1}{2}x^TAx$. These relations may then be interpreted as tridiagonalizing the Hessian of a transformed quadratic $\bar{\psi}(y) = a + (\bar{Q}b)^Ty + \frac{1}{2}y^T\bar{Q}A\bar{Q}y$ (cf. 2.2.3), using transformations $(\bar{Q}G)$ whose columns are orthogonal in the Euclidean metric.

Successive approximations $\hat{H}_i$ to the inverse A^{-1} in

Algorithm T-TC are defined analogously to (4.3e) by

$$\hat{H}_1 = QG\left[\begin{array}{c|c} \bar{R}_j^{-1}\begin{bmatrix} \bar{\alpha}_1^{-1} & & \\ & \ddots & \\ & & \bar{\alpha}_{j-1}^{-1} \end{bmatrix}\bar{R}_j^{T^{-1}} & \begin{matrix} 0 \\ \sim \end{matrix} \\ \hline \underset{\sim}{0} & L_{j+1} \end{array}\right] G^T Q \quad .$$

The relation between this transformed version of Algorithm TC, a transformed version of the Conjugate Gradient algorithm and a class of Variable Metric Algorithms will be discussed later.

B. Methods That Use First Derivatives and Function Values

The Classical Method here is the Cauchy Method which uses search directions given by the direction of steepest descent, i.e., the negative gradient vector. This method often has an unacceptably slow rate of convergence.

Our principal thesis in this section is that a large body of algorithms which require function values and gradients may be regarded, when applied to the quadratic $\psi(x)$, as being different methods for effecting one of two important decompositions of A -- namely either the QR factorization (cf. 3.2) or the Tridiagonal-Cholesky factorization (cf. 4.2).

B.1 Methods Using QR

4.5 Basic Algorithms

The first class of methods we shall consider are based upon the

QR factorization. This class of methods is distinguished by not requiring minimal line searches. An n-search strategy is employed (cf. 2.1). Each search direction d_j is generated as a linear combination of the gradient g_j at x_j and all previous directions $d_1,\ldots,d_{j-1}$, i.e., the Direction-Gradient relation (2.5) is satisfied. Furthermore d_j is required to be conjugate to all previous search directions, $d_1,\ldots,d_{j-1}$. Assuming no g_i $(i = 1,\ldots,n)$ vanishes, i.e., that premature termination does not occur, then this class of methods, when applied to a quadratic, uses the relations

$$\begin{aligned} G &= DR \\ D^TAD &= \alpha \\ AD &= Y\lambda^{-1} \end{aligned} \tag{4.5a}$$

where $\alpha = \text{diag}[\alpha_i]$ and $\lambda^{-1} = \text{diag}[\lambda^{-1}_i]$. The third relation comes from properties of quadratics discussed in 2.2 and since minimal line searches are not used the step length may be predetermined, e.g. $\lambda_i = 1$ for all i. Recall that $Y = (g_2-g_1,\ldots,g_{n+1}-g_n)$. The first two relations are seen to be identical to those involved in a QR factorization (see 3.2). These relations (4.5a) may be written as

$$\begin{aligned} G &= DR \\ Y^TD &= \text{diag}[\alpha_i\lambda_i] = \alpha\lambda \quad . \end{aligned} \tag{4.5b}$$

Taking R to be <u>unit</u> upper triangular the recurrence relations (3.2b) of 3.2, which are an alternative expression of the above relations, now become

$$d_1 = g_1$$
$$d_j = g_j - \sum_{i=1}^{j-1} r_{ij} d_i \qquad (4.5c)$$

where

$$r_{ij} = y_i^T g_j / (y_i^T d_i)$$

and

$$y_i = (g_{i+1} - g_i)$$

whence the n-search directions may be generated. For a quadratic we obtain directions d_i that are A-orthogonal or conjugate with respect to A. From (2.3b) we have

$$A^{-1} = \sum_{1}^{n} \left(\frac{1}{\alpha_i}\right) d_i d_i^T \qquad (4.5c.1)$$

with $\alpha_i = (y_i^T d_i)/\lambda_i$. Thus for a quadratic we have (n+1) step convergence.

In implementing the above all the possibilities considered in the discussion on the QR factorization (3.2) are now available to us.

Thus, we may minimize unconstrained functions using an algorithm built around (4.5c) and (4.5c.1) which we shall call Gram-Schmidt Minimization. For implementations that use explicit projection matrices see Powell [14] or Zoutendijk [15]. In this case the process takes the form given by (3.2e), namely

$$d_i = P_i g_i \qquad i \geq 1 \qquad (4.5d)$$

where $P_1 = I$ and P_j is updated using (3.2c) or (3.2d). Together with the relation $AD = Y\lambda^{-1}$, namely

$$P_{i+1} = P_i - P_i Ad_i d_i^T A P_i / (d_i^T A P_i Ad_i) \qquad (4.5d.1)$$

and $Ad_i = y_i \lambda_i^{-1}$.

4.6 QR On a Transformed Quadratic

Now if there exists an initial pds approximation to A^{-1}, say $Q = \bar{Q}^2$, there are good reasons for working with the transformed quadratic $\bar{\psi}(y) = y + (\bar{Q}b)^T y + \frac{1}{2} y^T (\bar{Q}A\bar{Q})y$ of 2.2.3. In this case we would be dealing with a function whose Hessian is close to the unit matrix and thus very well conditioned. Moreover, the first direction of search which is along with direction of steepest descent is then close to the optimum direction used in the Newton Method. Writing

$$\bar{G} = (\bar{g}_1, \ldots, \bar{g}_n) \quad \text{where} \quad \bar{g}_i = \nabla\bar{\psi}(y)$$
$$\bar{D} = (\bar{d}_1, \ldots, \bar{d}_n)$$
$$\bar{A} = \bar{Q}A\bar{Q} \quad \text{where} \quad Q = \bar{Q}^2$$

the relations (4.5a) become

$$\left.\begin{aligned} \bar{G} &= \bar{D}\bar{R} \\ \bar{D}^T\bar{A}\bar{D} &= \bar{\alpha} \\ \bar{A}\bar{D} &= \bar{Y}\bar{\lambda}^{-1} \end{aligned}\right\} . \qquad (4.6a)$$

In the space of the original variables with $d_i = \bar{Q}\bar{d}_i$ and $g_i = (\bar{Q})^{-1}\bar{g}_i$ (cf. 2.2.3) these become

$$\left.\begin{aligned} QG &= D\bar{R} \\ D^T AD &= \bar{\alpha} \\ AD &= Y\bar{\lambda}^{-1} \end{aligned}\right\} \quad . \tag{4.6b}$$

The recurrence relations given by (4.6a) are

$$\begin{aligned} \bar{d}_1 &= \bar{g}_1 \\ \bar{d}_j &= \bar{g}_j - \sum_{i=1}^{j-1} \bar{r}_{ij}\bar{d}_i \end{aligned} \tag{4.6c}$$

where

$$\bar{r}_{ij} = (\bar{y}_i^T\bar{g}_j)/(\bar{y}_i^T\bar{d}_i)$$

and those given by (4.6b) are

$$\begin{aligned} d_1 &= Qg_1 \\ d_j &= Qg_j - \sum_{i=1}^{j-1} \bar{r}_{ij}d_i \end{aligned} \tag{4.6d}$$

where

$$\bar{r}_{ij} = (y_i^T Qg_i)/(y_i^T d_i) \quad .$$

An n-search that starts at a particular point x_1, and employs directions $d_1,\ldots,d_n$ successively generated by using (4.6d) together with some predetermined set of step lengths λ_i, $i = 1,\ldots,n$ is equivalent to an n-search starting at $\bar{Q}^{-1}x_1$ and using directions $\bar{d}_1,\ldots,\bar{d}_n$ generated from (4.6c) with λ_i the same. Successive iterates $x_1,\ldots,x_n$ and $y_1,\ldots,y_n$ obtained by these two processes satisfy $y_i = \bar{Q}^{-1}x_i$. A Gram-Schmidt minimizing algorithm for the transformed quadratic would use relations (4.6d).

4.6.1. QR Using Orthogonal Projection Matrices

Now two distinct classes of methods come from the use of orthogonal projection matrices in expressing the above basic relationships.

An important class of methods, see Hestenes [16], Powell [13] is the following. Assume that we are working with the transformed quadratic $\bar{\psi}(y)$, using relations (4.6a). Each gradient $\bar{g}_i$ is projected into the subspace orthogonal to $(\bar{A}\bar{d}_1,\ldots,\bar{A}\bar{d}_{i-1})$ using the orthogonal projector denoted by $\bar{P}_i$. This orthogonal projector, from (3.2c) applied to $\bar{\psi}(y)$, is given by

$$\begin{aligned}\bar{P}_i &= I - (\bar{A}\bar{D}_i)[(\bar{A}\bar{D}_i)^T(\bar{A}\bar{D}_i)]^{-1}(\bar{A}\bar{D}_i)^T \\ &= I - \bar{Q}(AD_i)[(AD_i)^TQ(AD_i)]^{-1}(AD_i)^T\bar{Q} \quad . \end{aligned} \tag{4.6e}$$

The process (cf. (4.5d)) is therefore:

$$\left.\begin{aligned}\bar{d}_1 &= \bar{g}_1 \\ \bar{d}_i &= \bar{P}_i\bar{g}_i \quad \text{for all} \quad i \geq 2\end{aligned}\right\} \quad . \tag{4.6f}$$

The recursive relation for $\bar{P}_i$ is obtained from (3.2d) applied to $\bar{\psi}(y)$ using $\bar{d}_i = \bar{Q}^{-1}d_i$ and $\bar{A} = \bar{Q}A\bar{Q}$

$$\bar{P}_{i+1} = \bar{P}_i - \bar{P}_i\bar{Q}Ad_id_i^TA\bar{Q}\bar{P}_i \,|\, (d_i^TA\bar{Q}\bar{P}_i\bar{Q}Ad_i) \tag{4.6g}$$

where $\bar{P}_1 = I$. Substituting for $\bar{d}_i = \bar{Q}^{-1}d_i$ and $\bar{g}_i = \bar{Q}g_i$ we have that (4.6f) becomes

$$\left.\begin{aligned}d_1 &= Qg_1 \\ d_i &= (\bar{Q}\bar{P}_i\bar{Q})g_i \quad i \geq 2\end{aligned}\right\} \quad . \tag{4.6h}$$

Writing $\hat{P}_i = (\bar{Q}\bar{P}_i\bar{Q})$ we have $\hat{P}_1 = Q$ and in its final form the process may be described by

$$d_i = \hat{P}_i g_i \qquad i \geq 1 \tag{4.6i}$$

where from (4.6e) and (4.6g), for all $i \geq 2$

$$\hat{P}_i = Q - Q(AD_i)[(AD_i)^T Q(AD_i)]^{-1}(AD_i)^T Q \tag{4.6j}$$

and

$$\hat{P}_{i+1} = \hat{P}_i - \hat{P}_i Ad_i d_i^T A\hat{P}_i / (d_i^T A\hat{P}_i Ad_i) \quad . \tag{4.6k}$$

The last relation is particularly interesting because comparing (4.6k) and (4.5d.1) shows that they are identical. Thus, the recurrence relation for the projector is invariant. Only the initial matrices P_j and $\hat{P}_j$ differ since $P_j = I$ whereas $\hat{P}_j = Q$.

Since A is not available explicitly we will, as before employ the relation $AD = Y\lambda^{-1}$ in developing P_i with a predetermined set of steps λ_i (where usually we take $\lambda_i = 1$ for all i).

4.6.2. QR Using Alternative Orthogonal Projection Matrices

The other class of methods using orthogonal projectors effect the relations (4.6b) and (4.6d) as follows:

Relation (4.6d) may be regarded as premultiplying each g_i by some pds matrix Q and then projecting Qg_i into a subspace orthogonal to $(Ad_1,\ldots,Ad_{i-1})$ using an orthogonal projector P_i^*. This is from (3.2d)

$$P_i^* = I - (AD_i)[(AD_i)^T(AD_i)]^{-1}(AD_i)^T$$

with $P_1^* = I$ and the process is given by

$$\begin{aligned} d_1 &= Qg_1 \\ d_i &= P_i^* Q g_i \quad \text{for all} \quad i \geq 2 \end{aligned} \tag{4.6ℓ}$$

together with the relation $AD = Y\lambda^{-1}$; usually $\lambda_i = 1$ for all i.

In the light of our previous discussion we know that the directions generated by (4.6i) and (4.6ℓ) must be identical. However, all we can say about the relationship between the two projectors is that

$$(P_i^* Q - \hat{P}_i) g_i = 0 \quad .$$

Because Q must be used explicitly at each iteration, (4.6ℓ) is not as attractive a process as (4.6i).

We have seen above several possible implementations each giving a different algorithm for effecting a QR factorization. For a quadratic in exact arithmetic with the same initial starting point and the same step length λ_i at corresponding iterates, the Gram-Schmidt minimizing algorithm using (4.6d), the Hestenes type algorithm using (4.6i) and the algorithm given by (4.6ℓ) would generate identical directions and give identical iterates to the minimum. Their numerical behavior in finite precision arithmetic may, of course, be quite different and is worthy of further study.

This completes our discussion of methods that use the QR factorization.

B.2 Methods Using TC

We must now justify our claim that a second important class of algorithms that uses first derivative and function value information may be regarded as implicitly carrying out a tridiagonalization followed by Cholesky factorization of A, when applied to the quadratic $\psi(x)$.

4.7 Properties of Four Fundamental Relations. Three Theorems.

In this section, as a first step, we study purely algebraically the properties of four fundamental relations. These results enable us to prove our main theorem in Part I, which identifies a set of conditions under which different algorithms give identical iterates to the minimum of a quadratic. We use this theorem in subsequent sections.

Theorem 4.1. Consider non-singular matrices satisfying the relations

$$\text{(FR)} \qquad \begin{aligned} G &= DR \\ D^TAD &= \alpha \\ AD &= GH \\ G^TG &= \beta \end{aligned} \qquad (4.7a)$$

where H is an upper Hessenberg matrix, R an upper triangular matrix and α and β are diagonal matrices.

Then H and R must be bidiagonal and

$$G^TAG = T$$

where T is tridiagonal. We shall henceforth call (4.7a) the Four Relations (FR).

Remark 1. An instance of the above four relations is obtained by augmenting (4.5a) with the additional requirements that the columns of G be orthogonal and that $g_{n+1} = 0$. The latter condition implies that the third relation of (4.5a) may be written $AD = Y\lambda^{-1} = GH$ (cf. 2.2.1).

Proof. Since we have

$$\begin{aligned} D^T &= (R^T)^{-1}G^T \quad , \\ (R^T)^{-1}G^TAD &= \alpha \quad , \\ (R^T)^{-1}G^TGH &= \alpha \quad , \\ H &= \beta^{-1}R^T\alpha \quad . \end{aligned}$$

Since R^T is lower triangular and H is upper Hessenberg, H and R^T are both of the form

$$\begin{pmatrix} x & & & & & \\ x & x & & & & \\ & x & x & & \underset{\sim}{0} & \\ & & x & x & & \\ & & & x & . & \\ & & & & . & . \\ & \underset{\sim}{0} & & & . & . \\ & & & & . & x \\ & & & & x & x \\ & & & & & x \end{pmatrix} .$$

Substituting for D in $D^TAD = \alpha$, we obtain

$$(R^T)^{-1}G^TAGR^{-1} = \alpha$$

Hence

$$\left.\begin{aligned} G^TAG &= R^T\alpha R = T \\ G^TG &= \beta \end{aligned}\right\} \qquad (4.7a.1)$$

where $R^T\alpha R$ is a tridiagonal matrix denoted by T. □

Remark 2. $H = (h_{ij})$, $R = (r_{ij})$ and $T = (t_{ij})$. If $h_{i+1,i} \neq 0$ for all i, then from the relation

$$H = \beta^{-1}R^T\alpha$$

it follows that $r_{i,i+1} \neq 0$, for all i. This in turn implies that $t_{i,i+1} \neq 0$, for all i.

Corollary. If, instead, the four relations are given by

$$\begin{aligned} QG &= D\bar{R} \\ D^TAD &= \bar{\alpha} \\ AD &= G\bar{H} \\ G^TQG &= \bar{\beta} \end{aligned} \tag{4.7b}$$

then

$$G^TQAQG = \bar{R}^T\bar{\alpha}\bar{R} = \bar{T} \tag{4.7b.1}$$

where $\bar{T}$ is also tridiagonal.

An instance of (4.7b) is obtained for (4.6b) with the additional requirements that the columns of G be orthogonal in the Q-metric and that $g_{n+1} = 0$. □

Note. Relations (4.7a.1) are precisely the TC decomposition (4.2a).

Theorem 4.2. Suppose there are two different sets of non-singular matrices G_1, D_1, R_1 & H_1 and G_2, D_2, R_2 & H_2 satisfying (4.7a) and thus (4.7a.1) by Theorem 4.1. Also let the first column

of G_1 and G_2 both be g_1, i.e., $G_1 e_1 = G_2 e_1 = g_1$. Assume $h^1_{i+1,i} \neq 0$ and $h^2_{i+1,i} \neq 0$ for all i, where $H_1 = (h^1_{ij})$ and $H_2 = (h^2_{ij})$.

Then $D_2 = D_1\omega$ where $\omega = \text{diag}[\omega_i]$ is a diagonal matrix.

