United Kingdom Atomic Energy Authority

RESEARCH GROUP

Report

A FORTRAN SUBROUTINE FOR UNCONSTRAINED MINIMIZATION, REQUIRING FIRST DERIVATIVES OF THE OBJECTIVE FUNCTION

M. J. D. POWELL

Theoretical Physics Division,

Atomic Energy Research Establishment,

Harwell, Berkshire.

1970

Price 4s. 6d. net from H. M. Stationery Office.

U.D.C.
681.14:517

AERE-R.6469

A FORTRAN SUBROUTINE FOR UNCONSTRAINED MINIMIZATION, REQUIRING FIRST DERIVATIVES OF THE OBJECTIVE FUNCTION

by

M. J. D. Powell

Abstract A Fortran subroutine is described and listed for calculating the least value of a function of several variables, $F(x_1,x_2,\ldots,x_n)$ say. The user must provide a subroutine to calculate $F(x_1,x_2,\ldots,x_n)$, and the first derivative vector of the objective function, for any $(x_1,x_2,\ldots,x_n)$. The method used is the new one proposed by Powell (1970), which has the advantage that convergence is guaranteed in theory, even if no good initial estimate of the required vector of variables is available. The convergence criterion is that the algorithm finishes when its search to reduce $F(x_1,x_2,\ldots,x_n)$ gives a point at which the magnitude of the first derivative is less than a prescribed positive tolerance. Numerical examples and the convergence theorems (Powell, 1970) suggest that the new subroutine compares favourably with the other general methods for solving unconstrained optimization problems.

Mathematics Branch,
Theoretical Physics Division,
Atomic Energy Research Establishment,
Harwell,
Berkshire.

July, 1970.

HL.70/4037 (C.13)

1. Introduction

Recently (Powell, 1970) I suggested a new algorithm for unconstrained minimization, and I proved that it has some nice super-linear convergence properties. But I omitted some details that are unimportant to the convergence theorems. Therefore this report provides a complete description of the new algorithm, and it includes a Fortran listing.

We let $F(\underset{\sim}{x})$ be the function whose least value is required, where $\underset{\sim}{x}$ stands for the vector of variables $(x_1,x_2,\ldots,x_n)$, and we let $g(\underset{\sim}{x})$ denote the gradient vector whose components are

$$g_i(\underset{\sim}{x}) = \partial F(\underset{\sim}{x})/\partial x_i, \quad i=1,2,\ldots,n. \tag{1}$$

To use the new algorithm it is necessary to provide a subroutine that calculates $F(\underset{\sim}{x})$ and $g_i(\underset{\sim}{x})$ $(i=1,2,\ldots,n)$ for any $\underset{\sim}{x}$.

The new algorithm is iterative, and it calculates a sequence of vectors $\underset{\sim}{x}^{(k)}$ $(k=1,2,\ldots)$, that is intended to converge to the point, $\underset{\sim}{x}^*$ say, at which $F(\underset{\sim}{x})$ is least. The k^{th} iteration of the algorithm uses $\underset{\sim}{x}^{(k)}$, the gradient

$$\underset{\sim}{g}^{(k)} = \underset{\sim}{g}(\underset{\sim}{x}^{(k)}) \tag{2}$$

say, some information that is accumulated from previous iterations, and just one evaluation of $F(\underset{\sim}{x})$ and $\underset{\sim}{g}(\underset{\sim}{x})$, to define $\underset{\sim}{x}^{(k+1)}$ and $\underset{\sim}{g}^{(k+1)}$. Also it revises the accumulated information.

This accumulated information is a step-bound $\Delta^{(k)}$, and three nxn matrices, namely $G^{(k)}$, $H^{(k)}$ and $D^{(k)}$. The value of $\Delta^{(1)}$ has to be set by the user of the subroutine, and all iterations force the condition

$$\| \underset{\sim}{x}^{(k+1)} - \underset{\sim}{x}^{(k)} \| \leq \Delta^{(k)}, \tag{3}$$

where throughout this report the vector norm is the Euclidean vector norm

$$\| \underset{\sim}{x} \| = \left[\sum_{i=1}^{n} x_i^2 \right]^{\frac{1}{2}}. \tag{4}$$

The matrices $G^{(k)}$ and $H^{(k)}$ are symmetric. $G^{(k)}$ is an estimate of the second derivative matrix of $F(\underset{\sim}{x})$ at $\underset{\sim}{x}^{(k)}$, and $H^{(k)}$ is the inverse of $G^{(k)}$. We require both these matrices in order to keep the number of operations of each iteration of the algorithm to within a multiple of n^2. $D^{(k)}$ is a matrix whose columns are orthogonal directions, that depend mainly on the directions of the vectors $\{\underset{\sim}{x}^{(k+1)} - \underset{\sim}{x}^{(k)}\}$, and it is used to ensure that $G^{(k)}$ is an adequate approximation to the second derivative matrix at $\underset{\sim}{x}^{(k)}$. Initially $G^{(k)}$, $H^{(k)}$ and $D^{(k)}$ are set to certain multiples of the nxn unit matrix.

The algorithm provides the inequalities

$$F(\underset{\sim}{x}^{(1)}) \geqslant F(\underset{\sim}{x}^{(2)}) \geqslant \ldots \geqslant F(\underset{\sim}{x}^{(k)}) \geqslant \ldots \tag{5}$$

It finishes as soon as the test

$$\| \underset{\sim}{g}^{(k+1)} \| \leqslant \varepsilon \tag{6}$$

is satisfied, where ε is a positive number whose value is assigned by the user of the subroutine.

Section 2 of this report states how to call the algorithm from a Fortran program, and the next five sections give details of the algorithm. Section 3 describes the instructions that precede the first iteration. Section 4 specifies the usual method for calculating the difference $\underset{\sim}{x}^{(k+1)} - \underset{\sim}{x}^{(k)}$. Section 5 defines the revision of the step-bound $\Delta^{(k)}$. Section 6 specifies the sequence of second derivative approximations $G^{(k)}$ $(k=2,3,\ldots)$, and also it gives the formula for calculating $H^{(k)}$. Section 7 describes the way in which the matrices $D^{(k)}$ are calculated and used. Finally Section 8 gives some numerical results, obtained by applying the algorithm to two test functions.

In fact there are two Fortran listings in the appendix to this report. The first one is intended to be understood, and the second one is intended to be useful. The main difference between them is that in the first listing two-dimensional arrays are used to hold the elements of the matrices $D^{(k)}$, $G^{(k)}$ and $H^{(k)}$, in order that it is straightforward to distinguish different matrix elements. But, because some Fortran compilers do not handle two-dimensional arrays efficiently, and to exploit the symmetry of $G^{(k)}$ and $H^{(k)}$, the second listing uses a single working space vector, namely W, to hold all the matrix elements. Therefore in practice the second listing is the better subroutine. In

the main text of this report we refer to the instructions of the first listing only, because the reader should have no difficulty in identifying the instructions of the second listing with the instructions of the first listing, due to the similarities in label numbers and comment cards.

As well as giving details of the new algorithm, this report contains some remarks that may be of general interest. For example in Section 6 the stability of any differences between $[H^{(k)}]^{-1}$ and $G^{(k)}$ due to computer rounding errors, is discussed. And in Section 7 it is proved that the role of $D^{(k)}$ gives a condition that is necessary for Powell's (1970) proof of the convergence of the second derivative approximations $G^{(k)}$.

2. The parameters of the Fortran subroutine

In the first listing of the appendix the name of the subroutine and its parameters are

SUBROUTINE MINFA(N,X,F,G,STEP,ACC,MAXFUN,IPRINT).

X and G are one-dimensional arrays, whose lengths must not be less than N, and the remaining parameters are all real or integer variables. The user must give values to N, X(I) (I=1,2,...,N), STEP, ACC, MAXFUN and IPRINT. The subroutine changes X(I) (I=1,2,...,N), and it calculates values for F and G(I) (I=1,2,...,N).

N is the number of variables of the objective function, and it must be in the range $2 \leqslant N \leqslant 50$. The upper bound is imposed by the dimensions of the arrays that are private to the subroutine, so it is straightforward to change it.

Initially X(I) (I=1,2,...,N) must be set to an estimate of the required vector of variables. The algorithm adjusts this vector, so that, when the subroutine finishes, it contains the best calculated values of the variables.

F will contain the least calculated value of $F(\underset{\sim}{x})$, corresponding to the final vector X(I) (I=1,2,...,N).

G(I) (I=1,2,...,N) will be set to the components of the gradient $\underset{\sim}{g}(\underset{\sim}{x})$, for the final vector X(I) (I=1,2,...,N).

STEP must be set to a positive number, that is the user's recommendation of the initial change to make to $\underset{\sim}{x}$ in the search for the least value of $F(\underset{\sim}{x})$. Indeed, following expression (3), the algorithm satisfies the inequality

$$\| \underset{\sim}{x}^{(2)} - \underset{\sim}{x}^{(1)} \| < \text{STEP}. \qquad (7)$$

About one tenth of the total expected change in $\underset{\sim}{x}$ is a moderate value for the parameter STEP.