Proof. From (4.7a.1)

$$G_1^T A G_1 = T_1 \quad \text{and} \quad G_2^T A G_2 = T_2$$

and

$$G_1^T G_1 = \beta \quad \text{and} \quad G_2^T G_2 = \gamma \quad .$$

From the uniqueness result, Lemma 3.1 of 3.4 and Remark 2 above, we have

$$G_1 \beta^{-1/2} = G_2 \gamma^{-1/2} \delta$$

and

$$\beta^{-1/2} T_1 \beta^{-1/2} = \delta \gamma^{-1/2} T_2 \gamma^{-1/2} \delta \quad .$$

Writing $\gamma^{-1/2} \delta \beta^{+1/2} = \Lambda$, we have

$$G_1 = G_2 \Lambda$$
$$T_1 = \Lambda T_2 \Lambda \quad .$$

Now

$$T_1 = R_1^T \alpha_1 R_1 \quad \text{and} \quad T_2 = R_2^T \alpha_2 R_2$$

where R_1 and R_2 are upper triangular. Thus

$$R_1^T \alpha_1 R_1 = \Lambda R_2^T \alpha_2 R_2 \Lambda$$

$$[\Lambda R_2^T]^{-1} R_1^T = \alpha_2 R_2 \Lambda [\alpha_1 R_1]^{-1}$$

assuming R_1 and R_2 are invertible, which will certainly be true if D_1 and D_2 are non-singular. Since the left hand side in the above relation is lower triangular and the right hand side is upper triangular each must be a diagonal matrix say, $\text{diag}[\omega_i] = \omega$. Then

$$[\Lambda R_2^T]^{-1} R_1^T = \omega$$

$$R_1 = \omega R_2 \Lambda \quad .$$

Thus

$$G_1 R_1^{-1} = G_2 \Lambda \Lambda^{-1} R_2^{-1} \omega^{-1}$$

which implies that

$$D_2 = D_1 \omega \quad . \tag{4.7c}$$

The directions defined by the i^{th} columns of D_1 and D_2 are therefore multiples of each other.

Corollary. A similar result holds if we replace, in Theorem 4.2, the conditions (4.7a) by (4.7b), namely the conditions for the Corollary to Theorem 4.1.

Using the preceding Theorems we derive our main result:

Theorem 4.3. Consider the class of algorithms that satisfy the following conditions:

(i) When applied to a quadratic each generates in general a set of n mutually conjugate directions $(d_1,\ldots,d_n)$ used in a

minimal n-search, starting from a given initial point x_1.

(ii) Implicitly or explicitly d_i is a linear combination of $g_1,\ldots,g_i$ (or stated alternatively of g_i and all previous directions $d_i,\ldots,d_{i-1}$) for $i = 1,2,\ldots,n$, and $d_1 = g_1$ is given.

Then, for a quadratic, all algorithms in this class generate an identical set of iterates $x_1,\ldots,x_n,x_{n+1}$ to the minimum.

Proof. From condition (i) above the Quadratic Termination Property, cf. 2.6 is satisfied, i.e., $G^TD = V$, where V is upper triangular. By (ii) the Direction-Gradient relation, $G = DR$, holds. Thus Lemma 2.1 implies that the gradients $G = (g_1,\ldots,g_n)$ satisfy

$$G^TG = \beta = \text{diag}[\beta_i] \quad .$$

A consequence of the Quadratic Termination Property is that the n search terminates. Also $\lambda_i \neq 0$ for all i. Thus, for a quadratic, the relation $AD = GH$ always holds (2.2.1), and $h_{i+1,i} \neq 0$ for all i.

Therefore the conditions (4.7a) are all satisfied. As proved in Theorem 4.2 any two sets of directions $D_1 = (d_1^1,\ldots,d_n^1)$ and $D_2 = (d_1^2,\ldots,d_n^2)$ satisfying (4.7a) must be such that

$$d_i^2 = \omega d_i^1 \quad ,$$

$i = 1,2,\ldots,n$ and ω_i are scalars.

Then clearly a minimal n-search that uses any set of directions for which (4.7a) holds will generate a unique set of iterates to the minimum of a quadratic.

Corollary. The result can easily be generalized to cover the case when condition (ii) above, namely the relation $G = DR$ is replaced by the more general relation $QG = DR$, where Q is pds. Each Q will determine a class of algorithms. From condition (i) we shall have $G^TD = V$. Thus $G^TQG = \bar{\beta}$ and the uniqueness of the search directions up to multiplication by scalars then follows from the Corollary to Theorem 4.2.

4.8 The Conjugate Gradient Algorithm

The relations underlying the Conjugate Gradient Algorithm applied to a quadratic $\psi(x)$, essentially amount to (4.7a). Thus, this algorithm uses a minimal n-search strategy in which the directions d_i are linear combinations of g_i and all previous directions. This is precisely the Direction Gradient Relation and R is taken to be unit upper triangular. The directions generated are conjugate. Since line searches are minimal, from the Quadratic Termination Property, $g_{n+1} = 0$ and thus the basic relation for a quadratic (2.2b) may be written as $AD = GH$. Finally, by Lemma 2.1 the columns of G are orthogonal. (Alternatively as described by Beckman [17], the Conjugate Gradient Algorithm may be viewed as performing two interleaved Gram-Schmidt orthogonalization processes. When his discussion is transformed into matrix notation it can be reduced to conditions (4.7a)).

It follows, therefore, from the preceding theorems that the Conjugate Gradient algorithm implicitly effects the TC factorization and that there is a close correspondence between the Conjugate Gradient Algorithm and Algorithm TC of 4.2. Indeed, if we consider

the latter as generating n directions given by GR^{-1}, then these directions will be multiples of those generated by the Conjugate Gradient Algorithm.

The standard recursion defining the Conjugate Gradient Process is easily deduced from the four relations and is given by

$$\left.\begin{aligned} d_1 &= g_1 \\ d_r &= g_r + \frac{\|g_r\|_E^2}{\|g_{r-1}\|_E^2} d_{r-1} \end{aligned}\right\} \qquad (4.8a)$$

where $\| \ \|_E$ denotes the Euclidean norm.

After n iterations of this process (assuming no prior termination) we have a set of conjugate directions. Also, the elements $(d_i A d_i)$ for all i can be estimated to be $(d_i^T y_i)\lambda_i^{-1} = \alpha_i$. Hence, we may obtain an estimate of the inverse to be

$$Q = \sum_{i=1}^{n} \frac{1}{\alpha_i} d_i d_i^T$$

and this estimate will be exact for a quadratic.

4.9 Conjugate Gradient Algorithm on a Transformed Quadratic

This leads us to suggest that for the subsequent n iterations we work with the transformed quadratic $\bar{\psi}(y)$ of 2.2.3. The equations underlying this process are then

$$\left.\begin{aligned} \bar{G} &= \bar{D}\bar{R} \\ \bar{D}^T\bar{A}\bar{D} &= \bar{\alpha} \\ \bar{A}\bar{D} &= \bar{G}\bar{H} \\ \bar{G}^T\bar{G} &= \bar{\beta} \end{aligned}\right\} \qquad (4.9a)$$

or in the space of the original variables

$$\left.\begin{aligned} QG &= D\bar{R} \\ D^T AD &= \bar{\alpha} \\ AD &= G\bar{H} \\ G^T QG &= \bar{\beta} \end{aligned}\right\} \qquad (4.9b)$$

Then from the Corollaries to Theorems 4.1, 4.2 and 4.3 it follows that there is a close correspondence between this algorithm and Algorithm T-TC of 4.4.

The recursion relations defining this Conjugate Gradient Process are given by relation (4.8a) applied to $\bar{\psi}(y)$

$$\bar{d}_1 = \bar{g}_1$$

$$\bar{d}_r = \bar{g}_r + \frac{\|\bar{g}_r\|_E^2}{\|\bar{g}_{r+1}\|_E^2}\,\bar{d}_{r-1}$$

where $\|\ \|_E$ denotes the Euclidean norm. Whence

$$d_1 = Qg_1$$

$$d_r = Qg_r + \frac{\|g_r\|_Q^2}{\|g_{r-1}\|_Q^2}\,d_{r-1}$$

where $\|\ \|_Q$ denotes the norm defined by psd matrix Q. However, having to store Q explicitly negates some of the advantages of the Conjugate Gradient Algorithm.

4.10 Methods That Employ the Variable Metric Relation

We now show that a large class of algorithms using the variable

metric relation 2.8 may also be regarded as effecting a TC Decomposition. We show further that successive approximations to the inverse Hessian of a quadratic obtained by these algorithms correspond to those obtained by Algorithm TC.

Variable Metric Algorithms (VMA's) use an n-search strategy whereby the search directions $d_1,\ldots,d_n$ are generated and the inverse Hessian A^{-1} developed in successive stages, for $i = 1,\ldots,n$ as follows

$$\text{(VMA)} \qquad \left.\begin{aligned} \bar{\delta}^{(i)} d_i &= H_i g_i \\ H_i A d_j &= d_j \quad \text{for all } j < i \end{aligned}\right\} \qquad (4.10a)$$

where $H_1 = Q$ a pds matrix and $\bar{\delta}^{(i)}$ is an arbitrary parameter which we have explicitly extracted from d_i. Usually H_i is defined by a recurrence relation. (We also refer the reader to the discussion in 2.8).

An important and widely used class of VMA's imposes the further requirement that the line searches be minimal. Our interest is in this case. Then from Lemma 2.2, for a quadratic $\psi(x)$, the Variable Metric Relation combined with minimal line searches implies mutual conjugacy of the search directions.

Further, we temporarily restrict attention to members of the class of Variable Metric algorithms for which $Q = I$, the identity matrix and which generate directions implicitly or explicitly satisfying the Direction-Gradient relation $G = DR$. The latter condition may easily be verified for many known algorithms from the recurrence relations that they use to define H_i. Such

algorithms then generate directions $D = (d_1,\ldots,d_n)$ and gradients $G = (g_1,\ldots,g_n)$ that obey all the conditions of Theorem 4.3 (or equivalently the Four Relations (4.7a)), and are thus members of the class of algorithms identified in that Theorem. Clearly then they implicitly effect the TC decomposition, and generate the same iterates to the minimum of a quadratic as any other member of the same class that starts from the same initial point x_1.

Since the conditions of Theorem 4.3 are equivalent to FR (4.7a) we seek to derive an expression for H_i that is consistent with FR (4.7a). This shows precisely the extent to which VMA's can differ when applied to quadratics.

Theorem 4.4. Consider the class of Variable Metric Algorithms VMA (4.10a) that generate directions and gradients satisfying the four relations FR (4.7a), namely

$$G = DR$$
$$D^TAD = \alpha$$
$$AD = GH$$
$$G^TG = \beta$$

when applied to a quadratic, & R is unit upper triangular. Then any VMA in this class gives successive approximations H_i to the inverse Hessian A^{-1} that must satisfy

$$H_i = G\left[\begin{array}{c|c} R_i^{-1}\begin{bmatrix}\alpha_1^{-1} & & \\ & \ddots & \\ & & \alpha_{i-1}^{-1}\end{bmatrix}R_i^{T^{-1}} & \begin{matrix}\\ \underset{\sim}{0}\end{matrix} \\ \hline \underset{\sim}{0} & L_{i+1}\end{array}\right]G^T \qquad (4.10b)$$

where R_i is the i^{th} leading principal submatrix of R, $\delta^{(i)}$ is an arbitrary parameter, and L_{i+1} is an arbitrary pds matrix.

Proof. In order to avoid possible confusion of notation we point out that in this proof H_i represents the i^{th} approximation to the inverse Hessian whilst H stands for the upper Hessenberg matrix in the above relations.

(i) From VMA (4.10a)

$$H_i A D E_{i-1} = D E_{i-1}$$

where $E_{i-1} = (e_1, e_2, \ldots, e_{i-1})$ and e_j is the j^{th} column of the identity matrix. Thus

$$\begin{aligned} H_i G H E_{i-1} &= G R^{-1} E_{i-1} && \text{from FR (4.7a)} \\ H_i G \beta^{-1} R^T \alpha E_{i-1} &= G R^{-1} E_{i-1} && \text{since } H = \beta^{-1} R^T \alpha \\ \beta^{-1} G^T H_i G \beta^{-1} R^T \alpha E_{i-1} &= R^{-1} E_{i-1} && \text{from Theorem 4.1} \\ \beta^{-1} G^T H_i G \beta^{-1} R^T E_{i-1} \operatorname{diag}[\alpha_1, \ldots, \alpha_{i-1}] &= R^{-1} E_{i-1} && . \end{aligned}$$

(ii) Similarly

$$\begin{aligned} H_i g_i &= \bar{\delta}^{(i)} d_i && \text{from VMA (4.10a)} \\ H_i G e_i &= \bar{\delta}^{(i)} G R^{-1} e_i \\ \beta^{-1} G^T H_i G \beta^{-1} e_i &= \bar{\delta}^{(i)} R^{-1} e_i \beta_i^{-1} && \text{since } \beta^{-1} e_i = e_i \beta_i^{-1} \quad . \end{aligned}$$

(iii) Combining (i) and (ii)

$$\beta^{-1} G^T H_i G \beta^{-1} (R^T E_{i-1}, e_i) = R^{-1} (E_{i-1}, e_i) \operatorname{diag}[\alpha_1^{-1}, \ldots, \alpha_{i-1}^{-1}, \delta^{(i)}]$$

where $\delta^{(i)} = \bar{\delta}^{(i)} / \beta_i$. Since R is unit upper triangular

$$(R^T E_{i-1}, e_i) = \left(\frac{E_i^T R^T E_i}{0}\right) \quad .$$

Writing $X = \beta^{-1} G^T H_i G \beta^{-1}$

$$X\left(\frac{E_i^T R^T E_i}{0}\right) = R^{-1} E_i \mathrm{diag}[\alpha_1^{-1}, \ldots, \alpha_{i-1}^{-1}, \delta^{(i)}] \quad .$$

Note that $(E_i^T R^T E_i)^{-1} = E_i (R^T)^{-1} E_i$.

There are standard formulae for exhibiting the full set of solutions to these overdetermined equations

$$X = R^{-1} E_i \mathrm{diag}[\alpha_1^{-1}, \ldots, \alpha_{i-1}^{-1}, \delta^{(i)}](E_i^T (R^T)^{-1} E_i | 0) + Z(I - E_i E_i^T)$$

where Z is an arbitrary matrix. We want symmetric solutions. Tidying up and writing $R_i = R^{-1} E_i$ we obtain

$$\beta^{-1} G^T H_i G \beta^{-1} = \left(\begin{array}{c|c} R_i^{-1} \mathrm{diag}[\alpha_1^{-1}, \ldots, \alpha_{i-1}^{-1}, \delta^{(i)}](R_i^T)^{-1} & 0 \\ \hline 0 & 0 \end{array}\right)$$

$$+ \left(\begin{array}{c|c} 0 & 0 \\ \hline 0 & L_{i+1} \end{array}\right)$$

when L_{i+1} is an arbitrary positive definite symmetric matrix (since we want the left hand side to be symmetric and positive definite).

Now $G^T G = \beta \Rightarrow \beta^{-1} G^T = G^{-1}$. Hence

$$H_i = G\left[\begin{array}{c|c} R_i^{-1}\mathrm{diag}[\alpha_1^{-1},\ldots,\alpha_{i-1}^{-1},\delta^{(i)}](R_i^T)^{-1} & 0 \\ \hline 0 & L_{i+1} \end{array}\right]G^T \quad (4.10c)$$

□

Comparing (4.10c) with the i^{th} approximation to the inverse in Algorithm TC shows that they are identical. The subsequent analysis of 4.3 is therefore applicable here and, in particular, we can identify two recurrence relations for H_i given by (4.3g) and (4.3j) respectively. The latter we have observed is related to the family of Huang in the form given by Powell [13].

We see that the recurrence relation (4.3g) suggests another variable metric updating technique. This is the updating formula.

$$H_{j+1} = H_j + \frac{1}{\alpha_j} d_j d_j^T - \delta^{(j)} d_j d_j^T - \delta^{(j+1)} r_{j+1,j}[d_j g_{j+1}^T + g_{j+1} d_j^T] - \gamma^{(j+1)} g_{j+1} g_{j+1}^T \quad (4.10d)$$

where $r_{j+1,j} = \dfrac{\|g_{j+1}\|_E^2}{\|g_j\|_E^2}$ may be absorbed into the arbitrary parameter $\delta^{(j+1)}$.

4.11 Discussion and Removal of Restriction $Q = I$

There is however a strong argument in favor of Huang's relation, namely (4.3j) versus (4.10d). To see this let us work with the transformed quadratic $\bar{\psi}(y)$ given by $x = \bar{Q}y$ for some pds matrix $Q = \bar{Q}^2$, cf. 2.2.3. Denoting successive approximations to the inverse Hessian by $\bar{H}_i$ the variable metric algorithms

considered in this section are defined by

$$\bar{H}_1 = I$$
$$\bar{H}_i\bar{g}_i = \bar{\delta}^{(i)}\bar{d}_i$$

and satisfy the conditions of the Corollary to Theorem 4.3 or equivalently the conditions (4.7b) and (4.7b.1). In the space of the original variables we have, using $\bar{g}_i = \bar{Q}g_i$ and $\bar{d}_i = (\bar{Q})^{-1}d_i$

$$\bar{H}_1 = I$$
$$(\bar{Q}\bar{H}_i\bar{Q})g_i = \bar{\delta}^{(i)}d_i$$

and writing $\hat{H}_i = \bar{Q}\bar{H}_i\bar{Q}$ we have

$$\left.\begin{aligned}\hat{H}_1 &= Q\\ \hat{H}_i g_i &= \bar{\delta}^{(i)}d_i\end{aligned}\right\} \qquad (4.11b)$$

Since $\bar{H}_j$ is defined by the recurrence relations (4.3j) applied to $\bar{\psi}(y)$

$$\begin{aligned}\bar{H}_{j+1} = \bar{H}_j &+ p^{(j)}\bar{d}_j\bar{d}_j^T + q^{(j)}\bar{H}_j(\bar{g}_{j+1}-\bar{g}_j)(\bar{g}_{j+1}-\bar{g}_j)^T\bar{H}_j\\ &+ r^{(j)}[\bar{d}_j(\bar{g}_{j+1}-\bar{g}_j)^T\bar{H}_j + \bar{H}_j(\bar{g}_{j+1}-\bar{g}_j)\bar{d}_j^T]\end{aligned} \qquad (4.11c)$$

it is easy to see that $\hat{H}_i$ is given by

$$\begin{aligned}\hat{H}_1 &= Q\\ \hat{H}_{j+1} &= \hat{H}_j + p^{(j)}d_jd_j^T + q^{(j)}\hat{H}_j(g_{j+1}-g_j)(g_{j+1}-g_j)^T\hat{H}_j\\ &\quad + r^{(j)}[d_j(g_{j+1}-g_j)^T\hat{H}_j + \hat{H}_j(g_{j+1}-g_j)d_j^T]\end{aligned} \qquad (4.11d)$$

i.e., apart from the initial $\bar{H}_1 = Q$, the recurrence relation defining $\hat{H}_j$ is invariant. This property is not shared by the relation in the form (4.10d) which for $\hat{H}_i$ is

$$\hat{H}_{j+1} = \hat{H}_j + \frac{1}{\alpha_j} d_j d_j^T - \delta^{(j)} d_j d_j^T - \delta^{j+1} \bar{r}_{j+1,j} [d_j g_{j+1}^T Q + Q g_{j+1} d_j^T]$$
$$+ \gamma^{(j+1)} Q g_{j+1} g_{j+1}^T Q \quad . \tag{4.11e}$$

Directions generated by a VMA using (4.11b) and (4.11d or e) satisfy the Direction-Gradient relations $QG = DR$. The Corollary of Theorem 4.3 implies that Algorithm T-TC, the transformed Conjugate Gradient Algorithm of 4.9 and the Variable Metric Algorithms given by (4.11b) and (4.11d or e) just discussed, are therefore closely interrelated, and generate search directions d_i that are scalar multiples of each other.

4.12 Quasi-Newton Methods

Quasi-Newton Methods employ the relations and results of 2.9 instead of those of 2.8. These relations define a more general class of algorithms than those of 2.8. For example, since the Quasi-Newton Relation + Minimal Line Searches do not necessarily imply the Variable Metric Relation, algorithms within this class do not necessarily have the Finite Termination Property. However, many such algorithms, e.g. see Broyden [18], when applied to a quadratic, may be shown to satisfy the conditions of Theorem 4.3. The close correspondence between such algorithms and those of previous sections of this thesis then follows from Theorem 4.3.

This completes our discussion on methods that use first derivatives and function values.

C. Algorithms That Do Not Require Derivatives

So far most of the algorithms we have considered have effected explicitly or implicitly either the LU factorization, the QR factorization or the TC factorization. The tridiagonalization involved in the last of these three corresponds to a partial solution of the algebraic eigenproblem for the Hessian of a quadratic (cf. Givens or Householder's Method discussed in Wilkinson [7]). In this section, after briefly mentioning some standard methods, we discuss an algorithm of Brodlie, for unconstrained minimization without use of derivatives, that corresponds to a complete solution of the algebraic eigenproblem. Brodlie proves that his algorithm for a quadratic is an exact parallel of a cyclic Jacobi Method applied to the Hessian, but his algorithm is described in terms which allow its application to general functions. We describe the principal steps of his algorithm and make an observation which Brodlie seems not to have mentioned, which permits a slightly different implementation of his algorithm. This observation also enables us to suggest algorithms that parallel other techniques for partially solving the eigenproblem and thus complete the unifying thread which we have attempted to draw between the numerous algorithms in the field of unconstrained minimization.

4.13 Direct Search Methods, e.g. Rosenbrock [19], the cyclic co-ordinate ascent method and the method of Powell [3] are well known algorithms in this area. These methods maintain a full set of n-search directions which are revised as the algorithm progresses. This cyclic co-ordinate ascent method and Rosenbrock's method

maintain a set of n orthogonal directions; the former always searches in fixed directions parallel to the n co-ordinate axes, whilst the latter revises the orthogonal search directions. In Powell's algorithm when applied to a quadratic, the directions are updated so as to converge to a set of mutually conjugate directions in a finite number of steps. In order to generate such directions it is necessary to perform minimal line searches. There is also a danger that these n search directions can become linearly dependent and the precautions necessary to prevent this often adversely affect the efficiency of the procedure. Brodlie [4] has suggested a method which maintains an orthogonal set of n directions at any iteration, and these are updated so as to also ensure convergence, for a quadratic, to a mutually conjugate set. Thus his method converges to directions D such that

$$\begin{aligned} D^T D &= I \\ D^T A D &= \mathrm{diag}(\mu_i) \quad . \end{aligned} \tag{4.13a}$$

This, of course, is an infinite process and if μ_i are distinct these directions, clearly the eigendirections of A, are unique. We see that more work is being done than is necessary, since all that is required for conjugacy is a set of directions orthogonal in the A metric; this set of directions is by no means unique. However, by maintaining orthogonal search direction, there is no danger of the directions becoming linearly dependent. Furthermore, this technique of updating the directions does not require minimal line searches. The method he uses for updating the search directions

is as follows. Suppose that the search directions are $D^{(k)} = (d_1^{(k)} \cdots d_n^{(k)})$ at some stage of the algorithm and the current estimate of the minimum is $x^{(k)}$.

Algorithm B:

i) Select a pair (p,q) according to some cyclic pattern C.

ii) Approximate the function $\phi(x)$ restricted to the subspace spanned by $d_p^{(k)}$ and $d_q^{(k)}$, by the following quadratic

$$\hat{\phi}^{(k)} = \hat{a} + \hat{b}\binom{\lambda}{\mu} + \frac{1}{2}(\lambda,\mu)\hat{H}\binom{\lambda}{\mu}$$

where $\hat{b}$ is a 2-element vector and $\hat{H}$ is a 2×2 matrix. The constants $\hat{a}$, $\hat{b}$ and $\hat{H}$ may be determined by fitting $\hat{\phi}^{(k)}$ to six function values in the subspace spanned by $d_p^{(k)}$ and $d_q^{(k)}$, no more than three of which may be colinear. Further function evaluations may be performed in this two-dimensional search process.

iii) Take $x^{(k+1)}$ to be the point at which the function had its least value in the two-dimensional search process, in step (ii).

iv) Revise the directions by

$$\begin{aligned} d_p^{(k+1)} &= d_p^{(k)} \cos\theta + d_q^{(k)} \sin\theta \\ d_q^{(k+1)} &= -d_p^{(k)} \sin\theta + d_q^{(k)} \cos\theta \\ d_j^{(k+1)} &= d_j^{(k)} \qquad \text{for all } j \neq p, q\ . \end{aligned} \tag{4.13b}$$

θ is chosen so that $d_p^{(k+1)}$, $d_q^{(k+1)}$ lie along the principal axes of the quadratic, i.e., if $\hat{H} = (\hat{h}_{ij})$ is the Hessian of $\hat{\phi}^{(k)}$

then

$$\tan 2\theta = 2\hat{h}_{12}/(\hat{h}_{11}-\hat{h}_{22}) \quad .$$

The above constitutes an iteration of the algorithm and Brodlie shows that when the algorithm is applied to a quadratic function $\psi(x) = a + bx + \frac{1}{2}x^T Ax$, the successive approximations to the set of eigenvectors are identical to those given by a cyclic Jacobi Method (with cyclic pattern given by C) applied to the Hessian A. He implements a particular choice of C, chosen in a subtle way, so that no search direction dominates in any portion of a cyclic pattern.

4.14 The following observation suggests an alternative method to the one that Brodlie uses to update his search directions. Estimates of second derivatives along the directions $d_p^{(k)}$, $d_q^{(k)}$ and $d_p^{(k)} + d_q^{(k)}$ are easily obtained. For a quadratic $\psi(x)$ this determines the values of the following three elements of $D^{(k)T}\tilde{AD}^{(k)}$: the $(p,p)^{th}$, $(q,q)^{th}$ and $(p,q)^{th}$. The first two, $d_p^{(k)}Ad_p^{(k)}$ and $d_q^{(k)}Ad_q^{(k)}$, are obtained directly from the second derivative estimates along $d_p^{(k)}$ and $d_q^{(k)}$. Also from second derivative estimates along $d_p^{(k)} + d_q^{(k)}$ we have

$$(d_p^{(k)} + d_q^{(k)})A(d_p^{(k)} + d_q^{(k)}) = d_p^{(k)}Ad_p^{(k)} + d_q^{(k)}Ad_q^{(k)} + 2d_p^{(k)}Ad_q^{(k)}$$

whence the $(p,q)^{th}$ element is given by

$$d_p^{(k)}Ad_q^{(k)} = \frac{1}{2}\{(d_p^{(k)}+d_q^{(k)})A(d_p^{(k)}+d_q^{(k)}) - d_p^{(k)}Ad_p^{(k)} - d_q^{(k)}Ad_q^{(k)}\} \quad . \tag{4.14a}$$

By the Jacobi rule 3.3 the angle of rotation in the (p,q) plane needed to reduce the $(p,q)^{th}$ element to zero is given by θ where

$$\tan 2\theta = 2d_p^{(k)}Ad_q^{(k)}/(d_p^{(k)}Ad_p^{(k)} - d_q^{(k)}Ad_q^{(k)}) \quad . \qquad (4.14b)$$

Thus, in Brodlie's algorithm we may carry out the above procedure in place of step ii) and perform the revision in step iv) using expressions (4.13b) but with θ calculated as in (4.14b).

The above idea also suggests other possible algorithms based upon the Arnoldi, Givens or Householder methods for finding the tridiagonal reduction of a matrix, see 3.5. In the above variation of the Brodlie algorithm, at any iteration k only a single off diagonal element $d_p^{(k)}Ad_q^{(k)}$ of $D^{(k)^T}AD^{(k)}$ is estimated, in order to perform a revision of the search directions. The new search directions are then revised by

$$D^{(k+1)} = D^{(k)}\Omega^{(k)}$$

where $\Omega^{(k)}$ was an elementary rotation in the (p,q) plane with angle given by θ in (4.14b). In an analogous fashion we may develop a procedure based upon a tridiagonal reduction. At any iteration k, only certain elements of $D^{(k)^T}AD^{(k)}$ need be estimated, namely, those that determine the transformation needed to revise the current directions. We close with the suggestion that it should be possible to devise algorithms for unconstrained minimization without derivatives which, for a quadratic $\psi(x)$, parallel the tridiagonalization of the Hessian coupled with Cholesky

factorization in much the same way as Brodlie's algorithm parallels a cyclic Jacobi process, and to describe these algorithms in a form that permits application to general functions. Such algorithms would for a quadratic then be equivalent to Algorithm TC of 4.2, but would differ significantly in implementation in that, at any iteration, only certain elements of $D^{(k)}AD^{(k)}$ (usually a single column) need be estimated. When applied to a non-quadratic function they would thus be quite different from Algorithm TC.

Part II

Generation of Conjugate Directions for Unconstrained Minimization without Derivatives

Chapter 5

The algorithm that we discuss and analyse in Part II of this thesis was introduced in 1.2.2. As noted there, it stems from two theorems proved by M.J.D. Powell [20].

5.1 Powell's Theorems

We use the notation ΔD to mean the absolute value of the determinant of the matrix D.

Theorem 5.1 (Powell, 1964). Given a quadratic function $\psi(x) = a + b^T x + \frac{1}{2}x^T A x$ where A is positive definite and symmetric (pds), let $(d_1,\ldots,d_n)$ be any set of n directions satisfying the normalization conditions

$$d_i^T A d_i = 1 \qquad i = 1,2,\ldots,n\ , \qquad (5.1a)$$

i.e., d_i are defined to be of unit length in the A-norm. If D is the matrix whose columns are the directions $(d_1,\ldots,d_n)$, then the maximum value of ΔD is attained if and only if the directions d_i $(i = 1,\ldots,n)$ are mutually conjugate. □

Theorem 5.2 (Powell, 1972). Let $(d_1,\ldots,d_n)$ be any set of n directions normalized to satisfy (5.1a). Let D be the matrix whose columns are the directions $(d_1,\ldots,d_n)$, and let Ω be any orthogonal matrix.

Let the columns of the matrix D given by $\bar{D} = D\Omega$, define a new set of directions $(\bar{d}_1,\ldots,\bar{d}_n)$.

Normalize each of the directions $\bar{d}_i$ so that each is of unit

length in the A-norm, thus obtaining directions d_i^* and matrix $D^* = (d_1^*,\ldots,d_n^*)$, where

$$d_i^* = \bar{d}_i/(\bar{d}_i^T A \bar{d}_i)^{1/2} \quad .$$

Then

$$\Delta D^* \geq \Delta D \tag{5.1b}$$

where Δ is defined as above. □

For proofs of these theorems we refer the reader to Powell [20].

The first theorem may be interpreted as follows: Consider the vectors $\hat{d}_i = Rd_i$ for all i, where $A = R^T R$ is the Cholesky factorization of the pds matrix A. The ordinary volume spanned by the columns of the matrix $\hat{D} = (\hat{d}_1,\ldots,\hat{d}_n)$ is given by $\Delta\hat{D}$ (see Franklin [21]). By Hadamard's Inequality this volume $\Delta\hat{D}$ is a maximum if and only if the vectors $\hat{d}_i$ are orthogonal. But since R is fixed $\Delta\hat{D} = (\Delta R)(\Delta D)$ is a maximum if and only if ΔD is a maximum, and from the definitions of $\hat{d}_i$ and R, the directions $\hat{d}_i$ are orthogonal if and only if the directions d_i are mutually conjugate. Thus, ΔD is a maximum if and only if the directions d_i are mutually conjugate. The maximum possible value of D is given by

$$\Delta D_{max} = (\Delta R)^{-1} = (\Delta A)^{-1/2} \quad . \tag{5.1c}$$

Since ΔD is a measure of closeness to mutual conjugacy of normalized search directions d_i, the second theorem states that the normalized directions d_i^* are at least as close to mutual

conjugacy as are the normalized directions d_i.

5.2 Normalization

Suppose $(\bar{d}_1,\ldots,\bar{d}_n)$ are a set of unnormalized search directions at the current iterate x^c. (We shall henceforth denote unnormalized directions by $\bar{d}_i$ and normalized directions by d_i.) In order to satisfy the normalization condition (5.1a) each direction $\bar{d}_i$ must be divided by $(\bar{d}_i^T A \bar{d}_i)^{1/2}$, its length in the A-norm. For a quadratic $\psi(x)$ this may be obtained by estimating the value of the second derivative of $\psi(x)$ at x^c in the direction $\bar{d}_i$ since:

$$\psi(x^c + \lambda\bar{d}_i) = \psi(x^c) + \lambda(b + Ax^c)^T\bar{d}_i + (\frac{\lambda^2}{2})\bar{d}_i^T A\bar{d}_i$$

and

$$\psi(x^c - \lambda\bar{d}_i) = \psi(x^c) - \lambda(b + Ax^c)^T\bar{d}_i + (\frac{\lambda^2}{2})\bar{d}_i^T A\bar{d}_i \quad .$$

Thus

$$\bar{d}_i^T A\bar{d}_i = \frac{\psi(x^c+\lambda\bar{d}_i) - 2\psi(x^c) + \psi(x^c-\lambda\bar{d}_i)}{\lambda^2} \quad .$$

Similar results hold approximately for a general smooth function $\phi(x)$ with A replaced by $A(x^c)$ and $0 < \lambda << 1$.

5.3 The Resulting Algorithm and Questions to be Discussed

The algorithm derived from these two theorems is as follows: A set of n search directions is maintained. At any iteration a search is conducted in sequence along each direction of this set, and the current estimate of the minimum improved in some way. It

is an easy matter to estimate second derivatives along each direction. Thus each direction may be normalized and the set of search directions revised by post-multiplying by a suitably chosen orthogonal transformation. This completes an iteration of the algorithm.

We shall denote the unnormalized search directions at the start of iteration k by the columns of $\bar{D}^{(k)}$, the normalized search directions at the k^{th} iteration by the columns of $D^{(k)}$ and the orthogonal transformation used during the i^{th} iteration by $\Omega^{(k)}$. In general, knowledge of the off-diagonal elements of $D^{(k)^T}AD^{(k)}$ is not explicitly available without further work and we do not therefore assume this knowledge in determining which orthogonal transformation $\Omega^{(k)}$ to use in the updating process at the k^{th} iteration. Later we shall discuss why it might be worthwhile to estimate one or more off-diagonal elements of $D^{(k)^T}AD^{(k)}$.

As pointed out in Chapter 1, 1.2.2, when an algorithm developed along the lines suggested by Powell's theorems is applied to a quadratic function $\psi(x)$, convergence of the search directions $D^{(k)}$ to mutual conjugacy is not assured. The optimal choice of $\Omega^{(k)}$ to be used to update $D^{(k)}$ are the eigenvectors of $D^{(k)^T}AD^{(k)}$. These are expensive to obtain. A more reasonable approach, and one that is in accordance with the methods of Computational Linear Algebra (cf. the methods of Givens or Householder) would be to restrict attention to a class of orthogonal transformations, hopefully well chosen, from which $\Omega^{(k)}$ at each iteration is selected. Certain questions then arise quite naturally. Suppose that at each iteration an arbitrary orthogonal transformation from this class is used to revise the search directions. Provided that no search

direction is neglected, will convergence of the search directions to mutual conjugacy with respect to A (the fixed Hessian of $\psi(x)$) always occur? If not, can cases be exhibited for which one obtains non-convergence or cycling? Can convergence always be assured by judiciously choosing, at each iteration, an orthogonal transformation from the class? Do we get convergence to fixed directions or merely to some set of directions that are mutually conjugate? What is the ultimate rate of convergence? Settling these questions is crucial to understanding the algorithm's behaviour, particularly its local behaviour in the neighborhood of a minimum, where it will be well approximated by a quadratic whenever $\phi(x)$ is smooth there.

5.4 The Algorithm Arising From Use of Plane Rotations

Let the orthogonal transformation $\Omega^{(k)}$ be selected from the class of plane rotations, cf. (3.3a.1). This seemed to us a worthwhile context within which to investigate some of the above questions for several reasons:

a) It is a natural choice, particularly in the light of the discussion on Brodlie's Algorithm in 4.13.

b) Proofs of convergence of the search directions to mutual conjugacy for this case could help illuminate the Jacobi eigenvalue process. In fact, it turned out that with suitable alterations, the proofs of convergence obtained (see Chapter 8) could be used to show the convergence of the cyclic Jacobi process, for a much larger class of cyclic patterns than has currently been proved in the literature [5].

c) An investigation of the algorithm for plane rotations could help in understanding the behaviour of and the difficulties one might encounter, with an algorithm that uses a more general class of orthogonal transformations.

We mentioned in Chapter 1 that under this restriction on $\Omega^{(k)}$, the resulting algorithm is related to, but in many respects different from the Jacobi eigenvalue process. Let us examine this further.

Suppose we are dealing with a quadratic function $\psi(x)$ and are given at a typical step k, a set of search directions $d_1^{(k)},\ldots,d_n^{(k)}$ normalized to be of unit length in the A-norm. The magnitude of the off-diagonal elements of $D^{(k)^T}AD^{(k)}$ indicate how close the columns of $D^{(k)}$ are to mutual conjugacy.