ACC controls the accuracy of the calculation. The parameter ε of inequality (6) is set equal to ACC. Note that this convergence criterion is such that it is sensible to scale the variables $(x_1,x_2,\ldots,x_n)$ so that the components of the gradient vectors $\underset{\sim}{g}(\underset{\sim}{x})$ have similar magnitudes. There are other good reasons for trying to achieve this scaling.

The parameter MAXFUN provides an upper bound on the total number of evaluations of $F(\underset{\sim}{x})$ that are made by the algorithm. The subroutine provides an error return if the iteration number k attains the value MAXFUN, and it indicates this by printing the diagnostic message "MINFA has made MAXFUN calls of CALCFG". However usually the subroutine finishes because the convergence criterion (6) is satisfied. Experiments show that frequently the subroutine requires less than max [100, 10*N] evaluations of $F(\underset{\sim}{x})$, but MAXFUN should be set to a greater number, unless the total amount of calculation needs to be limited by a conservative value of MAXFUN.

IPRINT controls the amount of printed output from the subroutine. If IPRINT is zero there is no printing, except perhaps the diagnostic message of the last paragraph. Otherwise the best calculated values of $\underset{\sim}{x}$ and $F(\underset{\sim}{x})$ are printed after every |IPRINT| iterations, and, if IPRINT is positive, the gradient vector $\underset{\sim}{g}(\underset{\sim}{x})$ is printed also.

The subroutine, provided by the user, to calculate $F(\underset{\sim}{x})$ and $\underset{\sim}{g}(\underset{\sim}{x})$ must have the name

```
SUBROUTINE CALCFG (N,X,F,G),
```

where N,X,F and G correspond to the N,X,F and G of the parameter list of subroutine MINFA. Subroutine MINFA calls CALCFG once on every iteration, and it assigns the components of X. CALCFG must not change the values of N and X(I) (I=1,2,...,N), but it must set F to the function value $F(\underset{\sim}{x})$, and it must set the elements G(I) (I=1,2,...,N) to the components of the gradient $\underset{\sim}{g}(\underset{\sim}{x})$, where $\underset{\sim}{x}$ is the vector whose components are X(I) (I=1,2,...,N).

The second subroutine of the appendix has the name

```
SUBROUTINE VA06A(N,X,F,G,STEP,ACC,MAXFUN,IPRINT,W).
```

Its first eight parameters are identical to the parameters of subroutine MINFA, and the

last parameter W is a one-dimensional array, whose elements are used as working space. The length of this array must be at least $(2n^2 + 6n)$. Note that one difference between MINFA and VA06A is that VA06A does not use any private arrays, so it does not impose an artificial upper bound on N.

The array W is utilised in such a way that, when the execution of subroutine VA06A finishes, its first $\frac{1}{2}n(n+1)$ elements are estimates of the second derivatives $\partial^2 F(\underset{\sim}{x})/\partial x_i \, \partial x_j$ $(i=1,2,\ldots,n;\ j=i,\ i+1,\ldots,n)$, calculated at the point $\underset{\sim}{x}$, whose components are the final values of X(I) (I=1,2,...,N). To be specific the estimate of $\partial^2 F(\underset{\sim}{x})/\partial x_i \, \partial x_j$, $j \geq i$, is the element $W(\{n-\frac{1}{2}i\}\{i-1\}+j)$. Moreover the next $\frac{1}{2}n(n+1)$ locations of W contain the elements of the upper triangle of the inverse of the matrix of the second derivative approximation.

3. The initial operations of the algorithm

The description of the algorithm that is given in the next five sections refers frequently to the listing of subroutine MINFA, which is in the appendix. Indeed we do not comment on the operations whose purpose is obvious from the listing, and when we dwell on a detail, the corresponding line number of the listing is quoted.

In Section 2 we explained the parameter list of MINFA, so now we consider the dimension statement, given in lines 2 and 3. Here the vector X is reserved for the vector of variables that has given the least of all the calculated values of $F(\underset{\sim}{x})$, and G is reserved for the corresponding value of $\underset{\sim}{g}(\underset{\sim}{x})$. The vectors XA, GA, WA, WB and WC are used for scratch working space. At the beginning of the k^{th} iteration (k=1,2,...), the matrices, GG, H and DD contain the elements of the matrices $G^{(k)}$, $H^{(k)}$ and $D^{(k)}$, which are mentioned in Section 1.

The real variable DSS, whose initial value is set in line 8, is equal to the square of the bound $\Delta^{(k)}$, that occurs in inequality (3). The integer variable MAXC (see line 9) is used to count the number of calls of subroutine CALCFG. The integer variable ITSPEC (see line 10) is used to distinguish every third iteration, because when $k=3\ell+2$, ℓ an integer, the purpose of the k^{th} iteration is just to improve the approximation $G^{(k)}$. We call these iterations "special iterations", and we discuss them in detail in Section 7. The integer variable IPTEST (see line 11) is adjusted so that the best values of $\underset{\sim}{x}$ and $F(\underset{\sim}{x})$ are printed after every |IPRINT| iterations.

The initial elements of the matrices $G^{(k)}$, $H^{(k)}$ and $D^{(k)}$ are set by instructions 15-28. We note that $G^{(1)}$ is the matrix

$$G^{(1)} = \frac{0.01 \, \| g^{(1)} \|}{\Delta^{(1)}} \, I, \tag{8}$$

$H^{(1)}$ is equal to $[G^{(1)}]^{-1}$, and $D^{(1)}$ is the nxn unit matrix, namely I. A consequence of the choice (8) is that the quadratic approximation

$$F(\underset{\sim}{x}) \approx F(\underset{\sim}{x}^{(1)}) + (\underset{\sim}{g}^{(1)}, \underset{\sim}{x}-\underset{\sim}{x}^{(1)}) + \tfrac{1}{2}(\underset{\sim}{x}-\underset{\sim}{x}^{(1)}, \, G^{(1)}\{\underset{\sim}{x}-\underset{\sim}{x}^{(1)}\}) \tag{9}$$

suggests that the least value of $F(\underset{\sim}{x})$ is at the point

$$\underset{\sim}{x} = \underset{\sim}{x}^{(1)} - 100 \, \Delta^{(1)} \, \underset{\sim}{g}^{(1)} / \, \| g^{(1)} \| . \tag{10}$$

Without further information we cannot do more than try changing $\underset{\sim}{x}^{(1)}$ by a positive multiple of $-\underset{\sim}{g}^{(1)}$, so the choice (8) is adequate in the sense that the resultant approximation (9) supports a step down the steepest descent direction. But the length of the correction of equation (10) is one hundred times the user's recommendation of the initial step-length. The factor 100 is present because it is preferable if initially the estimate of the second derivative matrix is too small, rather than too large. $D^{(1)}$ is set to the unit matrix because it must be set to some orthogonal matrix, and there is no good reason for making any other choice.

The instructions for a general iteration of the algorithm begin at line 35 of the Fortran listing.

4. Adjusting the vector of variables

First a general iteration of the algorithm tests for convergence (see line 38 of the Fortran listing of MINFA), and then it provides any printing that is required by the value of IPRINT. Next it predicts a correction $\underset{\sim}{\delta}^{(k)}$ to apply to the current best vector of variables, namely $\underset{\sim}{x}^{(k)}$, and it is the purpose of this section to describe the calculation of $\underset{\sim}{\delta}^{(k)}$ for the iterations that are not "special iterations", because the special iterations are treated in Section 7. Our definition of $\underset{\sim}{\delta}^{(k)}$ depends only on the vector $\underset{\sim}{g}^{(k)}$, the bound $\Delta^{(k)}$, and the matrices $G^{(k)}$ and $H^{(k)}$. When $\underset{\sim}{\delta}^{(k)}$ has been calculated, the subroutine calculates $F(\underset{\sim}{x}^{(k)} + \underset{\sim}{\delta}^{(k)})$ and $\underset{\sim}{g}(\underset{\sim}{x}^{(k)} + \underset{\sim}{\delta}^{(k)})$, and for the next iteration the vector of variables is defined by the formula

$$\underset{\sim}{x}^{(k+1)} = \begin{cases} \underset{\sim}{x}^{(k)}, & F(\underset{\sim}{x}^{(k)}+\underset{\sim}{\delta}^{(k)}) \geq F(\underset{\sim}{x}^{(k)}) \\ \underset{\sim}{x}^{(k)}+\underset{\sim}{\delta}^{(k)}, & F(\underset{\sim}{x}^{(k)}+\underset{\sim}{\delta}^{(k)}) < F(\underset{\sim}{x}^{(k)}). \end{cases} \tag{11}$$

Thus inequality (5) is obtained, using just one call of CALCFG on each iteration.

The definition of $\underset{\sim}{\delta}^{(k)}$ for the ordinary iterations is like the one used by Powell (1968a), and it is based on the quadratic approximation

$$\begin{aligned} F(\underset{\sim}{x}^{(k)}+\underset{\sim}{\delta}) &\approx \Phi^{(k)}(\underset{\sim}{\delta}) \\ &= F(\underset{\sim}{x}^{(k)}) + (\underset{\sim}{g}^{(k)}, \underset{\sim}{\delta}) + \tfrac{1}{2}(\underset{\sim}{\delta}, G^{(k)}\underset{\sim}{\delta}). \end{aligned} \tag{12}$$

Moreover following inequality (3) we require $\underset{\sim}{\delta}^{(k)}$ to satisfy the condition

$$\| \underset{\sim}{\delta}^{(k)} \| \leq \Delta^{(k)}. \tag{13}$$

If it happens that the quadratic approximation (12) decreases monotonically along its steepest descent direction (at $\underset{\sim}{\delta}=\underset{\sim}{0}$) for the distance $\Delta^{(k)}$, then $\underset{\sim}{\delta}^{(k)}$ is set to the multiple of the steepest descent vector

$$\underset{\sim}{\delta}^{(k)} = -\Delta^{(k)}\underset{\sim}{g}^{(k)} / \| \underset{\sim}{g}^{(k)} \| . \tag{14}$$

The condition that $\Phi^{(k)}(-\lambda\underset{\sim}{g}^{(k)} / \|\underset{\sim}{g}^{(k)}\|)$ decreases monotonically for $0 \leq \lambda \leq \Delta^{(k)}$ is the inequality

$$(\underset{\sim}{g}^{(k)}, G^{(k)}\underset{\sim}{g}^{(k)})\ \Delta^{(k)} \leq \| \underset{\sim}{g}^{(k)} \|^3, \tag{15}$$

which is tested by instruction 103 of the Fortran listing. However, if the test (15) fails, then, instead of formula (14), the correction $\underset{\sim}{\delta}^{(k)}$ is defined in a way that depends on the position of the stationary point of $\Phi^{(k)}(\underset{\sim}{\delta})$, which is the point

$$\underset{\sim}{v}^{(k)} = -H^{(k)}\underset{\sim}{g}^{(k)}, \tag{16}$$

because $H^{(k)}$ is the inverse of $G^{(k)}$. Also the definition of $\underset{\sim}{\delta}^{(k)}$ depends on the best steepest descent correction for the approximation (12), which is the vector

$$\underset{\sim}{s}^{(k)} = -\| \underset{\sim}{g}^{(k)} \|^2 \underset{\sim}{g}^{(k)} / (\underset{\sim}{g}^{(k)}, G^{(k)} \underset{\sim}{g}^{(k)}), \tag{17}$$

whose length is less than $\Delta^{(k)}$, due to the failure of inequality (15).

Because we want the algorithm to have a fast final rate of convergence, we frequently let $\underset{\sim}{\delta}^{(k)}$ equal the vector (16). But $\| \underset{\sim}{y}^{(k)} \|$ may exceed $\Delta^{(k)}$, and moreover if $G^{(k)}$ is not positive definite then it can happen that $\Phi^{(k)}(\underset{\sim}{y}^{(k)}) > F(\underset{\sim}{x}^{(k)})$, so these are two cases when $\underset{\sim}{\delta}^{(k)} \neq \underset{\sim}{y}^{(k)}$. Therefore, when formula (14) is not used, the strategy of the algorithm is based on the variation of $\Phi^{(k)}(\underset{\sim}{\delta})$ for vectors $\underset{\sim}{\delta}$ that are on the straight line through the points (16) and (17). Indeed $\underset{\sim}{\delta}^{(k)}$ has the form

$$\underset{\sim}{\delta}^{(k)} = \underset{\sim}{s}^{(k)} + \theta^* \{ \underset{\sim}{y}^{(k)} - \underset{\sim}{s}^{(k)} \}, \tag{18}$$

where the value of θ^* depends on the function

$$\phi(\theta) = \Phi^{(k)}(\underset{\sim}{s}^{(k)} + \theta \{ \underset{\sim}{y}^{(k)} - \underset{\sim}{s}^{(k)} \}). \tag{19}$$

We calculate the value of θ^* so that $\phi(\theta^*)$ is small subject to condition (13). Therefore we let the sign of θ^* be opposite to the sign of $\phi'(0)$, and from equations (12), (16), (17) and (19) we find that this derivative has the value

$$\begin{aligned} \phi'(0) &= (\underset{\sim}{g}^{(k)} + G^{(k)} \underset{\sim}{s}^{(k)}, \underset{\sim}{v}^{(k)} - \underset{\sim}{s}^{(k)}) \\ &= (\underset{\sim}{g}^{(k)}, \underset{\sim}{v}^{(k)} - \underset{\sim}{s}^{(k)}). \end{aligned} \tag{20}$$

Moreover, because the quadratic function $\phi(\theta)$ is stationary at $\theta=1$, we note that $\phi(\theta)$ is convex if and only if the value of expression (20) is negative or zero.

If $\phi(\theta)$ is concave, then $\phi'(0) > 0$, and we let θ^* be the most negative number subject to the condition (13), so θ^* is the smaller root of the quadratic equation

$$\| s^{(k)} + \theta \{ v^{(k)} - s^{(k)} \} \|^2 = \{ \Delta^{(k)} \}^2. \tag{21}$$

However if $\phi(\theta)$ is convex, then it is sensible to define $\theta^* = 1$, unless this choice conflicts with condition (13), in which case θ^* is made equal to the larger root of

equation (21). Now it is fortunate that the sign of expression (20) is always opposite to the sign of the coefficient of θ in the quadratic equation (21), because this fact implies that, whenever the test (15) fails, we define θ^* to be the quantity

$$\theta^* = \min [1, \text{THETA}], \tag{22}$$

where THETA is the root of equation (21) with smaller modulus. Indeed THETA has the value

$$\text{THETA} = \frac{[\{\Delta^{(k)}\}^2 - \|\underset{\sim}{s}^{(k)}\|^2] \text{ sign } (\underset{\sim}{s}^{(k)}, \underset{\sim}{w}^{(k)})}{|(s^{(k)}, w^{(k)})| + [(\underset{\sim}{s}^{(k)}, \underset{\sim}{w}^{(k)})^2 + \|\underset{\sim}{w}^{(k)}\|^2(\{\Delta^{(k)}\}^2 - \|\underset{\sim}{s}^{(k)}\|^2)]^{\frac{1}{2}}}, \tag{23}$$

where $\underset{\sim}{w}^{(k)} = \underset{\sim}{y}^{(k)} - \underset{\sim}{s}^{(k)}$. The components of $\underset{\sim}{s}^{(k)}$ and $\underset{\sim}{w}^{(k)}$ are set by instructions 115 and 116 of the Fortran listing, the value of THETA is set by instruction 121, and when instruction 127 is reached, the components of $\underset{\sim}{\delta}^{(k)}$ are present in the working space array WA.

To prove a convergence theorem (Powell, 1970), we have to note that $\underset{\sim}{\delta}^{(k)}$ always has the property that the gradient of the function $\Phi^{(k)}(\underset{\sim}{\delta})$, at the point $\underset{\sim}{\delta} = \underset{\sim}{\delta}^{(k)}$, along the direction $\underset{\sim}{\delta}^{(k)}$, is always negative or zero, which is the condition

$$(\underset{\sim}{g}^{(k)} + G^{(k)} \underset{\sim}{\delta}^{(k)}, \underset{\sim}{\delta}^{(k)}) \leq 0 . \tag{24}$$

It is easy to verify this statement when formula (14) is used, because in this case the condition (15) is satisfied. Therefore we now show that inequality (24) is valid when $\underset{\sim}{\delta}^{(k)}$ is defined by expressions (18), (22) and (23).

The function

$$\sigma(\theta) = \tfrac{1}{2}(\underset{\sim}{g}^{(k)} + G^{(k)}[\underset{\sim}{s}^{(k)} + \theta\{\underset{\sim}{y}^{(k)} - \underset{\sim}{s}^{(k)}\}], \underset{\sim}{s}^{(k)} + \theta\{\underset{\sim}{y}^{(k)} - \underset{\sim}{s}^{(k)}\}) \tag{25}$$

is a quadratic function that is zero at $\theta = 0$ and at $\theta = 1$. Therefore inequality (24) holds if $\sigma(\theta)$ is convex and $0 \leq \theta^* \leq 1$. Also inequality (24) holds if $\sigma(\theta)$ is concave and $\theta^* \leq 0$. Now the coefficients of θ^2 in expressions (19) and (25) are the same, so $\sigma(\theta)$ is

convex if and only if the function $\phi(\theta)$ is convex. Therefore the definition of θ^* implies that inequality (24) is true.

The Fortran instructions that are applied immediately after $\underset{\sim}{\delta}^{(k)}$ is defined are discussed in Section 7, because they calculate the matrix $D^{(k+1)}$ for the next iteration. The values of $F(\underset{\sim}{x}^{(k)} + \underset{\sim}{\delta}^{(k)})$ and $\underset{\sim}{g}(\underset{\sim}{x}^{(k)} + \underset{\sim}{\delta}^{(k)})$ are obtained by instruction 161 of the Fortran listing, and the definition (11) is realised by instructions 192 to 196.