Now, revising $D^{(k)}$ at iteration k, by means of a plane rotation through angle θ, say in the (p,q) plane, gives unnormalized directions $\bar{D}^{(k+1)}$ satisfying

$$\begin{aligned}
\bar{d}_p^{(k+1)} &= d_p^{(k)} \cos\theta + d_q^{(k)} \sin\theta \\
\bar{d}_q^{(k+1)} &= -d_p^{(k)} \sin\theta + d_q^{(k)} \cos\theta \qquad (5.4a) \\
\bar{d}_r^{(k+1)} &= d_r^{(k)} \qquad \text{for all } r \neq p \text{ or } q\ .
\end{aligned}$$

At iteration k, only directions $d_p^{(k)}$ and $d_q^{(k)}$ are altered and all other directions are unchanged. The best θ to choose is the angle that makes $d_p^{(k+1)}$ and $d_q^{(k+1)}$ conjugate. This is the requirement that the $(p,q)^{th}$ element of $D^{(k)^T}AD^{(k)}$ be reduced to zero and, analogously to (3.3a) this angle is given by

$$\tan 2\theta = 2d_p^{(k)}Ad_q^{(k)} / (d_p^{(k)}Ad_p^{(k)} - d_q^{(k)}Ad_q^{(k)}) \quad . \qquad (5.4b)$$

Since $d_p^{(k)}$ and $d_q^{(k)}$ are both normalized to unity, the denominator = 0. Thus, always $\theta = \pm\pi/4$ unless $d_p^{(k)}Ad_q^{(k)} = 0$, in which case any θ will do.

We note that knowledge of the magnitude of $d_p^{(k)}Ad_q^{(k)}$ is not needed.

Thus, revising the search directions at iteration k corresponds to post-multiplication by a member of the two element set of fixed matrices $\{U_{pq}, U_{pq}^T\}$ where

$$\begin{aligned} U_{pq} &= \{u_{ij}\} \qquad (5.4c) \\ u_{ii} &= 1 \quad \text{for all} \quad i \neq p \quad \text{or} \quad q \\ u_{ij} &= 0 \quad \text{for all} \quad (i,j) \neq (p,q) \\ &\qquad \text{or} \quad (q,p) \quad \text{and} \quad i \neq j \end{aligned}$$

and the submatrix

$$\begin{pmatrix} u_{pp} & u_{pq} \\ u_{qp} & u_{qq} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \quad . \qquad (5.4d)$$

Let

$$S = \{U_{pq}, U_{pq}^T : \forall (p,q) \text{ s.t. } 1 \leq p < q \leq n\} \quad . \qquad (5.4e)$$

The natural choice would then be to select $\Omega^{(k)}$ at each iteration from this set.

Drawing an analogy with the Classical Jacobi eigenvalue process would suggest that, at each iteration, one choose the

pair (p,q) to be revised that corresponds to the maximum off-diagonal element of $D^{(k)^T}AD^{(k)}$. From a practical standpoint however, this is not feasible since it assumes knowledge of all off-diagonal elements of $D^{(k)^T}AD^{(k)}$. Since the Hessian A is not known explicitly, the estimation of all off-diagonal elements of $D^{(k)^T}AD^{(k)}$ requires a large number of function evaluations. Also, such a selection rule could result in certain search directions being neglected during a large number of iterations. We would prefer that the pairs (p,q) be selected in some fixed order. Henceforth we consider the Cyclic Selection Rule which selects successive members of a cyclic pattern, i.e., a permutation of the $n(n-1)|2$ pairs $(1,2),(1,3),\ldots,(1,n),(2,3),\ldots,(2,n),\ldots,(n-1,n)$. A sweep of $n(n-1)|2$ iterations completes a cycle, and a fresh cycle is then started. In this case, knowledge of off-diagonal elements of the Hessian A is not required, either for the choice of the pair to be revised or, as we saw above, for the orthogonal transformation to be used in revising this pair. This is in marked contrast to Brodlie's algorithm 4.13.

5.4.1. The algorithm outlined in 5.3 then specializes to Algorithm C below, whose k^{th} iteration initiated with a set of directions $(\bar{d}_k^{(k)},\ldots,\bar{d}_n^{(k)})$ is as follows:

Algorithm C:

i) Choose a pair (p,q) (called the current pair) according to some cyclic pattern.

ii) Conduct a linear search, not necessarily minimal, in sequence along the p^{th} and q^{th} search directions. Improve the

estimate of the minimum in some way. Normalize these two directions by estimating the second derivative along each direction during the search (as discussed in 5.2) thus obtaining directions $d_p^{(k)}$ and $d_q^{(k)}$. All other directions remain unaltered.

iii) Postmultiply the matrix whose columns consist of the set of search directions by $\Omega^{(k)} = U_{pq}$ or U_{pq}^T thus revising the search directions. This gives directions

$$\bar{d}_p^{(k+1)} = \frac{1}{\sqrt{2}}(d_p^{(k)} \mp d_q^{(k)})$$

$$\bar{d}_q^{(k+1)} = \frac{1}{\sqrt{2}}(\pm d_p^{(k)} + d_q^{(k)})$$

and all other directions remain unaltered. Then start iteration (k+1).

The algorithm is initiated with a set of linearly independent directions $\bar{d}_1^{(1)},\ldots,\bar{d}_n^{(1)}$ and terminates using, at step ii), some suitable criterion based upon change in current estimate of the minimum value and change in function value at this estimate during a complete cycle. See e.g. Brodlie [4].

Remarks

1. Note that at the start of any iteration the set of search directions need not be of unit length in the A-norm, when Algorithm C is applied to the quadratic $\psi(x) = a + b^Tx + \frac{1}{2}x^TAx$. After step ii) the p^{th} and q^{th} directions are normalized to unit length but this property is again destroyed after revising them at step iii). We return to this in the next section 5.4.2.

2. A more general algorithm would search, in step ii) along several directions, and employ, in step iii), a broader set of orthogonal transformations. A generalization of (5.4d) to the case when four directions are updated at any iteration, would post-multiply these, for example, by the matrix

$$\frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} . \qquad (5.4f)$$

This matrix is a typical member of a class of matrices defined analogously to (5.4d) and (5.4e). The generalization to the case when any even number of vectors are updated is clear. We shall return briefly to such transformations later.

3. Our definition of Algorithm C has been influenced by the assumption that, in general, the off-diagonal elements of $D^{(k)^T} A D^{(k)}$ are not available. In particular, note that in Algorithm C directions that are already conjugate will, nevertheless, be revised. We may amend step iii) of Algorithm C so that a mutually conjugate current pair of directions are left unchanged using considerations discussed in 4.14; namely, after obtaining $\bar{d}_p^{(k+1)}$ we could estimate $\|\bar{d}_p^{(k+1)}\|_A$. Then $d_p^{(k)} A d_q^{(k)}$ could be deduced from (4.14a). If this is zero then the original directions $d_p^{(k)}$ and $d_q^{(k)}$ could be left unchanged, since they are already conjugate. We take this up again in Chapter 7.

5.4.2. Two Perspectives on Algorithm C

Consider Algorithm C applied to the quadratic $\psi(x)$. Let the columns of $D^{(k)} = (d_1^{(k)},\ldots,d_n^{(k)})$ represent the normalized set of search directions at the start of iteration k, and $D^{(1)}$ the initial normalized set of search directions. As discussed in Remark 1 above, these are not explicitly maintained by Algorithm C at the start of any iteration. We introduce them for the purpose of analysis and during subsequent analysis we shall usually work with the normalized search directions. Let us postulate also a diagonal normalizing matrix represented by $N^{(k)}$ which restores the search directions at the end of any iteration k, to unit length in the A-norm. Again, Algorithm C does not do this explicitly.

Then the normalized directions, at the start of iteration k, are given by the columns of $D^{(k)}$

$$D^{(k)} = D^{(1)}\Omega^{(1)}N^{(1)}\Omega^{(2)}N^{(2)}\ldots\Omega^{(k-1)}N^{(k-1)} \qquad (5.4g)$$

where $\Omega^{(k)}$ is the orthogonal transformation employed by the algorithm at iteration k.

We saw from Powell's first theorem that when $\Delta D^{(k)}$ is a maximum, the columns of $D^{(k)}$ are conjugate. The magnitude of the off-diagonal elements of the following matrix $A^{(k)}$ also indicate how close to mutual conjugacy the search directions are:

$$\begin{aligned} A^{(k)} &= D^{(k)}AD^{(k)} \\ &= N^{(k-1)}\Omega^{(k-1)^T}\ldots N^{(1)}\Omega^{(1)^T}D^{(1)^T}AD^{(1)}\Omega^{(1)}N^{(1)}\ldots\Omega^{(k-1)}N^{(k-1)} . \end{aligned} \qquad (5.4h)$$

Algorithm C may be regarded as implicitly effecting a congruence transformation of A, through a sequence of similarity transformations using elements from the set S and diagonal congruence transformations for the normalization. In the Jacobi eigenvalue process there are no normalizing factors but the analogy with the Jacobi process defined in 3.3 is apparent.

A second perspective would be to consider the process as carrying out a sequence of orthogonal linear combinations in the following way: At iteration k, take a pair of directions $\bar{d}_p^{(k)}$ and $\bar{d}_q^{(k)}$ and normalize them to be of unit length in the A-norm, thus obtaining $d_p^{(k)}$ and $d_q^{(k)}$. Then replace $d_p^{(k)}$ and $d_q^{(k)}$ by vectors in the direction of their sum and difference. These lie in the plane spanned by $d_p^{(k)}$ and $d_q^{(k)}$ and correspond to the diagonals of the rhombus they define. All other directions are untouched. This is, of course, postmultiplication by $\Omega^{(k)} \in U_{pq}$ or U_{pq}^T. It is interesting to note that for a positive definite symmetric matrix A, the Jacobi Process for solving the eigenproblem may be viewed as a series of revisions of pairs of directions currently approximating the eigenvectors. However, since these directions are not normalized, the columns of the current approximation to the matrix of eigenvectors will not necessarily be of the same length, in the A-norm. Thus in revising any pair the angle θ given by (5.4b) need not be $\pm\pi/4$, since now the denominator in (5.4b) need not vanish. In particular, to determine θ it will be necessary to know the off-diagonal elements of $A^{(k)}$.

5.4.3. Basic Relations of the Algorithm

After carrying out the k^{th} iteration of Algorithm C, with (p,q) the current pair we have:

$$d_p^{(k+1)} = \frac{d_p^{(k)} \pm d_q^{(k)}}{\| d_p^{(k)} \pm d_q^{(k)} \|_A} \quad \text{and} \quad d_q^{(k+1)} = \frac{\mp d_p^{(k)} + d_q^{(k)}}{\| \mp d_p^{(k)} + d_q^{(k)} \|_A} \tag{5.4i}$$

where both upper or both lower signs are taken together. Also

$$d_r^{(k+1)} = d_r^{(k)} \quad \text{for all} \quad r \neq p \text{ or } q \, .$$

Given any such r, we have

$$\begin{aligned} \left| d_p^{(k+1)} A d_r^{(k+1)} \right| &= (d_p^{(k)} A d_r^{(k)} \pm d_q^{(k)} A d_r^{(k)}) / (\| d_p^{(k)} \pm d_q^{(k)} \|_A) \\ \left| d_q^{(k+1)} A d_r^{(k+1)} \right| &= (\mp d_p^{(k)} A d_r^{(k)} + d_q^{(k)} A d_r^{(k)}) / (\| \mp d_p^{(k)} + d_q^{(k)} \|_A) \end{aligned} \tag{5.4j}$$

Definition 1. We call $|d_i^{(k)} A d_j^{(k)}|$ the weight of pair (i,j) at iteration k. Note that in defining the weight we are using the normalized search directions.

Definition 2. If (p,q) is the current pair at the k^{th} iteration, then for any $r \neq p$ or q, we say that (p,r) and (q,r) are linked during the k^{th} iteration.

5.5 Interpretation of Convergence

We want $A^{(k)}$ in (5.4h) to converge to the unit matrix I, whence $D^{(k)}$ must tend, as $k \to \infty$, to a matrix whose columns are mutually conjugate. Henceforth when we say that the set of search

directions converge to mutual conjugacy we mean that $A^{(k)} \to I$. This does not imply that the search directions converge to a fixed set of mutually conjugate directions. To make this clear let us consider a cyclic Jacobi process applied to the Hessian A of the quadratic $\psi(x)$ (and assume that A has distinct eigenvalues). Then the eigendirections defined by the set of eigenvectors are unique. By taking a sufficiently large number of steps one can ensure that two consecutive iterations of this Cyclic Jacobi Process give approximations to the eigendirections that are arbitrarily close to each other and to the fixed set of unique eigendirections of A. Brodlie's algorithm will converge to this set of directions. In contrast, if Algorithm C is applied to this quadratic $\psi(x)$, the sets of search directions in two consecutive iterations can differ substantially for any $A^{(k)}$ arbitrarily close to the identity matrix I. Two directions $d_p^{(k)}$ and $d_q^{(k)}$ satisfying $|d_p^{(k)} A d_q^{(k)}| = \varepsilon$, $0 < \varepsilon << 1$ are nevertheless replaced by directions that could make an angle of as much as $\pi/4$ with the original directions. The final directions obtained will depend upon the initial directions and the complete sequence of $\Omega^{(k)}$ used. One can only state with assurance that $D^{(k)}$ can be made arbitrarily close to the class of mutually conjugate directions whenever $A^{(k)} \to I$.

These observations are not influenced by changing step iii) of Algorithm C so that already conjugate directions are not revised as discussed in Remark 3 above. There will still always be some current pair (p,q) during the course of the iteration for which $d_p^{(k)} A d_q^{(k)}$ is non-zero, though perhaps arbitrarily small.

5.6 Overview

We study the questions raised at the end of 5.3 within the context of subsequent sections.

We saw in 5.4.1 that the subset of plane rotations given by S (5.4e) arises very naturally. In the next chapter we show that given an arbitrary cyclic pattern there must always exist a sequence of orthogonal transformations $\Omega^{(k)}$ chosen from S such that the search directions converge to mutual conjugacy. Then, in Chapter 7, we show that there are certain cyclic patterns for which a misguided policy for choosing the orthogonal transformation at each iteration from S can lead to cycling of the elements of A. The proofs and examples are sensitive to changes in the definition of the algorithm but various versions that we consider are all shown to have associated difficulties when seeking to ensure convergence of the search directions to mutual conjugacy. By considering, in Chapter 8, a restricted class of cyclic patterns P we get around these difficulties and a proof of convergence of the search directions to mutual conjugacy is given. We have also used the ideas underlying these proofs to show the convergence of the cyclic Jacobi process for a large class of cyclic patterns. See [5]. We close with a discussion on ultimate rate of convergence, and the use of more general classes of orthogonal transformations.

Chapter 6

The next theorem claims the existence of certain sequences of orthogonal transformations chosen from S, that ensure convergence of $D^{(k)}$. It is non-constructive in the sense that since off-diagonal elements of $D^{(k)T}AD^{(k)}$ are not available, the theorem does not enable one to know, beforehand, whether a particular policy for choosing $\Omega^{(k)}$ will succeed in generating a mutually conjugate set of directions.

Theorem 6.2. Consider Algorithm C applied to the quadratic $\psi(x)$ using an arbitrary cyclic pattern. Then there must always exist a sequence of orthogonal transformations $\Omega^{(1)},\Omega^{(2)},\ldots,\Omega^{(k)},\ldots$ (with $\Omega^{(k)}$ either U_{pq} or U_{pq}^T when (p,q) is the current pair) for which the search directions converge to a set of mutually conjugate directions.

We shall need the following two simple lemmas:

Lemma 6.1. Given search directions $d_p^{(k)}$, $d_q^{(k)}$ and $d_r^{(k)}$ such that

(i) $|d_p^{(k)}Ad_r^{(k)}| \geq \mu$

(ii) (p,q) is the current pair.

Then for at least one of the matrices U_{pq} and U_{pq}^T we must have

$$|d_p^{(k+1)}Ad_r^{(k+1)}| \geq \mu/2 \quad .$$

Proof. From the basic relations for Algorithm C (5.4j)

$$|d_p^{(k+1)}Ad_r^{(k+1)}| = \left|(d_p^{(k)}Ad_r^{(k)} \pm d_q^{(k)}Ad_r^{(k)})/(\|d_p^{(k)} \pm d_q^{(k)}\|_A)\right| . \quad (6.2a)$$

Now

$$\|d_p^{(k)} \pm d_q^{(k)}\|_A \leq 2 \quad .$$

Thus by choosing $\Omega^{(k)}$ so that both terms in (6.2a) have the same sign it follows that

$$\left|d_p^{(k+1)} A d_r^{(k+1)}\right| \geq \mu/2 \quad . \quad \square$$

Using the terminology introduced in 5.4.3 for this transformation the weight on (p,r) is at <u>most</u> halved after iteration k.

<u>Corollary</u>. If the weight on pair (p,r) exceeds μ and the pair (p,q) is revised, then after the revision the weight on at least one of the pairs (p,r) and (q,r) must exceed $\mu/2$, for any $\Omega^{(k)} \in \{U_{pq}, U_{pq}^T\}$.

<u>Lemma 6.2</u>. Suppose at iteration t the current pair is (p,r), $\Omega^{(t)} \in \{U_{pr}, U_{pr}^T\}$ and $|d_p^{(t)} A d_r^{(t)}| = \gamma$. Then

$$\Delta D^{(t+1)} = \Delta D^{(t)}/(1-\gamma^2)^{1/2} \qquad (6.2b)$$

where $\Delta D^{(t)}$ denotes the absolute value of the determinant of $D^{(t)}$.

<u>Proof</u>. From Theorem 5.2

$$\begin{aligned}
\Delta D^{(t+1)} &= 2\Delta D^{(t)}/[(\|d_p^{(t)}+d_r^{(t)}\|_A)(\|d_p^{(t)}-d_r^{(t)}\|_A)] \\
&= 2\Delta D^{(t)}/[(2+2d_p^{(t)} A d_r^{(t)})^{1/2}(2-2d_p^{(t)} A d_r^{(t)})^{1/2}] \\
&= \Delta D^{(t)}/(1-\gamma^2)^{1/2} \quad . \quad \square
\end{aligned}$$

Using these two lemmas the Proof of Theorem 6.2 is as follows:

Consider any sequence of orthogonal transformations $\Omega^{(1)},\Omega^{(2)},\ldots,\Omega^{(k)},\ldots$. At iteration k the orthogonal transformation $\Omega^{(k)}$ is either U_{pq} or U_{pq}^T if the current pair is (p,q).

From Theorem 5.2, the successive values $\Delta D^{(k)}$ form a monotonically non-decreasing sequence for which $(\Delta A)^{-1/2}$ is an upper bound. Therefore $\Delta D^{(k)}$ must tend to a limit.

Suppose $\Delta D^{(k)} \to (\Delta A)^{-1/2} - \delta$ as $k \to \infty$, $0 < \delta < 1$. There must exist $\beta > 0$ such that at any iteration k, $|d_i^{(k)} A d_j^{(k)}| \geq \beta$ for some pair (i,j) dependent on k. If this were not true $\Delta D^{(k)}$ could be made arbitrarily close to $(\Delta A)^{-1/2}$. Also given ε $(0 < \varepsilon << 1)$ take k such that $\Delta D^{(k)} > (\Delta A)^{-1/2} - \delta - \varepsilon$, for all $k > K$.

Consider, therefore, Algorithm C at the start of a cycle of iterations and suppose that the number of previous iterations exceeds K. From the above discussion, there must be some pair say (p,r) for which $|d_p^{(k)} A d_r^{(k)}| \geq \beta$.

Proceeding through this cycle of iterations suppose that at any iteration k:

(i) the current pair includes neither of the indices p nor r. Then

$$|d_p^{(k+1)} A d_r^{(k+1)}| = |d_p^{(k)} A d_r^{(k)}| \quad .$$

(ii) the current pair includes either p or r. Without loss of generality take the current pair to be (p,q). Then by Lemma 6.1,

$$|d_p^{(k+1)} A d_r^{(k+1)}| \geq \beta/2 \qquad (6.2c)$$

for at least one member of U_{pq} or U_{pq}^T. Carry out iteration k, with $\Omega^{(k)}$ replaced by this member and continue.

Proceeding in this way the pair (p,r) must be encountered before the end of the cycle, say at iteration t. We must have

$$|d_p^{(t)} A d_r^{(t)}| \geq \beta/2^M$$

where $M = 2(n-1)$ is the number of pairs which includes either p or r.

Then from Lemma 6.2,

$$\Delta D^{(t+1)} = \Delta D^{(t)} / [1 - (\beta^2/2^{2M})]^{1/2} \quad .$$

Since

$$\Delta D^{(t)} > (\Delta A)^{-1/2} - \delta - \varepsilon$$

it follows that

$$\Delta D^{(t+1)} > ((\Delta A)^{-1/2} - \delta - \varepsilon) / [1 - (\beta^2/2^{2M})]^{1/2} \quad .$$

Taking

$$\varepsilon < [(\Delta A)^{-1/2} - \delta][1 - (1 - \beta^2/2^{2M})^{1/2}]$$

we must have

$$\Delta D^{(t+1)} > (\Delta A^{-1/2} - \delta) \quad .$$

Thus by replacing certain members (at step (ii) above) of the original sequence $\Omega^{(1)}, \Omega^{(2)}, \ldots, \Omega^{(k)}, \ldots$, we have obtained search directions which are closer to mutual conjugacy than any set of

directions generated by the original sequence, even in the limit.

Since the above argument holds for any sequence of orthogonal transformations, there must exist a sequence for which the search directions generated converge to mutual conjugacy. □

Chapter 7

Cycling in Algorithm C

We now consider the question of whether the search directions generated by Algorithm C must always converge to mutual conjugacy or whether there exist cyclic patterns and sequences of orthogonal transformations chosen from S for which the search directions do not converge.

We exhibit examples which demonstrate that a misguided policy for choosing the orthogonal transformations can lead to cycling of the elements of $D^{(k)^T}AD^{(k)}$. Some of these examples are related to examples published by Hansen [22] for the Jacobi eigenvalue process, but cycling in Algorithm C has certain distinctive features not shared by the Jacobi process.

7.1 Example of Cycling

Given a quadratic function in four variables, let us seek its minimum using Algorithm C, with cyclic pattern

$$(2,3),(1,4),(1,3),(2,4),(1,2),(3,4),\ldots \quad . \qquad (7.1a)$$

Suppose that the initial normalized search directions $d_1^{(1)},\ldots,d_4^{(1)}$ satisfy

$$A^{(1)} = D^{(1)^T}AD^{(1)} = \begin{bmatrix} 1 & 0 & x & 0 \\ 0 & 1 & 0 & x \\ x & 0 & 1 & 0 \\ 0 & x & 0 & 1 \end{bmatrix}$$

with (2,3) the current pair.

Use the following sequence of orthogonal transformations

chosen from the set S:

$$U_{23}U_{14}U_{13}U_{24}U_{12}U_{34}U_{23}^{T}U_{14}U_{13}U_{24}^{T}U_{12}^{T}U_{34} \quad . \qquad (7.1b)$$

These are then repeated in sweeps of 12 iterations.

Thus after one iteration the second and third directions are revised using the orthogonal matrix U_{23} and the trivial normalization performed.

$$A^{(2)} = D^{(2)^T}AD^{(2)} = \begin{bmatrix} 1 & x^1 & 0 & x^1 \\ x^1 & 1 & x^1 & 0 \\ 0 & x^1 & 1 & -x^1 \\ x^1 & 0 & -x^1 & 1 \end{bmatrix}$$

where $x^1 = x/\sqrt{2}$ and (1,4) is now the current pair.

We find that after six iterations

$$A^{(7)} = D^{(7)^T}AD^{(7)} = \begin{bmatrix} 1 & 0 & x & 0 \\ 0 & 1 & 0 & -x \\ x & 0 & 1 & 0 \\ 0 & -x & 0 & 1 \end{bmatrix} \qquad (7.1c)$$

and after 12 iterations

$$D^{(13)^T}AD^{(13)} = D^{(1)^T}AD^{(1)} \quad .$$

Therefore the search directions generated by the sequence (7.1b) do not converge to mutual conjugacy.

Note however that $D^{(1)}$ and $D^{(13)}$ may be distinct. It is possible that the search directions will also cycle.

7.2 Generalization

We may extend the policy for choosing $\Omega^{(k)}$. in step iii) of Algorithm C so that directions that are already conjugate may, but need not be revised, cf. Remark 3, 5.4.1. It is then not difficult to construct examples of cycling similar to the above example, for any cyclic pattern containing the subsequence

$$(i,j),(\ell,m),(j,\ell),(i,m),(i,\ell),(j,m) \quad . \qquad (7.2a)$$

Initial normalized search directions are chosen such that $d_j^{(1)}Ad_\ell^{(1)} = d_i^{(1)}Ad_m^{(1)} = x$, with all other pairs of directions mutually conjugate.

When any member of the cyclic pattern not in the above subsequence is the current pair, then use $\Omega^{(k)} = I$. Such pairs remain conjugate throughout. Orthogonal transformations for pairs that are in (7.2a) are chosen analogously to the sequence of transformations (7.1b).

7.3 Cycling When Already Conjugate Pairs Must Not Be Revised

Search directions $d_p^{(k)}$ and $d_q^{(k)}$ that are already conjugate are revised, nevertheless, when (p,q) is the current pair in the example of 7.1. This feature of the example is somewhat unsatisfactory, although it may be justified on the grounds that the necessary information is not available to Algorithm C prior to the revision of these directions. However, as noted in Remark 3 of Algorithm C, 5.4.1, step iii) may be extended so that we go to the additional expense of estimating $d_p^{(k)}Ad_q^{(k)}$, and if $d_p^{(k)}Ad_q^{(k)} = 0$, the p^{th} and q^{th} directions are not revised.

The example given in Table 7.1 which is extracted from a comptuer run is therefore somewhat more satisfactory. The initial matrix used is a particular instance of an initial matrix of the form

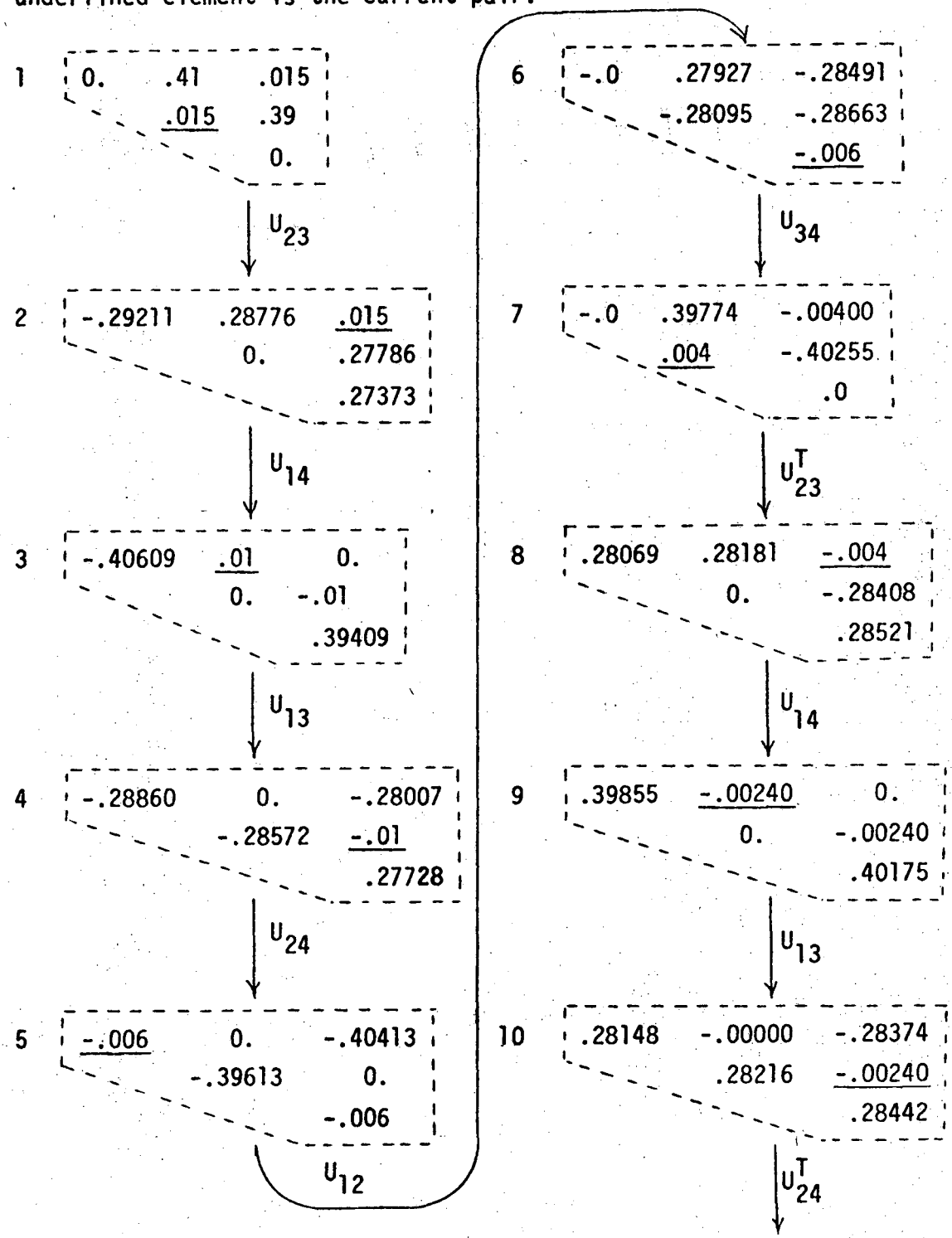
$$A^{(1)} = \begin{bmatrix} 1 & 0 & x_1 & \delta \\ 0 & 1 & \delta & x_2 \\ x_1 & \delta & 1 & 0 \\ \delta & x_2 & 0 & 1 \end{bmatrix} \qquad (7.3a)$$

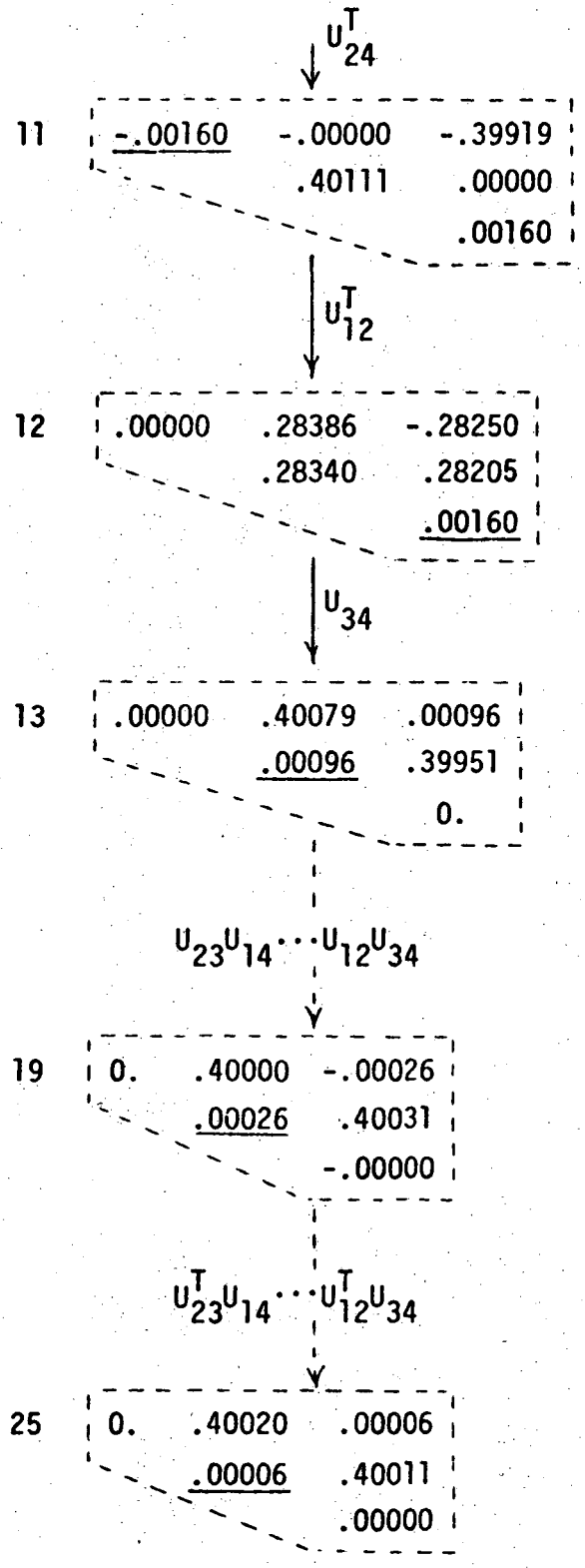
$0 < x_1 < 1$, $0 < x_2 < 1$, x_1 and x_2 are close to one another, $x_1 \neq x_2$ and $\delta \ll x_1$ or x_2. Using the same sequence of orthogonal transformations (7.1b) as before, we see from Table 7.1 that after 12 iterations, the improvement in overall conjugacy is very small and no current pair is strictly conjugate. The matrix $D^{(13)^T} A D^{(13)}$ is also of the form (7.3a). After another 12 iterations $D^{(25)^T} A D^{(25)}$ is again of this form, and during the remainder of the run this phenomenon repeated itself every 12 iterations. This is of course not conclusive evidence that $D^{(k)^T} A D^{(k)}$ does not converge to the identity matrix but we believe this to be the case. Starting from a general expression of the form $A^{(1)}$ above and examining the algebraic expressions for successive matrices $A^{(k)}$ supports this belief.

Whilst for the cyclic Jacobi process, examples related to those in 7.1 have been demonstrated by Hansen, we have not come across related examples of the form discussed in this section. Indeed, Hansen has proved that for a 4×4 matrix, no such example exists for the cyclic Jacobi process.

Table 7.1

Only superdiagonal elements of $D^{(k)^T} A D^{(k)}$ are shown. The underlined element is the current pair.




U_{24}^T
11
-.00160 -.00000 -.39919
.40111 .00000
.00160
U_{12}^T
12
.00000 .28386 -.28250
.28340 .28205
.00160
U_{34}
13
.00000 .40079 .00096
.00096 .39951
0.
$U_{23}U_{14}\cdots U_{12}U_{34}$
19
0. .40000 -.00026
.00026 .40031
-.00000
$U_{23}^TU_{14}\cdots U_{12}^TU_{34}$
25
0. .40020 .00006
.00006 .40011
.00000

7.4 Discussion

7.4.1. Implications of Above Examples for a Threshold Policy

We saw in the previous section that the policy for choosing $\Omega^{(k)}$ in Algorithm C could be extended, so that directions $d_p^{(k)}$ and $d_q^{(k)}$ are revised only if they are not already conjugate. The example of Table 7.1 then lends credence to the belief that the directions still need not converge to mutual conjugacy. The implications of this example may be carried a step further.

The above extension is a particular case of a threshold policy using a zero threshold level. In general the threshold level may be a positive number t, $0 \leq t \leq 1$. In this case only off-diagonal elements of $D^{(k)^T} A D^{(k)}$ are annihilated that are at least as large as t, in absolute value. The threshold Jacobi process, which can be shown to always converge, uses such a policy. Successive thresholds of 2^{-3}, 2^{-6}, 2^{-10}, 2^{-18}, τ, τ,...,τ,... have been suggested, where τ represents the smallest positive number that can be stored. See Wilkinson [7], page 277. Similarly a threshold level may be associated with each cycle of Algorithm C, which is then extended as follows:

During iteration k within this cycle with, say, (p,q) the current pair, at additional expense estimate $d_p^{(k)} A d_q^{(k)}$, as discussed in Remark 3 of Algorithm C, 5.4.1. If $|d_p^{(k)} A d_q^{(k)}| < t$ then the search directions $d_p^{(k)}$ and $d_q^{(k)}$ are not revised. The threshold levels are decreased after each iteration and with a well chosen threshold policy convergence can be established.