5. The calculation of the step-bound

This section specifies the calculation of $\Delta^{(k)}$ $(k=1,2,\ldots)$, which is the bound on $\| \underset{\sim}{\delta}^{(k)} \|$ of inequalities (3) and (13). The value of $\Delta^{(1)}$ is assigned by the user of the subroutine, and the value of $\Delta^{(k+1)}$ $(k=1,2,\ldots)$ depends on $\| \underset{\sim}{\delta}^{(k)} \|$ and on the success of the k^{th} iteration, except that every "special iteration" sets $\Delta^{(k+1)} = \Delta^{(k)}$. Instruction 178 of the Fortran listing tests whether the algorithm is applying an ordinary iteration, and, if it is, then the block of instructions on lines 179 to 186 defines the step-bound for the next iteration. In fact the algorithm calculates the value of $\{\Delta^{(k+1)}\}^2$, which is the real variable DSS, so we note that, when instruction 188 is reached, $\Delta^{(k+1)}$ has been set equal to one of the three numbers $\frac{1}{2}\| \underset{\sim}{\delta}^{(k)} \|$, $\| \underset{\sim}{\delta}^{(k)} \|$ and $2\| \underset{\sim}{\delta}^{(k)} \|$. We now explain the actual choice between these three numbers.

The choice that is made depends, of course, on the purpose of the step-bound, which is to make $\| \underset{\sim}{\delta}^{(k)} \|$ so small that the approximation (12) is adequate for $\underset{\sim}{\delta} = \underset{\sim}{\delta}^{(k)}$. However we must be careful not to make $\| \underset{\sim}{\delta}^{(k)} \|$ too small, because then a large number of iterations would be needed to change the vector of variables by a moderate amount. Therefore the three available values of $\Delta^{(k+1)}$ can let the step-length of the $(k+1)^{st}$ iteration be smaller than, greater than, or the same as the step-length of the k^{th} iteration.

To decide whether $\| \underset{\sim}{\delta}^{(k)} \|$ is too large, we compare the actual reduction in the objective function, namely $F(\underset{\sim}{x}^{(k)} + \underset{\sim}{\delta}^{(k)}) - F(\underset{\sim}{x}^{(k)})$, with the predicted reduction $\Phi^{(k)}(\underset{\sim}{\delta}^{(k)}) - F(\underset{\sim}{x}^{(k)})$. Indeed instruction 179 of the Fortran listing tests the inequality

$$F(\underset{\sim}{x}^{(k)}+\underset{\sim}{\delta}^{(k)})-F(\underset{\sim}{x}^{(k)} \leq 0.1\,\{\Phi^{(k)}(\underset{\sim}{\delta}^{(k)}) - F(\underset{\sim}{x}^{(k)})\}, \qquad (26)$$

and if this condition fails the algorithm sets $\Delta^{(k+1)} = \frac{1}{2}\| \underset{\sim}{\delta}^{(k)} \|$, in order that a smaller correction vector will be used by the next iteration. However, if inequality (26) is satisfied, we judge that the current step-length is not too large, so the value

of $\Delta^{(k+1)}$ is set either to $\| \underset{\sim}{\delta}^{(k)} \|$ or to $2\| \underset{\sim}{\delta}^{(k)} \|$. The factor 0.1 of the right-hand side of expression (26) is also used by Powell (1968a), and the reason for this particular choice is only that numerical experiments suggest that it is adequate.

When the inequality (26) holds, we have to guess whether it is preferable to let $\| \underset{\sim}{\delta}^{(k+1)} \|$ exceed $\| \underset{\sim}{\delta}^{(k)} \|$. One calculated number that helps this guess is the value of the scalar product $\{\underset{\sim}{g}(\underset{\sim}{x}^{(k)} + \underset{\sim}{\delta}^{(k)}), \underset{\sim}{\delta}^{(k)}\}$, because, if it is negative, then we know that $F(\underset{\sim}{x}^{(k)} + \lambda \underset{\sim}{\delta}^{(k)}) < F(\underset{\sim}{x}^{(k)} + \underset{\sim}{\delta}^{(k)})$ for some $\lambda > 1$, in which case it seems that a larger step length would have been preferable. Further an estimate of the optimal value of λ can be obtained by supposing that the function

$$s(\lambda) = \{\underset{\sim}{g}(\underset{\sim}{x}^{(k)} + \lambda\underset{\sim}{\delta}^{(k)}), \underset{\sim}{\delta}^{(k)}\} \tag{27}$$

depends linearly on λ. Indeed this approach predicts that $F(\underset{\sim}{x}^{(k)} + \lambda\underset{\sim}{\delta}^{(k)})$ is least when λ is the number

$$\lambda^* = \begin{cases} \infty, & s(1) \leqslant s(0) \\ s(0)/\{s(0)-s(1)\}, & s(1) > s(0). \end{cases} \tag{28}$$

Therefore the test on line 185 of the Fortran listing leads to the value $\Delta^{(k+1)} = 2\| \underset{\sim}{\delta}^{(k)} \|$ if λ^* is greater than or equal to two, because in this case it seems to be worthwhile to double the step-length.

When inequality (26) holds, and when $\lambda^* < 2$, the algorithm may set $\Delta^{(k+1)} = \| \underset{\sim}{\delta}^{(k)} \|$ or $\Delta^{(k+1)} = 2\| \underset{\sim}{\delta}^{(k)} \|$. We sometimes prefer the larger value when $\lambda^* < 2$, because, when the quadratic approximation (12) is exact, it can happen that λ^* is close to one and it is well worthwhile to double the step-bound for the next iteration. Therefore we want to estimate the goodness of the approximation (12). We do this by comparing the predicted gradient at $\underset{\sim}{x}^{(k)} + \underset{\sim}{\delta}^{(k)}$, namely $\underset{\sim}{g}(\underset{\sim}{x}^{(k)}) + G^{(k)}\underset{\sim}{\delta}^{(k)}$, with the actual gradient $\underset{\sim}{g}(\underset{\sim}{x}^{(k)} + \underset{\sim}{\delta}^{(k)})$, and specifically line 184 of the listing tests the inequality

$$\| \underset{\sim}{g}(\underset{\sim}{x}^{(k)} + \underset{\sim}{\delta}^{(k)}) - \underset{\sim}{g}^{(k)} - G^{(k)}\underset{\sim}{\delta}^{(k)} \|^2 \leqslant \tfrac{1}{4} \| \underset{\sim}{g}^{(k)} \|^2. \tag{29}$$

If it is satisfied then $\Delta^{(k+1)} = 2\| \underset{\sim}{\delta}^{(k)} \|$, but if it fails, if $\lambda^* < 2$, and if inequality (26) holds, then $\Delta^{(k+1)} = \| \underset{\sim}{\delta}^{(k)} \|$.

6. The calculation of $G^{(k)}$ and $H^{(k)}$

The calculation of the initial second derivative approximation, namely $G^{(1)}$, has been described in Section 3. Now we specify the later second derivative approximations $G^{(k)}$ $(k=2,3,\ldots)$, and also we give the formulae that define the matrices $H^{(k)} = [G^{(k)}]^{-1}$. Moreover the stability of these formulae is discussed, from the point of view of the cumulative effect of computer rounding errors.

The calculation of $G^{(k+1)}$ and $H^{(k+1)}$ is made by instructions 199 to 243 of the Fortran listing. It depends on the matrices $G^{(k)}$ and $H^{(k)}$, and on the vectors $\underset{\sim}{\delta}^{(k)}$ and

$$\underset{\sim}{\gamma}^{(k)} = \underset{\sim}{g}(\underset{\sim}{x}^{(k)} + \underset{\sim}{\delta}^{(k)}) - \underset{\sim}{g}^{(k)}, \tag{30}$$

and it is based on the equation

$$G^{*} = G + \frac{(\underset{\sim}{\gamma}-G\underset{\sim}{\delta})\,\underset{\sim}{\delta}^{T} + \underset{\sim}{\delta}(\underset{\sim}{\gamma}-G\underset{\sim}{\delta})^{T}}{\|\underset{\sim}{\delta}\|^{2}} - \frac{\underset{\sim}{\delta}\underset{\sim}{\delta}^{T}(\underset{\sim}{\gamma}-G\underset{\sim}{\delta},\ \underset{\sim}{\delta})}{\|\underset{\sim}{\delta}\|^{4}}, \tag{31}$$

which is proposed by Powell (1970). Usually we let $\underset{\sim}{\delta} = \underset{\sim}{\delta}^{(k)}$, $\underset{\sim}{\gamma} = \underset{\sim}{\gamma}^{(k)}$, $G = G^{(k)}$ and $G^{(k+1)} = G^{*}$, because then we obtain the identity

$$G^{(k+1)}\,\underset{\sim}{\delta}^{(k)} = \underset{\sim}{\gamma}^{(k)}, \tag{32}$$

which is worthwhile because, in the case that $F(\underset{\sim}{x})$ is exactly quadratic, then the true second derivative matrix of $F(\underset{\sim}{x})$, $\overline{G}$ say, satisfies the equation $\overline{G}\underset{\sim}{\delta}^{(k)} = \underset{\sim}{\gamma}^{(k)}$. Moreover note that, unlike other formulae for revising second derivative approximations, for example see Broyden (1970), Davidon (1959) and Powell (1969), the formula (31) does not involve divisions by scalar products of different vectors.