The example of Table 7.1 then illustrates the pitfalls of

a bad choice of initial and subsequent threshold levels. If successive threshold levels of $10^{-3},10^{-4},10^{-5},10^{-6},\tau,\tau,\ldots$ are taken, then the search directions will not necessarily converge to mutual conjugacy, whereas a threshold beginning with 10^{-1} will permit rapid convergence.

7.4.2. Certain Factors Upon Which Earlier Results Depend

The proofs of Theorem 6.2 and the examples of 7.1-7.3 are dependent upon the use of both positive and negative plane rotations through $\pi/4$ as equally valid choices of the orthogonal transformation Ω^k. Both Theorem 6.2 and the examples of cycling would be invalidated by changing the policy for choosing $\Omega^{(k)}$ at step iii) of Algorithm C to allow only positive (or only negative) rotations through $\pi/4$.

However, even for the policy $\Omega^{(k)} = U_{pq}$, an example exists for which there is no improvement in overall conjugacy during a single complete cycle. We refer the reader back to 7.1, where it may be seen from (7.1c) that for the cyclic pattern (7.1b),

$$\Delta D^{(7)} = \Delta D^{(1)} \quad .$$

Note however that for this policy for choosing $\Omega^{(k)}$ the cycling of off-diagonal elements of $A^{(k)}$ does not persist. During one of iterations $7,8,\ldots,12$ there will be a substantial improvement in overall conjugacy and we have not found an example for which $A^{(k)}$ does not ultimately converge to mutual conjugacy for the policy $\Omega^{(k)} = U_{pq}$.

The observations of the previous paragraph hold true for a zero threshold version of Algorithm C (i.e. $t = 0$ for every cycle) which uses any cyclic pattern containing a subsequence of the form

$$(i,j),(\ell,m),(\ell,j),(i,m) \quad , \tag{7.4a}$$

i.e., it is easy to construct an example (in a similar manner to that of 7.2) for which the policy $\Omega^{(k)} = U_{pq}$ gives an arbitrarily small increase in $\Delta D^{(k)}$ during one complete cycle of iterations and no current pair is mutually conjugate during this cycle. This means that most of the weight associated with off-diagonal elements of $D^{(k)^T} A D^{(k)}$ is pushed round ahead of the current pair during this cycle. For this reason proofs of convergence of $D^{(k)}$ to mutual conjugacy using an <u>arbitrary</u> cyclic pattern are difficult to obtain, even for the simple policy $\Omega^{(k)} = U_{pq}$ (with the revision being optional if $d_p^{(k)} A d_q^{(k)} = 0$).

Chapter 8

Convergence Proofs for a Restricted Class of Cyclic Patterns

In Theorem 6.2 we showed that when Algorithm C is applied to a quadratic and uses an arbitrary cyclic pattern, there must exist a sequence of orthogonal transformations chosen from the set S (cf. (5.4c)) for which $D^{(k)}$ converges to mutual conjugacy. We noted that Theorem 6.2 did not however prove that a particular policy for choosing the orthogonal transformations would lead to convergence of the search directions to mutual conjugacy. In Chapter 7 we gave examples of cyclic patterns for which either convergence does not occur or else proofs of convergence are difficult to obtain. In this chapter we show that convergence of the search directions to mutual conjugacy can be proven for all cyclic patterns within a certain class P, under any of the several different policies for choosing $\Omega^{(k)}$ discussed in Chapter 7.

The motivation behind choosing this class P is to exclude patterns containing a subsequence of the form (7.4a). Note, however, that this class P does not include all cyclic patterns which don't contain a subsequence (7.4a). The definition of class P characterizes a subset of such patterns.

8.1 Definition of the Class of Cyclic Patterns P

8.1.1. The class P of cyclic patterns is defined recursively using the following procedure:

Procedure P:

Step A: Given a set of directions G, partition them into two

groups G_1 and G_2. If G contains only a single member, stop.

Step B: Form a list L of pairs as follows:

Either (i) Pick one member of the first group G_1 and pair it with every member of the second group G_2 taken in any order. Repeat until all members of G_1 are exhausted.

or (ii) Carry out (i) with G_1 and G_2 interchanged.

Step C: Repeat recursively Steps A and B using the set of directions G_1 (in place of G) and put all pairs obtained at the head of list L.

Step D: Repeat recursively Steps A and B using the set of directions G_2 (in place of G) and put all pairs obtained at the tail of list L.

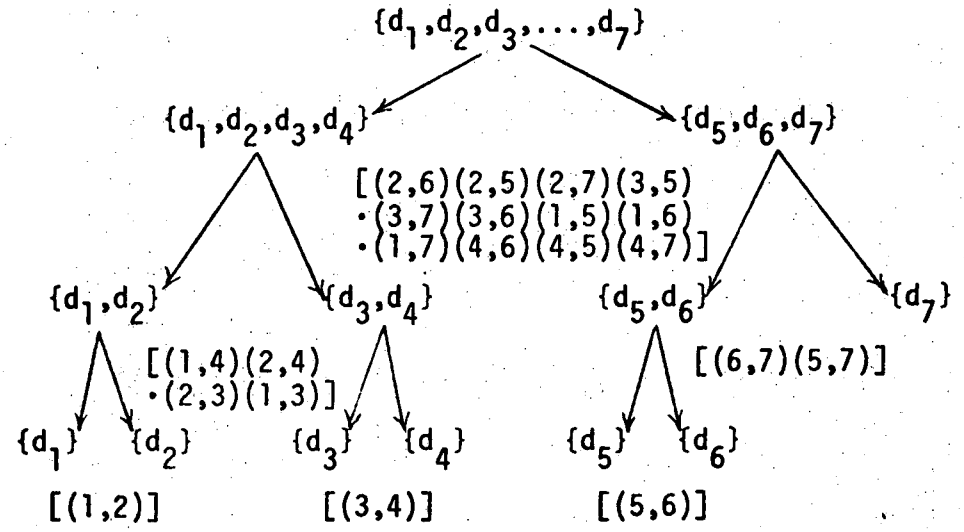
If we initiate Step A above with directions $d_1,d_2,\ldots,d_n$ then Procedure P defines the class of cyclic patterns P. An example of a cyclic pattern obtained is given in Figure 8.1, using seven directions. If all pairs of directions in this example are written as in Figure 8.2, then the cyclic pattern defined in the example is given by taking each pair of Figure 8.2 in the order determined by the number associated with it.

Consider the first partitioning of the example, i.e., $G_1 = \{d_1,d_2,d_3,d_4\}$ and $G_2 = \{d_5,d_6,d_7\}$. Referring to Figure 8.2 all pairs with both members in G_1 are contained in triangle T_1. All pairs with both members in G_2 are contained in triangle T_2. All pairs with one member in G_1 and the other in G_2 are given

by rectangle R. Forming the list L corresponds to doing Step B(i), i.e., corresponds to picking one row of rectangle R and taking all pairs in it in any order, then doing this in sequence until all rows of R are exhausted. The process is then repeated recursively within triangles T_1 and T_2. Thus another rectangle is identified within T and this time Step B (ii) is used namely, with selection by columns. The procedure terminates when no further pairs can be formed.

Figure 8.1

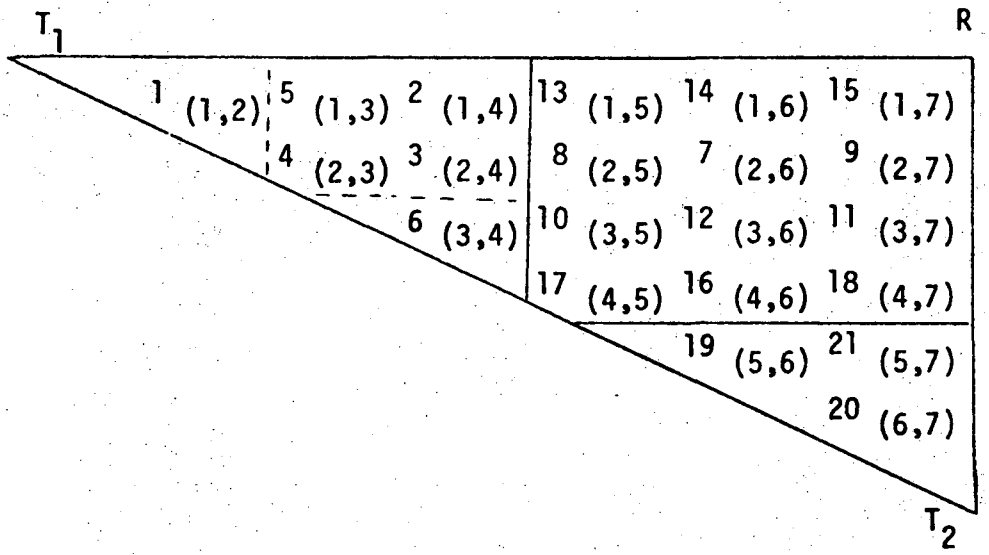
Example Illustrating Procedure P



Pattern defined

[(1,2)(1,4)(2,4)(2,3)(13)(34)(26)(25)(27)···(46)(45)(47)(56)(67)(57)]

Figure 8.2



8.1.2. Remarks on Class P

Remark 1. It may easily be verified that class P includes cyclic patterns in which pairs are taken sequentially by rows, i.e., in the order $(1,2),(1,3),\ldots,(1,n),(2,3),(2,4),\ldots,(2,n),(3,4),\ldots,(3,n),\ldots,(n-1,n)$ or sequentially by columns, i.e., in the order $(1,2),(1,3),(2,3),(1,4),(2,4),(3,4),(1,5),\ldots,(4,5),(1,6),\ldots,(1,n),\ldots,(n-1,n)$ (These are called special cyclic orderings in the context of the Jacobi process).

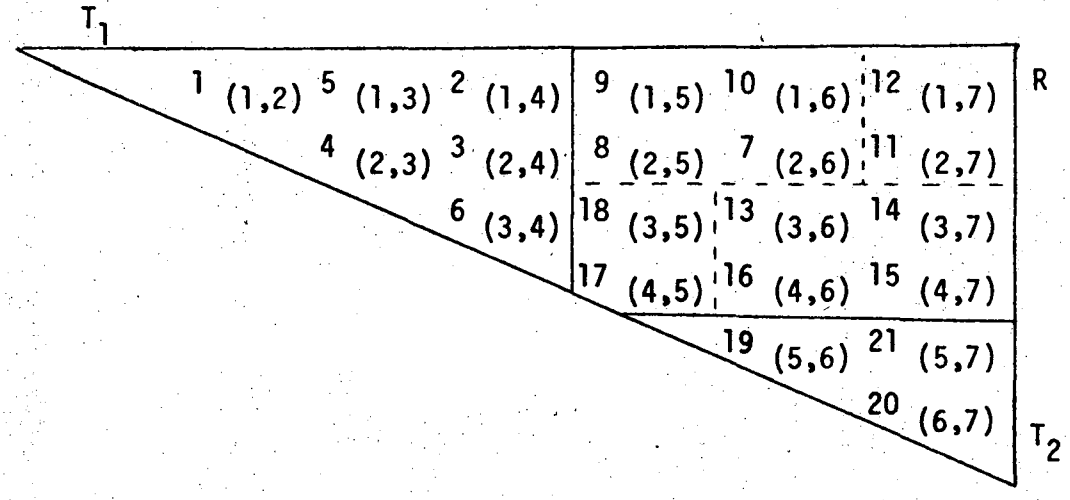
Remark 2. We point out that the class of cyclic patterns defined by Procedure P would be no more general if, at Step A of that procedure, the directions are subdivided into any two disjoint subsets instead of merely being partitioned. For example, in Figure 8.1, suppose we had subdivided $\{d_1,d_2,d_3,\ldots,d_7\}$ into $\{d_1,d_6,d_7,d_4\}$ and $\{d_2,d_3,d_5\}$, instead of partitioning into $\{d_1,d_2,d_3,d_4\}$ and $\{d_5,d_6,d_7\}$ with the list L being then formed as in Step B. Now precisely the same list would be obtained by initially permuting the directions and then partitioning. This argument may be applied at every stage of the example in Figure 8.1. It then becomes apparent that a cyclic pattern obtained by subdividing into subsets at Step A of Procedure P may also be obtained by applying Procedure P to some permuted set of the original directions.

Remark 3. A greater degree of generality is obtainable if, when forming the list L in Step B of Procedure P, we select pairs partly by rows and partly by columns. For example, we might pick

pairs in the order given in Figure 8.3 below, which should be compared with Figure 8.2. We see that the pairs in rectangle R are chosen, in Figure 8.3, in an order that cannot be obtained using Step B of Procedure P. It is possible to generalize Step B of Procedure P and thus obtain a more general class of cyclic patterns than class P. This more general class would still exclude sequences (7.4a). We only prove convergence here for cyclic patterns given by class P. Our proofs may be extended to cover the above more general class of cyclic patterns and this is detailed in [5].

Note, again that the cylic patterns defined by an extended version of Procedure P still need not embrace every cyclic pattern that does not contain a subsequence of the form (7.4a).

Figure 8.3



8.2 Proof of Convergence

We now set out to prove convergence of the search directions to mutual conjugacy using cyclic patterns in P.

We have not been able to avoid a certain degree of technical complexity in this proof. Therefore, to help guide the reader, its principal features are first outlined here:

The proof is by contradiction. Suppose when Algorithm C is applied to a quadratic $\psi(x) = a + b^T x + \frac{1}{2} x^T A x$, the search directions $D^{(k)}$ do not converge to mutual conjugacy for some cyclic pattern $e \in P$. Since, from Powell's second theorem (cf. 5.1) $\Delta D^{(k)}$ is a monotonically non-decreasing bounded sequence, it must tend to a limit as $k \to \infty$.

Now, $(\Delta A)^{-1/2}$ is the absolute value of the determinant of any matrix whose columns form a normalized set of mutually conjugate directions, by Powell's first theorem (cf. 5.1). Since we are assuming that the search directions given by columns of $D^{(k)}$ do not converge to mutual conjugacy, $\Delta D^{(k)}$ must tend to some limit strictly less than $(\Delta A)^{-1/2}$ say $(\Delta A)^{-1/2} - \delta$ where $\delta > 0$.

It is quite clear that there must exist a number $\beta > 0$ such that at any iteration there is always some pair of directions with weight (recall terminology introduced in 5.4.3) at least as great as β, i.e., if this pair is (μ,ν) then $|d_\mu^{(k)} A d_\nu^{(k)}| \geq \beta$. (With A fixed, this β is dependent only on δ). If there was no such β then $\Delta D^{(k)}$ could be made arbitrarily close to $(\Delta A)^{-1/2}$ contradicting our assumption.

Let us also assume that a sufficiently large number of iterations have been carried out so that $\Delta D^{(k)} \geq (\Delta A)^{-1/2} - \delta - \varepsilon$ for

some positive ε as small as we wish, and that we are at the start of a fresh cycle of iterations.

Some pair must have weight at least as great as β, as we have argued above. If we can claim that in proceeding through this cycle of iterations (using of course a cyclic pattern in P) we must come across some current pair, say (p,q) with weight $\geq M(n)\beta$, where $M(n)$ is a fraction dependent only on n, then we are practically home, because when pair (p,q) is revised we can obtain a contradiction to our assumption that $\Delta D^{(k)} \rightarrow (\Delta A)^{-1/2}-\delta$. To obtain this contradiction we use the above assumption that $\Delta D^{(k)} \geq (\Delta A)^{-1/2} - \delta - \varepsilon$. Taking ε sufficiently small we can appeal to Lemma 6.2 to show that after revising the current pair (p,q) we must have $\Delta D^{(k)} > (\Delta A)^{-1/2} - \delta$. □

Most of our effort therefore goes into showing the above claim namely, if some pair at the start of a cycle of iterations has weight at least as large as β, then at some point during the cycle, using a pattern $\in P$, we must come across a current pair with weight $\geq M(n)\beta$. We prove a series of lemmas leading to this result. Before each lemma we try to give some motivation for it.

Theorem 8.1. Suppose Algorithm C is applied to a quadratic and uses some cyclic pattern $\in P$. The current pair (p,q), at iteration k, is selected according to this cyclic pattern, with $\Omega^{(k)}$ either U_{pq} or U_{pq}^T (if $d_p^{(k)} A d_q^{(k)} = 0$ the directions may either be revised or left unaltered).

Then the search directions given by the columns of $D^{(k)}$

converge to mutual conjugacy, as $k \to \infty$.

To prove this theorem we shall need the following notation and several lemmas.

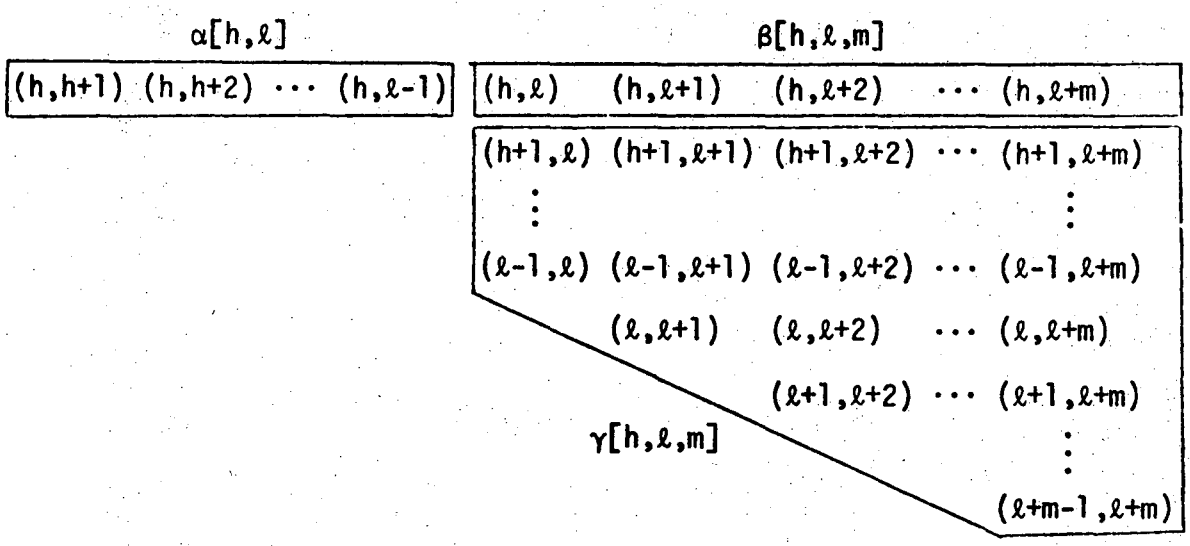
Notation.

(i) When it is unnecessary to specify the iteration number associated with a set of search directions, we shall denote the i^{th} direction in the set by $d_i^{(\ )}$ and the matrix whose columns form the set of directions by $D^{(\ )}$.

(ii) Given a set of search directions $d_h^{(\ )}, d_{h+1}^{(\ )}, \ldots, d_\ell^{(\ )}, \ldots, d_m^{(\ )}$ Figure 8.4 develops the notation for certain sets of pairs of these directions. This notation will help simplify the proofs below.

Figure 8.4

Notation



$$\alpha[h,\ell] = \{(h,j): h < j < \ell\}$$

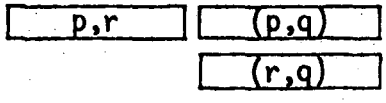
$$\beta[h,\ell,m] = \{(h,j): \ell \leq j \leq \ell+m\}$$

$$\gamma[h,\ell,m] = \{(i,j): (i<j) \;\&\; (h<i<\ell+m) \;\&\; (\ell \leq j \leq \ell+m)\}$$

$$Y[h,\ell,m] = \alpha[h,\ell] \cup \beta[h,\ell,m] \cup \gamma[h,\ell,m]$$

$$Z[h,\ell,m] = \beta[h,\ell,m] \cup \gamma[h,\ell,m]$$

Figure 8.5



The following simple lemma will prove useful later.

Lemma 8.1. Consider normalized search directions $d_h^{(k+m)},\ldots,d_\ell^{(k+m)},\ldots,d_{\ell+m}^{(k+m)}$ obtained after $(m+1)$ iterations of Algorithm C. These $(m+1)$ iterations consist of successively revising pairs from the list $(h,\ell),(h,\ell+1),\ldots,(h,\ell+m)$, i.e., from set $\beta[h,\ell,m]$ in Figure 8.4.

Let us assume that during these iterations no current pair had weight exceeding 1/2 (recall definition of weight in 5.4.3); suppose further that for some pair $(\lambda,\rho) \in Y[h,\ell,m]$ we have

$$|d_\lambda^{(k+m)} Ad_\rho^{(k+m)}| > \delta \qquad (0 < \delta < 1) \quad .$$

Then prior to carrying out these $(m+1)$ iterations (i.e., at iteration k) there must have been a pair $(\mu,\nu) \in Y[h,\ell,m]$ for which

$$|d_\mu^{(k)} Ad_\nu^{(k)}| \geq \frac{\delta}{2^{(m+1)}} \quad .$$

Proof. Suppose at some iteration the current pair is (h,q). By assumption

$$|d_h^{(\ )} Ad_q^{(\ )}| \leq \frac{1}{2} \quad .$$

Thus

$$\|d_h^{(\ )} \pm d_q^{(\ )}\| \geq 1 \quad . \qquad (8.2a)$$

With (h,q) the current pair, consider two linked pairs (h,r) and (q,r) each having weight not exceeding w. Then from

the basic relations of Algorithm C (cf. (5.4j)) together with (8.2a) above, it is obvious that after revising (h,q), the weight on neither of the two linked pairs (h,r) and (q,r) can exceed $2w$.

Suppose therefore initially at iteration k for all $(i,j) \in Y[h,\ell,m]$

$$|d_i^{(k)} A d_j^{(k)}| < \frac{\delta}{2^{(m+1)}} \quad .$$

After one iteration no pair can have weight exceeding $\delta/2^m$. Similarly after $(m+1)$ iterations no pair can have weight exceeding δ. This contradicts our assumptions.

Therefore there must exist a pair (μ,ν), at iteration k, for which

$$|d_\mu^{(k)} A d_\nu^{(k)}| \geq \frac{\delta}{2^{m+1}} \quad . \quad \square$$

An intuitive explanation of the next lemma is as follows:

Suppose the current pair at the k^{th} iteration of Algorithm C is (p,q). Consider two linked pairs (q,r) and (p,r) such that the weight on (q,r) is at least ε and that on (p,r) is at most $\varepsilon/2$. Then after revising (p,q) the weight on (q,r) must be at least $\varepsilon/4$, i.e., not all of the weight can be drawn off from (q,r).

Lemma 8.2. Suppose at iteration k of Algorithm C the current pair is (p,q) and

$$|d_q^{(k)} Ad_r^{(k)}| \geq \varepsilon \qquad (0 < \varepsilon \leq 1) \quad .$$

If $|d_p^{(k)} Ad_r^{(k)}| \leq \varepsilon/2$ then after revising (p,q)

$$|d_q^{(k+1)} Ad_r^{(k+1)}| \geq \frac{\varepsilon}{4} \quad .$$

Proof. Very similar to that of Lemma 6.1 and therefore omitted. □

Referring to Figure 8.5, Lemma 8.2 states that under certain specified conditions, the weight on (q,r) cannot all be transferred to (p,r) when revising the current pair (p,q). The following lemma is a generalization of this. Referring to Figure 8.4 it says that under certain specified conditions the total weight on $Z[h,\ell,m]$ cannot all be transferred to $\alpha[h,\ell]$ by successively revising pairs in $\beta[h,\ell,m]$.

Lemma 8.3. Given a set of search directions $d_h^{(k)},\ldots,d_\ell^{(k)},\ldots,d_{\ell+m}^{(k)}$ suppose that for some pair (λ,ρ) e $Z[h,\ell,m]$

$$|d_\lambda^{(k)} Ad_\rho^{(k)}| \geq \varepsilon \qquad 0 < \varepsilon < 1 \quad . \qquad (8.2b)$$

Suppose that $(m+1)$ further steps of Algorithm C are carried out, using successive pairs $(h,\ell),(h,\ell+1),\ldots,(h,\ell+m)$ (i.e., successive pairs from set $\beta[h,\ell,m]$).

Then there exist non-zero fractions $K_1(m)$, $K_2(m)$ and $K_3(m)$ which depend only upon m, and are monotonically non-increasing with m such that:

If

$$|d_h^{(k)} Ad_j^{(k)}| < K_1(m)\varepsilon \quad \text{for all} \quad (h,j) \in \alpha[\ell,m]$$

then at least one of the following two statements is true:

(i) There exists a pair $(\mu,\nu) \in \gamma[h,\ell,m]$ such that

$$|d_\mu^{(k+m)} Ad_\nu^{(k+m)}| \geq K_2(m)\varepsilon$$

(ii) Some current pair during these $(m+1)$ iterations has weight $\geq K_3(m)\varepsilon$.

Proof. With h and ℓ fixed, let us use induction on m.

Suppose the lemma is true for all values up to $(m-1)$. We must show it to be true for m.

1. If $\rho < m$ in (8.2b), then using the induction hypothesis and noting that the elements of $\gamma[h,\ell,m-1]$ are unaffected by revising pair (h,m), we see that the lemma holds with

$$K_i(m) = K_i(m-1) \qquad i = 1,2,3 .$$

2. If $\rho = m$, then after revising pairs $(h,\ell),\ldots,(h,\ell+m-1)$ the weight of some element in column $\ell+m$, the last column of Figure 8.4, must exceed $\varepsilon/2^m$. This follows from the Corollary to Lemma 6.1.

2.1 If this element is $(h,\ell+m)$, then (i) holds in the statement of Lemma 8.3 with $K_3(m) = 1/2^m$.

2.2 Suppose not. Say then that this element is $(i,\ell+m)$ where $h < i < \ell+m$.

2.2.1 After revising $(h,\ell+m)$ suppose the weight on $(i,\ell+m) \geq \frac{1}{4}(\frac{\varepsilon}{2^m})$. Then (i) holds in Lemma 8.3 with $K_2(m) = 1/2^{m+2}$.

2.2.2 Suppose the assumption in 2.2.1 does not hold, i.e., after revising $(h,\ell+m)$ suppose

$$|d_i^{(k+m)} Ad_{\ell+m}^{(k+m)}| < \frac{1}{4}(\frac{\varepsilon}{2^m}) \quad .$$

It then follows from Lemma 8.2 that prior to revising $(h,\ell+m)$ the weight on pair (h,i) must have been $\geq \frac{1}{2}(\frac{\varepsilon}{2^m})$. From Lemma 8.1 it follows that at the start of the process (i.e., at iteration k) some element in $Y[h,\ell,m]$ has weight $\geq \frac{1}{2}(\frac{\varepsilon}{2^m})\frac{1}{2^m}$. This pair must be in $Z[h,\ell,m]$ provided we define $K_1(m) = K_1(m-1)/2^{2m+1}$. It follows from the induction hypothesis that after revising $(h,\ell),\ldots,(h,\ell+m-1)$ at least one of the following two statements must be true:

(i) for some $(\mu,\nu) \in \gamma(h,\ell,m-1)$

$$|d_\mu^{(k+m-1)} Ad_\nu^{(k+m-1)}| \geq K_2(m-1)(\frac{\varepsilon}{2^{2m+1}})$$

(ii) some current pair has weight $\geq K_3(m-1)(\frac{\varepsilon}{2^{2m+1}})$.

Furthermore the pairs in $\gamma(h,\ell,m-1)$ are unaffected when revising the pair $(h,\ell+m)$.

3. We see therefore for all cases in 1 and 2 above, that suitable values for $K_1(m)$, $K_2(m)$ and $K_3(m)$ are

$$K_1(m) = \frac{K_1(m-1)}{2^{2m+1}} \quad ,$$

$$K_2(m) = \frac{K_2(m-1)}{2^{2m+1}} \quad ,$$

$$K_3(m) = \frac{K_3(m-1)}{2^{2m+1}} \quad .$$

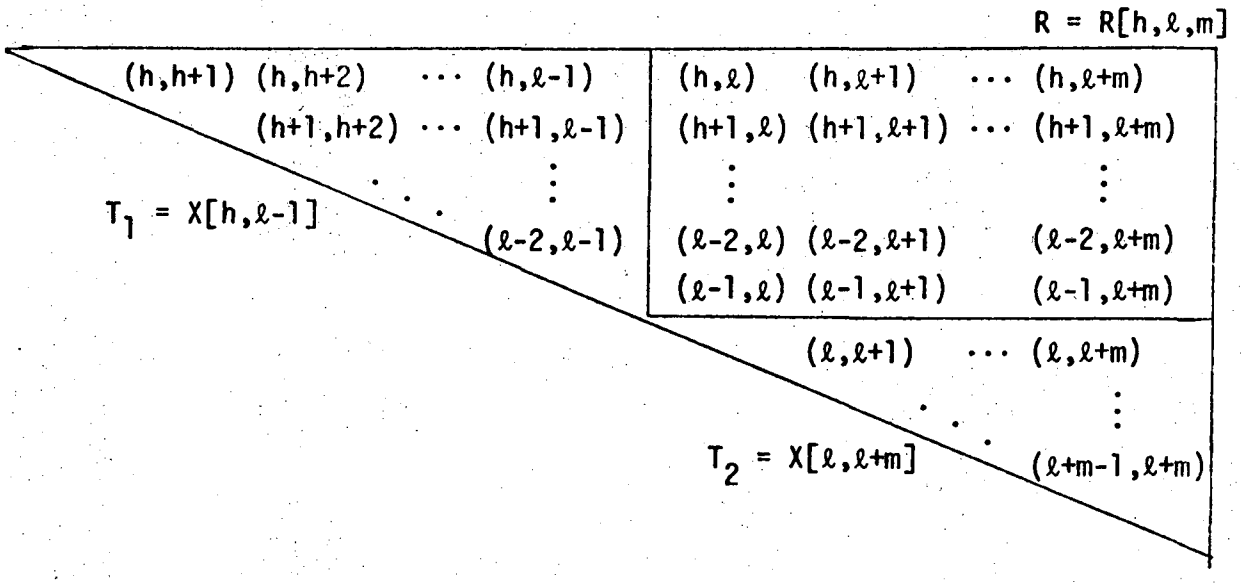
4. To complete the argument by induction we must show that Lemma 8.3 is true when $m = 0$. Suppose some element in $Z[h,\ell,0]$ has weight $\geq \varepsilon$. If this is the current pair then the lemma holds with $K_3(0) = 1$. If not, then by setting $K_1(0) = 1/2$ and $K_2(0) = 1/4$ we see that Lemma 8.3 is equivalent, for this case, to Lemma 8.2. Therefore Lemma 8.3 holds for $m = 0$.

This completes the inductive argument.

Corollary. The directions $d_{\ell}^{(k)}, d_{\ell+1}^{(k)}, \ldots, d_{\ell+m}^{(k)}$ may be permuted arbitrarily. It is clear therefore that Lemma 8.3 holds with the pairs in $\beta[h,\ell,m]$ revised in any order.

Figure 8.6

Further Notation



$$X[a,b] = \{(i,j) \mid a \leq i < j \leq b\}$$

$$R[h,\ell,m] = \{(i,j) \mid (i<j) \;\&\; (h \leq i \leq \ell-1) \;\&\; (\ell \leq j \leq \ell+m)\}$$

Lemma 8.3, we noted, was a generalization of Lemma 8.2. The next lemma is a generalization of Lemma 8.3. Referring to Figure 8.6 and using the notation developed there, Lemma 8.4 states that under certain specified conditions the total weight on pairs in $Z[h,\ell,m]$ (corresponding to triangle T_2 and rectangle R) cannot all be transferred to pairs in $X[h,\ell-1]$ (corresponding to triangle T_1) by revising pairs in rectangle R taken in sequence by rows. Note that it is our choice of P which permits the use of these sets Z and X.

Lemma 8.4. Given directions $d_h^{(k)},\ldots,d_\ell^{(k)},\ldots,d_{\ell+m}^{(k)}$ suppose that for some $(\lambda,\rho) \in Z[h,\ell,m]$

$$|d_\lambda^{(k)} A d_\rho^{(k)}| \geq \varepsilon \quad .$$

Suppose that $(\ell-h)(m+1)$ further steps of Algorithm C are carried out, using pairs selected in sequence by rows from $R[h,\ell,m]$, i.e., using successive pairs $(h,\ell),\ldots,(h,\ell+m),(h+1,\ell),\ldots,(h+1,\ell+m),\ldots,(\ell-1,\ell),\ldots,(\ell-1,\ell+m)$.

Suppose that for all $(i,j) \in X[h,\ell-1]$

$$|d_i^{(k)} A d_j^{(k)}| < K_1(m)K_2(m)^{\ell-h-1}\varepsilon \quad . \tag{8.2d}$$

Then at least one of the following two statements is true:

(i) At the end of this process some pair $(\mu,\nu) \in X[\ell,\ell+m]$ has weight $\geq K_2(m)^{\ell-h}\varepsilon$.

(ii) Some current pair during the above $(\ell-h)(m+1)$ iterations has weight $\geq K_3(m)K_2(m)^{\ell-h-1}\varepsilon$.

Proof. With ℓ and m fixed, the proof is by induction on h.

Suppose the lemma is true for directions $d_{h+1}^{(\ )},\ldots,d_{\ell}^{(\ )},\ldots,d_{\ell+m}^{(\ )}$.

We must show it to be true for directions $d_{h}^{(\ )},d_{h+1}^{(\ )},\ldots,d_{\ell}^{(\ )},\ldots,d_{\ell+m}^{(\ )}$.

Consider revising pairs $(h,\ell),(h,\ell+1),\ldots,(h,\ell+m)$.

By assumption all pairs in $X[h,\ell-1]$ satisfy (8.2d).

Thus, since $K_2(m) \leq 1$, all pairs in set $\alpha[h,\ell]$ have weight $< K_1(m)\varepsilon$.

Then by Lemma 8.3 at least one of the following statements is true.

(a) some element in $\gamma[h,\ell,m]$ has weight $\geq K_2(m)\varepsilon$

(b) some current pair has weight $\geq K_3(m)\varepsilon$.

Now, if (b) holds then for $d_{h}^{(\ )},\ldots,d_{\ell+m}^{(\ )}$ statement (ii) of Lemma 8.4 is true since $K_3(m) \geq K_3(m)[K_2(m)]^{\ell-h-1}$. Suppose therefore (b) does not hold. Then (a) must hold. Furthermore no pair in $X[h+1,\ell-1]$ is affected when revising pairs in $\beta[h,\ell,m]$ since directions $d_{h+1}^{(\ )},\ldots,d_{\ell-1}^{(\ )}$ remain unaltered. Then, by the induction hypothesis, one of the following two statements is true:

(i) at the end of the process some pair $(\mu,\nu) \in X[\ell,\ell+m]$ has weight $\geq K_2(m)^{\ell-h-1}(K_2(m)\varepsilon)$;

(ii) some current pair has weight $\geq K_3(m)K_2(m)^{\ell-h-2}(K_2(m)\varepsilon)$.

Therefore in all cases Lemma 8.4 is true for directions $d_{h}^{(\ )},\ldots,d_{\ell}^{(\ )},\ldots,d_{\ell+m}^{(\ )}$.

To complete the inductive argument, we need only observe that for directions $d_{\ell-1}^{(\ )},d_{\ell}^{(\ )},\ldots,d_{\ell+m}^{(\ )}$ Lemma 8.4 is equivalent to Lemma 8.3 and therefore the induction hypothesis holds for $h = \ell-1$. □

Corollary 1. Given two sets of directions $G_1 = \{d_h^{(\ )},\ldots,d_{\ell-1}^{(\ )}\}$ and $G_2 = \{d_\ell^{(\ )},\ldots,d_{\ell+m}^{(\ )}\}$ form a list of pairs L as in Step B(i) of Procedure P. Then Lemma 8.4 holds when successive pairs from $R[h,\ell,m]$ are revised in sequence given by list L. This follows immediately from application of the above proof to an appropriately permuted set of initial directions and use of the Corollary to Lemma 8.3.

Corollary 2. A similar result to Lemma 8.4 holds when the pairs in rectangle $R[h,\ell,m]$ of Figure 8.6 are selected in sequence given by a list L formed as in Step B(ii) of Procedure P.