However the matrix $G^{(k+1)}$ is not defined directly by equation (31) if this matrix is exactly or nearly singular, because in this case there are liable to be serious errors in the definition of $H^{(k+1)}$. Therefore the Fortran listing uses a method that guarantees the inequality

$$| \det G^{(k+1)} | \geq 0.1 \, |\det G^{(k)}| \, . \tag{33}$$

Later in this section we describe how this condition is maintained, and we will find that we have to depart from the usual definition of $G^{(k+1)}$ when the tests made on lines 216 and 217 of the Fortran listing lead to the statement on line 219.

To revise the inverse of the second derivative approximation we use the fact that if $H = G^{-1}$, and if G^* is defined by equation (31), then the inverse of G^* is the matrix

$$H^* = H - \{\underset{\sim}{\eta}\underset{\sim}{\eta}^T \, (\underset{\sim}{\delta},H\underset{\sim}{\delta}) - [\underset{\sim}{\eta}\underset{\sim}{\delta}^T H + H\underset{\sim}{\delta}\underset{\sim}{\eta}^T] \, (\underset{\sim}{\delta},H\underset{\sim}{\gamma})$$

$$+ H\underset{\sim}{\delta}\underset{\sim}{\delta}^T H \, (\underset{\sim}{\eta},\underset{\sim}{\gamma})\} / \{(\underset{\sim}{\eta},\underset{\sim}{\gamma})(\underset{\sim}{\delta},H\underset{\sim}{\delta}) - (\underset{\sim}{\delta},H\underset{\sim}{\gamma})^2\}, \tag{34}$$

where $\underset{\sim}{\eta}$ is the vector

$$\underset{\sim}{\eta} = H\underset{\sim}{\gamma} - \underset{\sim}{\delta}. \tag{35}$$

At instruction 199 of the Fortran listing the vectors $\underset{\sim}{\delta}$, $\underset{\sim}{\gamma} - G\underset{\sim}{\delta}$ and $\underset{\sim}{\gamma}$ are present in the working space arrays WA, WB and WC, and at instruction 215 the arrays XA and GA contain the vectors $\underset{\sim}{\eta}$ and $H\underset{\sim}{\delta}$. Then in the case that instruction 216 or 217 branches to line 230 of the Fortran listing, it is straightforward to verify that the matrices $G^{(k+1)}$ and $H^{(k+1)}$ are defined by formulae (31) and (34).

Of course the purpose of the conditional instruction on line 217 is to test whether the usual definition of $G^{(k+1)}$ satisfies inequality (33). We now prove a theorem to show that the conditional instruction is correct.

Theorem 1 If the matrix G is symmetric and non-singular, then the determinant of the matrix

$$G^*(\theta) = G + \theta \, \frac{\underset{\sim}{\mu}\underset{\sim}{\delta}^T + \underset{\sim}{\delta}\underset{\sim}{\mu}^T}{\| \underset{\sim}{\delta}\|^2} - \theta^2 \, \frac{\underset{\sim}{\delta}\underset{\sim}{\delta}^T(\underset{\sim}{\mu},\underset{\sim}{\delta})}{\| \underset{\sim}{\delta}\|^4} \tag{36}$$

is equal to the expression

$$\det\{G^*(\theta)\} = \det G \{1 + 2\theta \frac{(\underset{\sim}{\delta}, H\underset{\sim}{\mu})}{\| \underset{\sim}{\delta} \|^2}$$

$$+ \theta^2 \quad \frac{(\underset{\sim}{\delta}, H\underset{\sim}{\mu})^2 - (\underset{\sim}{\delta}, H\underset{\sim}{\delta})(\underset{\sim}{\mu}, H\underset{\sim}{\mu}) - (\underset{\sim}{\delta}, H\underset{\sim}{\delta})(\underset{\sim}{\delta}, \underset{\sim}{\mu})}{\| \underset{\sim}{\delta} \|^4} \}, \tag{37}$$

where H is the matrix G^{-1}.

Proof First we treat the special case when $\underset{\sim}{\mu}$ is a multiple of $\underset{\sim}{\delta}$, say $\underset{\sim}{\mu}=\lambda\underset{\sim}{\delta}$. Then $G^*(\theta)$ is the matrix

$$G^*(\theta) = G + \{ \frac{2\lambda\theta}{\| \underset{\sim}{\delta} \|^2} - \frac{\lambda\theta^2}{\| \underset{\sim}{\delta} \|^2} \} \underset{\sim}{\delta}\underset{\sim}{\delta}^T , \tag{38}$$

and its determinant has the value

$$\det G^*(\theta) = \det G \det \left[I + \{ \frac{2\lambda\theta}{\| \underset{\sim}{\delta} \|^2} - \frac{\lambda\theta^2}{\| \underset{\sim}{\delta} \|^2} \} H\underset{\sim}{\delta}\underset{\sim}{\delta}^T \right]. \tag{39}$$

Therefore, as stated in Section 1.1 of Householder (1964), we have the expression

$$\det\{G^*(\theta)\} = \det G \{1 + \left[\frac{2\lambda\theta}{\| \underset{\sim}{\delta} \|^2} - \frac{\lambda\theta^2}{\| \underset{\sim}{\delta} \|^2} \right] (\underset{\sim}{\delta}, H\underset{\sim}{\delta})\}, \tag{40}$$

which justifies equation (37).

For the usual case, when $\underset{\sim}{\mu}$ and $\underset{\sim}{\delta}$ are independent, we let Ω be an nxn orthogonal matrix, whose first column is $\underset{\sim}{\delta}/\| \underset{\sim}{\delta} \|$, and whose second column is in the space spanned by $\underset{\sim}{\mu}$ and $\underset{\sim}{\delta}$. Then the definition (36) shows that the elements of the matrix

$$G^+(\theta) = \Omega^T G^*(\theta) \Omega \tag{41}$$

are independent of θ, except for the elements $G^+_{11}(\theta)$, $G^+_{12}(\theta)$ and $G^+_{21}(\theta)$. Indeed $G^+_{11}(\theta)$ is a quadratic function of θ and $G^+_{12}(\theta)$ $\{=G^+_{21}(\theta)\}$ is a linear function of θ. Therefore the determinant of $G^+(\theta)$ is a quadratic function of θ, so equation (41) shows that $\det\{G^*(\theta)\}$ is also a quadratic function of θ.

We now look for the zeros of the quadratic function $\det\{G^*(\theta)\}$, and if necessary we allow θ to be complex. This function is zero if and only if there exists a non-zero vector $\underset{\sim}{\zeta}$ such that the equation

$$G^*(\theta)\underset{\sim}{\zeta} = \underset{\sim}{0} \tag{42}$$

holds. In this case equation (36) shows that the vector $G\underset{\sim}{\zeta}$ (which is non-zero because G is non-singular) is a linear combination of $\underset{\sim}{\mu}$ and $\underset{\sim}{\delta}$. Therefore there exist numbers α and β such that $\underset{\sim}{\zeta}$ is the vector

$$\underset{\sim}{\zeta} = \alpha H\underset{\sim}{\mu} + \beta H\underset{\sim}{\delta}, \tag{43}$$

and equations (36) and (42) imply that α and β satisfy the equations

$$\alpha + \frac{\theta}{\| \underset{\sim}{\delta} \|^2} \{ \alpha(\underset{\sim}{\delta}, H\underset{\sim}{\mu}) + \beta(\underset{\sim}{\delta}, H\underset{\sim}{\delta}) \} = 0 \tag{44}$$

and

$$\beta + \frac{\theta}{\| \underset{\sim}{\delta} \|^2} \{\alpha(\underset{\sim}{\mu}, H\underset{\sim}{\mu}) + \beta(\underset{\sim}{\mu}, H\underset{\sim}{\delta})\}$$

$$- \frac{\theta^2(\underset{\sim}{\mu}, \underset{\sim}{\delta})}{\| \underset{\sim}{\delta} \|^4} \{\alpha(\underset{\sim}{\delta}, H\underset{\sim}{\mu}) + \beta(\underset{\sim}{\delta}, H\underset{\sim}{\delta})\} = 0 . \tag{45}$$

Moreover it follows from equation (36) that the matrix $G^*(\theta)$ is singular if and only if θ is such that the equations (44) and (45) have a non-zero solution (α,β).

Now equations (44) and (45) imply the relation

$$\beta + \frac{\theta}{\| \underset{\sim}{\delta} \|^2} \{\alpha(\underset{\sim}{\mu}, H\underset{\sim}{\mu}) + \beta(\underset{\sim}{\mu}, H\underset{\sim}{\delta})\} + \frac{\alpha\theta(\underset{\sim}{\mu}, \underset{\sim}{\delta})}{\| \underset{\sim}{\delta} \|^2} = 0, \tag{46}$$

so the matrix $G^*(\theta)$ is singular if and only if θ satisfies the quadratic equation

$$\{1 + \frac{\theta}{\| \underset{\sim}{\delta} \|^2} (\underset{\sim}{\delta}, H\underset{\sim}{\mu})\}^2 = \frac{\theta^2}{\| \underset{\sim}{\delta} \|^4} (\underset{\sim}{\delta}, H\underset{\sim}{\delta}) \{(\underset{\sim}{\mu}, H\underset{\sim}{\mu}) + (\underset{\sim}{\mu}, \underset{\sim}{\delta})\} . \tag{47}$$

Therefore Theorem 1 must hold in the case that this quadratic equation has two separate roots, which happens if and only if the right-hand side of equation (47) is different from zero.

In the case that equation (47) does not have two separate roots, we find a vector, $\underset{\sim}{p}$ say, such that $(\underset{\sim}{p},H\underset{\sim}{p}) \neq 0$, and we consider replacing $\underset{\sim}{\delta}$ by $\underset{\sim}{\delta} + \varepsilon_1\underset{\sim}{p}$ and $\underset{\sim}{\mu}$ by $\underset{\sim}{\mu} + \varepsilon_2\,\underset{\sim}{p}$, where ε_1 and ε_2 are very small in modulus. We then regard the right-hand side of equation (47) as a function of ε_1 and ε_2, and we note that arbitrarily small values of these parameters cause the equation to have separate roots, in which case Theorem 1 holds. Now the determinant of a matrix is a continuous function of its coefficients, so it follows that equation (37) holds in the case that we are considering. The proof of Theorem 1 is now complete.

One important corollary of Theorem 1 is that the determinant of the matrix (31) is the expression

$$\frac{\det G}{\|\underset{\sim}{\delta}\|^4}\left\{\left[\|\underset{\sim}{\delta}\|^2 + (\underset{\sim}{\delta},H\underset{\sim}{\mu})\right]^2 - (\underset{\sim}{\delta},H\underset{\sim}{\delta})(\underset{\sim}{\mu},\underset{\sim}{\delta}+H\underset{\sim}{\mu})\right\}$$

$$= \frac{\det G}{\|\underset{\sim}{\delta}\|^4}\left\{(\underset{\sim}{\delta},H\underset{\sim}{\gamma})^2 - (\underset{\sim}{\delta},H\underset{\sim}{\delta})(\underset{\sim}{\gamma} - G\underset{\sim}{\delta},\ H\underset{\sim}{\gamma})\right\}$$

$$= \det G\left\{(\underset{\sim}{\delta},H\underset{\sim}{\gamma})^2 - (\underset{\sim}{\delta},H\underset{\sim}{\delta})(\underset{\sim}{\eta},\underset{\sim}{\gamma})\right\} / \|\underset{\sim}{\delta}\|^4, \qquad (48)$$

where $\underset{\sim}{\eta}$ is the vector (35). Therefore instruction 217 of the Fortran listing tests the inequality

$$\left|(\underset{\sim}{\delta},H\underset{\sim}{\gamma})^2 - (\underset{\sim}{\delta},H\underset{\sim}{\delta})(\underset{\sim}{\eta},\underset{\sim}{\gamma})\right| < 0.1\ \|\underset{\sim}{\delta}\|^4, \qquad (49)$$

because if it holds a special formula is needed to define $G^{(k+1)}$, in order to satisfy inequality (33).

In fact if inequality (49) holds, we let $G^{(k+1)}$ be the matrix $G^*(\theta)$ of equation (36), where $G = G^{(k)}$, $\underset{\sim}{\delta} = \underset{\sim}{\delta}^{(k)}$, $\underset{\sim}{\mu} = \underset{\sim}{\gamma}^{(k)} - G^{(k)}\underset{\sim}{\delta}^{(k)}$, and where the value of θ, $\theta^{(k)}$ say, is the number closest to one such that the equation

$$\det G^{(k+1)} = 0.1 \det G^{(k)} \qquad (50)$$

is satisfied. This choice is a good one because, in the case that $F(\underset{\sim}{x})$ is exactly a quadratic function, with second derivative matrix $\overline{G}$, it can be shown that the error in $G^{(k+1)}$ is related to the error in $G^{(k)}$ by the equation

$$\left(G^{(k+1)} - \overline{G} \right) = \left(I - \theta^{(k)} \frac{\underset{\sim}{\delta}\underset{\sim}{\delta}^T}{\| \underset{\sim}{\delta} \|^2} \right) \left(G^{(k)} - \overline{G} \right) \left(I - \theta^{(k)} \frac{\underset{\sim}{\delta}\underset{\sim}{\delta}^T}{\| \underset{\sim}{\delta} \|^2} \right). \qquad (51)$$

In Theorem 2 we prove that $| 1-\theta^{(k)} | \leq \sqrt{2/11}$, so our choice usually improves the accuracy of the second derivative approximation.

We now describe the calculation of $\theta^{(k)}$. Theorem 1 and equation (50) show that the required value of θ satisfies the quadratic equation

$$0.9 \| \underset{\sim}{\delta} \|^4 + 2\theta \| \underset{\sim}{\delta} \|^2 (\underset{\sim}{\delta}, H\underset{\sim}{\mu})$$

$$+ \theta^2 \{ (\underset{\sim}{\delta}, H\underset{\sim}{\mu})^2 - (\underset{\sim}{\delta}, H\underset{\sim}{\delta}) (\underset{\sim}{\mu}, H\underset{\sim}{\mu}) - (\underset{\sim}{\delta}, H\underset{\sim}{\delta}) (\underset{\sim}{\delta}, \underset{\sim}{\mu}) \} = 0. \qquad (52)$$

It is convenient to let $\phi = 1-\theta$, so, by using the equations $\underset{\sim}{\mu} = \underset{\sim}{\gamma} - G\underset{\sim}{\delta}$ and (35), we deduce that expression (52) is the same as the quadratic equation

$$\{ -0.1 \| \underset{\sim}{\delta} \|^4 + (\underset{\sim}{\delta}, H\underset{\sim}{\gamma})^2 - (\underset{\sim}{\delta}, H\underset{\sim}{\delta}) (\underset{\sim}{\eta}, \underset{\sim}{\gamma}) \}$$

$$-2\phi \{ (\underset{\sim}{\delta}, \underset{\sim}{\eta}) (\underset{\sim}{\delta}, H\underset{\sim}{\gamma}) - (\underset{\sim}{\delta}, H\underset{\sim}{\delta}) (\underset{\sim}{\eta}, \underset{\sim}{\gamma}) \}$$

$$+\phi^2 \{ (\underset{\sim}{\delta}, \underset{\sim}{\eta})^2 - (\underset{\sim}{\delta}, H\underset{\sim}{\delta}) (\underset{\sim}{\eta}, \underset{\sim}{\gamma}) \} = 0. \qquad (53)$$

To simplify this equation, we define σ to be the number

$$\sigma = (\underset{\sim}{\eta}, \underset{\sim}{\gamma}) (\underset{\sim}{\delta}, H\underset{\sim}{\delta}) - (\underset{\sim}{\gamma}, H\underset{\sim}{\delta})^2, \qquad (54)$$

for then expression (53) reduces to the equation

$$\{ -0.1 \| \underset{\sim}{\delta} \|^4 - \sigma \} + 2\phi \{ \sigma + (\underset{\sim}{\delta}, H\underset{\sim}{\gamma}) \| \underset{\sim}{\delta} \|^2 \}$$

$$+ \phi^2 \{ \| \underset{\sim}{\delta} \|^4 - 2 \| \underset{\sim}{\delta} \|^2 (\underset{\sim}{\delta}, H\underset{\sim}{\gamma}) - \sigma \} = 0. \qquad (55)$$

The zeros of this equation are

$$\frac{\sigma + 0.1 \, \| \underset{\sim}{\delta} \|^4}{\{\sigma+(\underset{\sim}{\delta},H\underset{\sim}{\gamma}) \, \| \underset{\sim}{\delta} \|^2 \} \pm \| \underset{\sim}{\delta} \|^2 \, [\{(\underset{\sim}{\delta},H\underset{\sim}{\gamma}) - 0.1 \, \| \underset{\sim}{\delta} \|^2\}^2 + 0.9\{\sigma+0.1 \, \| \underset{\sim}{\delta} \|^4\}]^{\frac{1}{2}}}. \tag{56}$$

Because $\phi = 1-\theta$, we require the zero that is smaller in modulus, so we let the $\pm$ sign of expression (56) be the same as the sign of $\{\sigma + (\underset{\sim}{\delta},H\underset{\sim}{\gamma}) \, \| \underset{\sim}{\delta} \|^2\}$. Thus instruction 221 of the Fortran listing sets CA to the required value of ϕ.