Lemma 8.5. Consider Algorithm C applied to a quadratic using a cyclic pattern ϵ P. Assume we are at the start of a fresh cycle of iterations, with search directions $d_1^{(k)},\ldots,d_n^{(k)}$ such that for some pair $(\lambda,\rho) \in X[1,n]$

$$|d_\lambda^{(k)} A d_\rho^{(k)}| \geq \varepsilon \quad . \tag{8.2e}$$

Then there exists a non-zero function $M(n)$ dependent only on the number of directions, such that some current pair (p,q) has weight satisfying

$$|d_p^{(\ )} A d_q^{(\ )}| \geq M(n)\varepsilon \quad .$$

Proof. The proof is by induction on the number of directions.

Suppose Lemma 8.5 is true for up to $(n-1)$ directions. We must show it to be true for n directions.

Consider the first partition used to define the cyclic pattern employed. Say it is $\{d_1^{(\ )},\ldots,d_{\ell-1}^{(\ )}\}$ and $\{d_\ell^{(\ )},\ldots,d_n^{(\ )}\}$.

1. Suppose some element in $X[1,\ell-1]$ has weight exceeding

$$\omega = K_1(n-\ell)[K_2(n-\ell)]^{\ell-2}\varepsilon/\ 2^I \qquad (8.2f)$$

where $I = \ell(\ell-1)/2$.

Then by the induction hypothesis some current pair has weight $\geq M(\ell-1)\{K_1(n-\ell)[K_2(n-\ell)]^{\ell-2}/2^I\}\varepsilon$.

Thus Lemma 8.5 is true with

$$M(n) = M(\ell-1)\{K_1(n-\ell)[K_2(n-\ell)]^{\ell-2}/2^I\} \quad .$$

2. Suppose therefore that no element in $X[1,\ell-1]$ has weight $\geq \omega$ as defined by (8.2f). After revising all elements in $X[1,\ell-1]$ in sequence given by the cyclic pattern, no element in $X[1,\ell-1]$ has weight $\geq K_1(n-\ell)[K_2(n-\ell)]^{\ell-2}\varepsilon$. This follows from Lemma 8.1. It follows also that the pair (λ,ρ) in (8.2e) must be $\in Z[1,\ell,n-\ell]$.

Then by Lemma 8.4, one of the following two statements is true:

(i) after revising all elements in $R[1,\ell,n-\ell]$ in sequence given by the cyclic pattern, some element in $X[\ell,n]$ has weight $\geq [K_2(n-\ell)]^{\ell-1}\varepsilon$;

(ii) some current pair has weight $\geq K_3(n-\ell)[K_2(n-\ell)]^{\ell-2}\varepsilon$.

2.1 If (ii) holds then Lemma 8.5 is true with

$$M(n) = K_3(n-\ell)[K_2(n-\ell)]^{\ell-2} \quad .$$

2.2 If (i) holds, then by the induction hypothesis applied

to the directions $d_\ell^{(\ )},\ldots,d_n^{(\ )}$ some current pair has weight $\geq M(n-\ell+1)[K_2(n-\ell)]^{\ell-1}\varepsilon$. Again Lemma 8.5 holds with

$$M(n) = M(n-\ell+1)[K_2(n-\ell)]^{\ell-1} \quad .$$

3. Therefore taking

$$M(n) = \{M(n-1)K_1(n)K_3(n)[K_2(n)]^n/2^N\}$$

where $N = n(n-1)/2$ covers all cases in 1 and 2 above. We see that Lemma 8.5 is true for n directions.

4. Trivially the lemma holds for two directions, with $M(2) = 1$, completing the proof by induction.

Proof of Theorem 8.1. Using Lemma 8.5, the proof of Theorem 8.1 is straightforward.

Suppose the search directions do not converge to a mutually conjugate set.

Using an identical argument to that used in the proof of Theorem 6.2, as $k \to \infty$

$$\Delta D^{(k)} \to \Delta A^{-1/2} - \delta \qquad (\delta > 0) \quad .$$

Given ε such that $0 < \varepsilon << 1$ let $\Delta D^{(k)} > (\Delta A^{-1/2} - \delta - \varepsilon)$ for all $k > K$.

We are assuming that the search directions do not converge to a mutually conjugate set. So, again using an identical argument to that employed in Theorem 6.2, there exists $\beta > 0$ such that at any iteration k, some pair (i,j) dependent on k, has weight

exceeding β.

Assume also we are at the start of a fresh cycle. Proceeding through a complete cycle of $n(n-1)/2$ iterations it follows from Lemma 8.5 that some current pair, (p,q) has weight $\geq M(n)\beta = \gamma$ say, at iteration t.

Then, using Lemma 6.2

$$\Delta D^{(t+1)} = \frac{\Delta D^{(t)}}{(1-\gamma^2)^{1/2}} > \frac{\Delta A^{-1/2}-\delta-\varepsilon}{(1-\gamma^2)^{1/2}} \quad .$$

If ε is chosen so that

$$\varepsilon < (\Delta A^{-1/2}-\delta)[1-(1-\gamma^2)^{1/2}]$$

then

$$\Delta D^{(t+1)} > \Delta A^{-1/2} - \delta \quad .$$

This contradicts our assumption that $\Delta D^{(k)} \to \Delta A^{-1/2} - \delta$.

Therefore the n search directions $d_1^{(\ )},\ldots,d_n^{(\ )}$ must converge to a mutually conjugate set. □

8.3 Extensions

We shall not tax the reader's patience any further save to point out that in Lemma 8.5, pairs in $R[h,\ell,m]$ may be successively revised, using a more general rule for forming the list L than that used in Step B of Procedure P. A typical example of such a list was introduced earlier in Figure 8.3. Thus proofs of convergence may be obtained for a more general class of cyclic patterns that includes as a subset the class P considered here. This will be examined in more detail in [5].

Chapter 9

9.1 Discussion of Ultimate Rate of Convergence and Other Classes of Orthogonal Transformations

The derivation of the fixed set of plane rotations S used in Algorithm C applied to quadratic $\psi(x)$ requires that all search directions be normalized to be of the same length in the A-norm. This leads to an attractive and simple way to revise the search directions that does not require explicit use of the off-diagonal elements of $D^{(k)^T} A D^{(k)}$.

Let us now assume that for the cyclic pattern used the search directions do indeed converge to mutual conjugacy, and study the rate of convergence.

After the iteration has progressed sufficiently, so that the off-diagonal elements of $D^{(k)^T} A D^{(k)}$ are $O(\varepsilon)$ for some small $\varepsilon > 0$, consider a pair (p,r) that has just been revised so that its weight, which was $O(\varepsilon)$, is reduced to zero. Let us investigate the extent to which the weight on (p,r) can build up again when, later in the cycle, some other pair involving either p or r is revised. Without loss of generality say that the first such pair is (p,q). Then the weight on (p,r) immediately after revising (p,q), using U_{pq}^T, is

$$\frac{d_p^{(\ )} A d_r^{(\ )}}{\|d_p^{(\ )} + d_q^{(\ )}\|_A} + \frac{d_q^{(\ )} A d_r^{(\ )}}{\|d_p^{(\ )} + d_q^{(\ )}\|_A} = 0 + \frac{d_q^{(\ )} A d_r^{(\ )}}{\|d_p^{(\ )} + d_q^{(\ )}\|_A} = O(\varepsilon) \tag{9.1a}$$

where we may assume that the iteration has progressed sufficiently so that $\|d_p^{(\ )} + d_q^{(\ )}\|_A \geq 1$.

The weight on (p,r) has built up again to $O(\varepsilon)$, the reason for this being that $d_p^{(\ )}$ and $d_q^{(\ )}$ are replaced by conjugate directions very different from the original ones. We must curb this buildup in order to obtain a rate of convergence that is ultimately quadratic, i.e., we must modify the rule for revising the current pair, so that they are replaced by mutually conjugate directions close to the original directions. Thus in the later stages of a convergent iterative process defined by Algorithm C, when the set of search directions have become close to mutual conjugacy, it might be worthwhile to incur the additional expense of estimating $d_p^{(k)}Ad_q^{(k)}$ for the current pair (p,q), as outlined in Remark 3, 5.4.1. Once $d_p^{(k)}Ad_q^{(k)}$ is available, this opens up a Pandora's box of possible ways to revise the search directions. We do not wish to study this in detail here, but in order briefly to consider the implications, let us look at the following technique. After a certain number of iterations of Algorithm C using the fixed set S, change the rule for revising the current pair as follows: Renormalize the p^{th} and q^{th} directions to be of different lengths in the A-norm (say lengths 1 and 2), and revise these directions using a plane rotation given by (5.4b), namely,

$$\tan 2\theta = \frac{2d_p^{(k)}Ad_q^{(k)}}{d_p^{(k)}Ad_p^{(k)} - d_q^{(k)}Ad_q^{(k)}} \qquad (9.1b)$$

where $d_p^{(k)}$ and $d_q^{(k)}$ represent the renormalized directions. Now the denominator in (9.1b) does not vanish. We see that $\tan 2\theta$ is $O(d_p^{(k)} A d_q^{(k)})$ and the revised directions are given by

$$\begin{aligned} \bar{d}_p^{(k+1)} &= d_p^{(k)} \cos\theta - d_q^{(k)} \sin\theta \\ \bar{d}_q^{(k+1)} &= d_p^{(k)} \sin\theta + d_q^{(k)} \cos\theta \end{aligned} \quad . \qquad (9.1c)$$

Assuming convergence, then after the elements of $D^{(k)T} A D^{(k)}$ have become $O(\varepsilon)$, $|\sin\theta|$ is also $O(\varepsilon)$. By an argument analogous to the one used in obtaining (9.1a) we find that the build up of weight of the revised pair (p,r) is now $O(|d_q^{(\ )} A d_r^{(\ )}||\sin\theta|)$ and this is $O(\varepsilon^2)$. A formal argument establishes that during a complete cycle of iterations the buildup remains $O(\varepsilon^2)$ and we thus have an ultimate quadratic rate of convergence of the search directions to mutual conjugacy.

Ultimate quadratic convergence of the cyclic Jacobi process applied to a matrix A with distinct eigenvalues is proven by a similar technique. An important difference between the two processes, however, is that we can <u>always</u> bound θ by ensuring that the denominator in (9.1b) does not vanish. In the Jacobi process renormalizations are not possible, because they would change the eigenvalues of A and thus θ can only be bounded when the diagonal elements of $A^{(k)}$ in 3.3 converge to distinct values, i.e., when the eigenvalues of A are distinct. Thus Brodlie's algorithm, which exactly parallels the cyclic Jacobi process, can only be proven to have ultimate quadratic convergence of the search directions when the eigenvalues of the Hessian A of $\psi(x)$ are distinct,

whereas a process as we have outlined above, would have ultimate quadratic convergence to mutual conjugacy regardless of the multiplicity of the eigenvalues of A.

We have seen why it may profitable to switch in later stages of the iteration from using fixed matrices chosen from S to using more general plane rotations. We close with some brief comments on other classes of orthogonal transformations that might be used.

As noted in Remark 2 of Algorithm C, 5.4.1 the set S can be generalized, (5.4f) being a typical example for the case when four directions are simultaneously revised. Let us represent the class of such matrices, when n directions are simultaneously revised, by S[n] where n is even. Thus S = S[2].

By taking products of certain matrices in S one may obtain matrices in S[n]. For example, a matrix in S[4] is given by the product

$$U_{12}U_{34}U_{13}U_{24}$$

corresponding to the ordering (1,2)(3,4)(1,3)(2,4). This is of the form (7.4a). From the discussion of cycling in Algorithm C this implies that when using orthogonal transformations chosen from S[n] convergence of the search directions to mutual conjugacy need not occur. Also the conclusions arrived at earlier about ultimate rate of convergence apply to any fixed class of orthogonal matrices. The use of classes of orthogonal transformations given by S[n] may, however, lead to fewer normalizations and hence fewer function evaluations. This might be a fruitful topic for further investigation.

References

[1] Kowalik, J. and Osborne, M.R. (1968). Methods for Unconstrained Optimization Problems, Elsevier, New York.

[2] Huang, H.U. (1970). "Unified approach to quadratically convergent algorithms for function minimization," JOTA 5, 405-423.

[3] Powell, M.J.D. (1964). "An efficient method of finding the minimum of a function of several variables without calculating derivatives," Comput. J. 7, 155-162.

[4] Brodie, K.W. (1972). "A New Method for Unconstrained Minimization without Evaluating Derivatives," IBM Report UKSC-0019, IBM, Peterlee, United Kingdom.

[5] Nazareth, J.L. (1973). "A Class of Cyclic Patterns for which the Jacobi Process Converges," (forthcoming Technical Report).

[6] Adachi, N. (1971). "On variable-metric algorithms," JOTA 7 (6), 391-410.

[7] Wilkinson, J.H. (1965). The Algebraic Eigenvalue Problem, Oxford University Press, London.

[8] Boullion, T.L. and Odell, P.L. (1971). Generalized Inverse Matrices, Wiley, New York.

[9] Forsythe, G.E. and Henrici, P. (1960). "The cyclic Jacobi method for computing the principal values of a complex matrix," Trans. Amer. Math. Soc. 94, 1-23.

[10] Fiacco, A.V. and McCormick, G.P. (1968). Non-linear Programming: Sequential Unconstrained Minimization Techniques, John Wiley, New York.

[11] Matthews, A. and Davies, D. (1971). "A comparison of modified Newton methods for unconstrained optimization," Comput. J. 14, 293-294.

[12] Greenstadt, J.L. (1967). "On the relative inefficiencies of gradient methods," Maths. Comput. 21, 360-367.

[13] Powell, M.J.D. (1971). "Recent advances in unconstrained optimization," Mathematical Programming 1, 26-57.

[14] Powell, M.J.D. (1962). "An iterative method for finding stationary values of a function of several variables," Comput. J. 5, 147-151.

[15] Loutendijk, G. (1970). "Non-linear programming-computational methods," in J. Abadie (ed.), Integer and Non-linear Programming, North-Holland, Amsterdam.

[16] Hestenes, M.R. (1969). "Multiplier and gradient methods," in L.A. Zadeh, L.W. Neustadt and A.V. Balakrishnan (eds.), Computing Methods in Optimization Problems II, Academic Press, London and New York, 143-163.

[17] Beckman, F.S. (1960). "The solution of linear equations by the conjugate gradient method," in A. Ralston and H.S. Wilf (eds.), Mathematical Models for Digital Computers, Chapter 4, John Wiley, New York.

[18] Broyden, C.G. (1967). "Quasi-Newton methods and their application to function minimization," Maths. Comput. 21, 368-381.

[19] Rosenbrock, H.H. (1960). "An automatic method for finding the greatest or least value of a function," Comput. J. 3, 175-184.

[20] Powell, M.J.D. (1972). "Unconstrained minimization algorithms without computation of derivatives," UK.A.E.A. Harwell Report HL72/1713.

[21] Franklin, J.N. (1968). Matrix Theory, Prentice-Hall, New Jersey.

[22] Hansen, Eldon R. (1963). "On cyclic Jacobi methods," J. Soc. Indust. Appl. Math. 11 (2), 448-459.

Other references used but not directly cited:

[23] Bartels, R.H., Golub, G.H. and Saunders, M.A. (1970). "Numerical techniques in mathematical programming," Stanford Computer Science Department Report #70-162, 1-60.

[24] Broyden, C.G. (1970). "The convergence of a class of double-rank minimization algorithms, 1. General considerations," J. Inst. Maths. Applics. 6, 76-90.

[25] Davidon, W.C. (1959). "Variable metric method for minimization," A.E.C. Research and Development Report ANL-5990 (Rev.).

[26] Dixon, L.C.W. (1972). "Quasi-Newton techniques generate identical points II: The proofs of four new theorems," Mathematical Programming 3, 345-358, North-Holland Publishing Company.

[27] Dixon, L.C.W. (1972). "Variable metric algorithms: necessary and sufficient conditions for identical behavior of nonquadratic functions," JOTA 10 (1), 34-40.

[28] Fletcher, R. and Reeves, C.M. (1964). "Function minimization by conjugate gradients," Computer Journal 7, 149-154.

[29] Fletcher, R. and Powell, M.J.D. (1963). "A rapidly convergent descent method for minimization," Computer Journal 6 (2).

[30] Gill, P.E., Golub, G.H., Murray, W. and Saunders, M.A. (1972). "Methods for modifying matrix factorizations," Stanford Computer Science Report STAN-CS-72-322.

[31] Henrici, Peter (1958). "On the speed of convergence of cyclic and quasicyclic Jacobi methods for computing eigenvalues of Hermitian matrices," J. Soc. Indust. Appl. Math. 6 (2), 144-162.

[32] Hestenes, M.R. and Stiefel, E. (1952). "Methods of conjugate gradients for solving linear systems," Journal of Research of the National Bureau of Standards 49 (6).

[33] Kemper, H.P.M. Van (1966). "On the convergence of the classical Jacobi method for real symmetric matrices with non-distinct eigenvalues," Numerische Mathematik 9, 11-18.

[34] Murray, W. (1972). Numerical Methods for Unconstrained Optimization, Academic Press, London & New York.

[35] Murtagh, B.A. and Sargent, R.W.H. (1969). "A constrained minimization method with quadratic convergence," in R. Fletcher (ed.), Optimization, Academic Press, London, 215-246.

[36] Myers, G.E. (1968). "Properties of the conjugate-gradient and Davidon methods," JOTA 2 (4), 209-219.

[37] Osborne, Michael (1972). "Topics in optimization," Stanford Computer Science Report 72-279, 1-140.

[38] Pearson, J.D. (1969). "On variable-metric methods of minimization," Computer Journal 12 (2).

[39] Powell, M.J.D. (1970). "A survey of numerical methods for unconstrained optimization," SIAM Review 12, 79-97.

[40] Powell, M.J.D. (1972). "Unconstrained minimization and extensions for constraints," U.K.A.E.A. Research Report, Harwell, United Kingdom, 1-49.

[41] Wilkinson, J.H. (1962). "Note on the quadratic convergence of the cyclic Jacobi process," Numerische Mathematik 4, 296-300.

[42] Zangwill, W. (1969). Nonlinear Programming, A Unified Approach, Prentice-Hall, New Jersey.

TECHNICAL INFORMATION DIVISION
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720