We could calculate the required matrix $G^{(k+1)}$ by substituting $G = G^{(k)}$, $\underset{\sim}{\delta} = \underset{\sim}{\delta}^{(k)}$, $\underset{\sim}{\mu} = \underset{\sim}{\gamma}^{(k)} - G^{(k)} \, \underset{\sim}{\delta}^{(k)}$ and $\theta = \theta^{(k)} = 1-CA$ in expression (36). However it is easy to verify that the same matrix can be obtained by setting $G = G^{(k)}$, $\underset{\sim}{\delta} = \underset{\sim}{\delta}^{(k)}$ and

$$\underset{\sim}{\gamma} = \theta^{(k)} \underset{\sim}{\gamma}^{(k)} + (1-\theta^{(k)}) \, G^{(k)} \underset{\sim}{\delta}^{(k)}$$

$$+ \theta^{(k)}(1-\theta^{(k)}) \, (\underset{\sim}{\gamma}^{(k)} - G^{(k)} \underset{\sim}{\delta}^{(k)}, \underset{\sim}{\delta}^{(k)}) \, \underset{\sim}{\delta}^{(k)} \, / \, \| \underset{\sim}{\delta}^{(k)} \|^2 \tag{57}$$

in expression (31). We prefer this second method, because it allows us to use equations (34) and (35) to define $H^{(k+1)}$, where $H = H^{(k)}$, $\underset{\sim}{\delta} = \underset{\sim}{\delta}^{(k)}$ and $\underset{\sim}{\gamma}$ is the vector (57). In the Fortran programme instruction 226 sets the elements of the array WC to the components of the vector (57), and instruction 225 sets the elements of WB to the corresponding components of $\underset{\sim}{\gamma} - G\underset{\sim}{\delta}$. Then instruction 228 branches to the part of the subroutine that evaluates the elements of the matrices (31) and (34). The description of the calculation of $G^{(k+1)}$ and $H^{(k+1)}$ is now complete.

Next we bound $|1-\theta^{(k)}|$, in order that equation (51) can be used to show that the iterations of the algorithm tend to reduce the error of the second derivative approximation.

Theorem 2 The given definition of $\theta^{(k)}$ implies the inequality

$$1 - \sqrt{2/11} \leq \theta^{(k)} \leq 1 + \sqrt{2/11} \, . \tag{58}$$

Proof Let $\psi(\theta)$ be the quadratic function

$$\psi(\theta) = \det\{G^*(\theta)\} \, / \, \det G, \tag{59}$$

where $G^*(\theta)$ is defined by equation (37). We note that $\psi(0) = 1$, and that $\theta^{(k)} = 1$ if $|\psi(1)| \geq 0.1$. Otherwise, if $|\psi(1)| < 0.1$, $\theta^{(k)}$ is the number closest to one such that $\psi(\theta^{(k)}) = 0.1$. Therefore it is sufficient to prove that if the coefficients of the quadratic function

$$\psi(\theta) \equiv 1 + a\theta + b\theta^2 \tag{60}$$

satisfy the condition

$$|1 + a + b| < 0.1, \tag{61}$$

then a zero of the equation $\psi(\theta) = 0.1$ is in the interval $\left[1 - \sqrt{2/11},\quad 1 + \sqrt{2/11}\right]$.

If condition (61) holds, and the theorem is false, then $\psi\left(1 \pm \sqrt{2/11}\right) < 0.1$, which gives the inequalities

$$1 + \left(1 - \sqrt{2/11}\right)a + \left(\{13/11\} - 2\sqrt{2/11}\right)b < 0.1 \tag{62}$$

and

$$1 + \left(1 + \sqrt{2/11}\right)a + \left(\{13/11\} + 2\sqrt{2/11}\right)b < 0.1. \tag{63}$$

Therefore if we multiply inequality (62) by $(11/18)\ (1+\sqrt{2/11})$ and inequality (63) by $(11/18)\ (1-\sqrt{2/11})$, and if we add these products, then we obtain the condition $(a + b) < -1.1$, which contradicts inequality (61). Theorem 2 is proved.

The last remarks of this section concern the numerical stability of formulae (31) and (34). The difficulty is that, due to computer rounding errors, H in fact will not equal the inverse of G, and therefore there will be discrepancies between G^* and $(H^*)^{-1}$. The rounding errors of a single iteration are nearly always tolerable, but in iterative numerical calculations like this one it sometimes happens that small errors accumulate in such a way that after a number of iterations the total error is disastrous. Fortunately we can prove that our formulae satisfy a nice stability theorem.

Theorem 3 If G and H are any non-singular nxn symmetric matrices, and if $\underset{\sim}{\delta}$ and $\underset{\sim}{\gamma}$ are n-component vectors subject only to the condition that the denominators of equations (31) and (34) are non-zero, then the matrices G* and H*, defined by equations (31) and (34), are related by the equation

$$G^* - (H^*)^{-1} = \left(I - \frac{\underset{\sim}{\delta\delta}^T}{\| \underset{\sim}{\delta} \|^2}\right) (G-H^{-1}) \left(I - \frac{\underset{\sim}{\delta\delta}^T}{\| \underset{\sim}{\delta} \|^2}\right). \tag{64}$$

Proof It is straightforward, but tedious, to verify that the inverse of H* is the matrix

$$H^{-1} + \frac{H^{-1}\underset{\sim}{\eta\delta}^T + \underset{\sim}{\delta}(H^{-1}\underset{\sim}{\eta})^T}{\| \underset{\sim}{\delta} \|^2} - \frac{\underset{\sim}{\delta\delta}^T(\underset{\sim}{\eta}, H^{-1}\underset{\sim}{\delta})}{\| \underset{\sim}{\delta} \|^4}. \tag{65}$$

Therefore, if we substitute $H\underset{\sim}{\gamma} - \underset{\sim}{\delta} = \underset{\sim}{\eta}$, equations (31) and (65) give the identity

$$\begin{aligned} G^* - (H^*)^{-1} &= G - \frac{G\underset{\sim}{\delta\delta}^T + \underset{\sim}{\delta\delta}^T G}{\| \underset{\sim}{\delta} \|^2} + \frac{\underset{\sim}{\delta\delta}^T(\underset{\sim}{\delta}, G\underset{\sim}{\delta})}{\| \underset{\sim}{\delta} \|^4} \\ &\quad - H^{-1} + \frac{H^{-1}\underset{\sim}{\delta\delta}^T + \underset{\sim}{\delta\delta}^T H^{-1}}{\| \underset{\sim}{\delta} \|^2} - \frac{\underset{\sim}{\delta\delta}^T(\underset{\sim}{\delta}, H^{-1}\underset{\sim}{\delta})}{\| \underset{\sim}{\delta} \|^4}. \end{aligned} \tag{66}$$

Because the right-hand sides of equations (64) and (66) are equal, the proof of Theorem 3 is complete.

Equation (64) states that any discrepancies between G and H^{-1} are multiplied by symmetric projection matrices, so any accumulation of errors is suppressed. Moreover, because the formulae (31) and (34) are used even when $\theta^{(k)} \neq 1$, this suppression of errors occurs on every iteration. Note that equations (51) and (64) show a correspondence between the numerical stability and the convergence of the second derivative approximations.

7. The special iterations of the algorithm

In Section 3 we stated that every third iteration of the algorithm is special, because these iterations do not use the method of Section 4 to calculate the correction

vector $\underset{\sim}{\delta}^{(k)}$. Instead $\underset{\sim}{\delta}^{(k)}$ is set to a value that is intended to improve the accuracy of the second derivative approximation $G^{(k+1)}$, and in fact $\underset{\sim}{\delta}^{(k)}$ is set to a multiple of the first row of the orthogonal matrix $D^{(k)}$. In this section we describe the calculation of $\underset{\sim}{\delta}^{(k)}$ by a special iteration, and we also define the calculation of $D^{(k+1)}$ for all $k \geqslant 1$. Finally we prove that, due to the special iterations, the sequence of directions $\underset{\sim}{\delta}^{(1)}$, $\underset{\sim}{\delta}^{(2)}$, ... satisfies a "strict linear independence condition". This condition enables one to prove some convergence properties of the algorithm (Powell, 1970).

These special iterations are included, not because they are needed to prove the convergence theorems, but because numerical experiments have indicated that they are worthwhile. I also found it desirable to include some special iterations when solving systems of non-linear equations, and the report on this work (Powell, 1968a) gives some reasons to justify the special iterations.

There are many possible ways of choosing the directions $\underset{\sim}{\delta}^{(k)}$ of the special iterations, and we decided to employ a method that would work very well in an ideal case. This case occurs when $F(\underset{\sim}{x})$ is exactly a quadratic function, with second derivative matrix $\bar{G}$ say, and when no value of $\theta^{(k)}$ is different from one in order to satisfy the condition (33). In this ideal case equation (51) shows that, for any $1 \leqslant q \leqslant k$, the error in $G^{(k+1)}$ is related to the error in $G^{(q)}$ by the equation

$$G^{(k+1)} - \bar{G} = P(q,k)^T (G^{(q)} - \bar{G}) P(q,k), \tag{67}$$

where $P(q,k)$ is the matrix

$$P(q,k) = \prod_{j=q}^{k} \left(I - \frac{\underset{\sim}{\delta}^{(j)} \underset{\sim}{\delta}^{(j)T}}{\| \underset{\sim}{\delta}^{(j)} \|^2} \right). \tag{68}$$

We would like the error matrix (67) to be zero, so the special iterations make some of the matrices $P(q,k)$ equal to zero.

When minimizing a non-quadratic function, it often happens that most of the iterations of the algorithm are spent in reaching a point that is close to the minimum, and then relatively few iterations are needed to achieve the required accuracy. A way of interpreting this behaviour is to say that the early iterations are needed to reach the neighbourhood of the minimum in which a quadratic function is an adequate approximation to $F(\underset{\sim}{x})$. Therefore the special iterations make some of the matrices (68) equal to zero

for large values of q and k, because equation (67) has some validity when $F(\underset{\sim}{x})$ is almost a quadratic function.

To make the matrix P(q,k) equal to zero, we have to reduce its rank to zero, and we note that the definition (68) gives the equation

$$\text{rank } P(q,k) = \begin{cases} \text{either rank } P(q,k-1) \\ \\ \text{or rank } P(q,k-1) - 1 , \end{cases} \tag{69}$$

where $1 \leqslant q \leqslant k-1$. Therefore the best that a special iteration can do is to calculate $\underset{\sim}{\delta}^{(k)}$ so that, if P(q,k-1) is not already zero, then the rank of P(q,k) is one less than the rank of P(q,k-1). In fact each special iteration calculates $\underset{\sim}{\delta}^{(k)}$ so that the second line of equation (69) holds for every integer q for which P(q,k-1) is non-zero. We now prove that we can do this.

Theorem 4 Given a sequence of non-zero vectors $\underset{\sim}{\delta}^{(1)}$, $\underset{\sim}{\delta}^{(2)}$, ..., $\underset{\sim}{\delta}^{(k-1)}$, there exists a vector $\underset{\sim}{\delta}^{(k)}$ such that the equation

$$\text{rank } P(q,k) = \max [0, \text{ rank } P(q,k-1)-1] \tag{70}$$

holds, for all q in $1 \leqslant q \leqslant k-1$, where the matrices P(q,k) are defined by equation (68).

Proof For $1 \leqslant q \leqslant k-1$, we let S_q be the right-hand null space of the matrix P(q,k-1), which is the definition

$$S_q = \{\underset{\sim}{x} | P(q,k-1)\underset{\sim}{x} = \underset{\sim}{0}\} . \tag{71}$$

Then the definition (68) implies the expression

$$S_1 \supset S_2 \supset S_3 \supset \ldots . \supset S_{k-1}. \tag{72}$$

We let d(q) be the dimension of S_q and expression (72) implies that there exists a set of n orthonormal vectors, $\underset{\sim}{\eta}_1, \underset{\sim}{\eta}_2, \ldots, \underset{\sim}{\eta}_n$ say, such that, for $1 \leqslant q \leqslant k-1$, the set S_q is spanned by the vectors $\underset{\sim}{\eta}_{n+1-d(q)}, \underset{\sim}{\eta}_{n+2-d(q)}, \ldots, \underset{\sim}{\eta}_n$. The theorem is proved by showing that it is sufficient to let $\underset{\sim}{\delta}^{(k)}$ be a multiple of $\underset{\sim}{\eta}_1$.

If $\underset{\sim}{\eta}_1$ is in the null space of S_q, then P(q,k-1) is the zero matrix, and therefore, by equation (68), P(q,k) is also the zero matrix, so equation (70) holds. And for the values of q for which P(q,k-1) is non-zero, $\underset{\sim}{\eta}_1$ must be orthogonal to all members of S_q. Therefore the definition (68) implies that, if $\underset{\sim}{\delta}^{(k)}$ is a multiple of $\underset{\sim}{\eta}_1$, then $P(q,k)\underset{\sim}{x} = \underset{\sim}{0}$ for all $\underset{\sim}{x}$ in S_q, and, moreover, the definition (68) gives the equation

$$P(q,k)\underset{\sim}{\delta}^{(k)} = \underset{\sim}{0} . \tag{73}$$

It follows that the null space of P(q,k) includes S_q and also $\underset{\sim}{\delta}^{(k)}$, so the rank of P(q,k) must be less than the rank of P(q,k-1). Therefore, because of expression (69), equation (70) holds. Theorem 4 is proved.

In the algorithm the rows of the orthogonal matrix $D^{(k)}$ are the vectors $\underset{\sim}{\eta}_1^T, \underset{\sim}{\eta}_2^T, \ldots, \underset{\sim}{\eta}_n^T$, defined in the proof of Theorem 4. Therefore we now understand why a special iteration sets $\underset{\sim}{\delta}^{(k)}$ to a multiple of the first row of $D^{(k)}$.

We now return to the Fortran listing of the appendix, and we describe the purpose of instructions 69-90, which are obeyed only on a special iteration. They are reached only if the variable ITSPEC of instruction 68 is negative, for it is the value of ITSPEC that ensures that exactly one out of three iterations is special. This part of the subroutine sets $\underset{\sim}{\delta}^{(k)}$ to a multiple of the first row of $D^{(k)}$, and it replaces the matrix $D^{(k)}$ by the matrix $D^{(k+1)}$.

The length of the correction $\underset{\sim}{\delta}^{(k)}$ satisfies inequality (13), and its sign makes $(\underset{\sim}{g}^{(k)}, \underset{\sim}{\delta}^{(k)}) \leq 0$, because we do not want the direction $\underset{\sim}{\delta}^{(k)}$ to be uphill with respect to $F(\underset{\sim}{x})$. Usually $\| \underset{\sim}{\delta}^{(k)} \| = \Delta^{(k)}$, but this is liable to be a poor choice if the length of the predicted gradient at $\underset{\sim}{x}^{(k)} + \underset{\sim}{\delta}^{(k)}$, namely $\| \underset{\sim}{g}^{(k)} + G^{(k)}\underset{\sim}{\delta}^{(k)} \|$, is much larger than $\| \underset{\sim}{g}^{(k)} \|$. Therefore $\| \underset{\sim}{\delta}^{(k)} \|$ is set to the smaller of $\Delta^{(k)}$ and $\| \underset{\sim}{g}^{(k)} \| / \| G^{(k)}\underset{\sim}{\eta}_1 \|$. The components of $\underset{\sim}{\delta}^{(k)}$ are calculated by instruction 82 of the Fortran listing.

It is easy to calculate the matrix $D^{(k+1)}$ on a special iteration, because the direction of the vector $\underset{\sim}{\delta}^{(k)}$ makes the null space of P(q,k) equal to all linear combinations of the vectors $\underset{\sim}{\eta}_{n+1-d(q)}, \underset{\sim}{\eta}_{n+2-d(q)}, \ldots, \underset{\sim}{\eta}_n$ and $\underset{\sim}{\eta}_1$. Moreover equation (73) implies that the null space of P(k,k) is $\underset{\sim}{\eta}_1$. Therefore the set of vectors $\underset{\sim}{\eta}_1, \underset{\sim}{\eta}_2, \ldots, \underset{\sim}{\eta}_n$ has to be replaced by the set $\underset{\sim}{\eta}_2, \underset{\sim}{\eta}_3, \ldots, \underset{\sim}{\eta}_n, \underset{\sim}{\eta}_1$. In other words to calculate $D^{(k+1)}$ we have only to shift the last (n-1) rows of $D^{(k)}$ up one position, and to set the last row of $D^{(k+1)}$ equal to the first row of $D^{(k)}$. These operations are carried out by

instructions 83 to 88 of the Fortran listing.

However, in the case of an ordinary iteration, the calculation of $D^{(k+1)}$ is less easy, for we require a method that is satisfactory for all possible values of $\underset{\sim}{\delta}^{(k)}$. We continue to use the notation $\underset{\sim}{\eta}_1^T, \underset{\sim}{\eta}_2^T, \ldots, \underset{\sim}{\eta}_n^T$ for the rows of $D^{(k)}$, and we let $\underset{\sim}{\zeta}_1^T, \underset{\sim}{\zeta}_2^T, \ldots, \underset{\sim}{\zeta}_n^T$ be the required rows of $D^{(k+1)}$. They are calculated by using the following theorem.

Theorem 5 The required vectors $\underset{\sim}{\zeta}_1, \underset{\sim}{\zeta}_2, \ldots, \underset{\sim}{\zeta}_n$ may be calculated in the following way. $\underset{\sim}{\zeta}_n$ is the vector

$$\underset{\sim}{\zeta}_n = \underset{\sim}{\delta}^{(k)} / \| \underset{\sim}{\delta}^{(k)} \| . \tag{74}$$

Let t be the greatest integer such that $(\underset{\sim}{\eta}_t, \underset{\sim}{\delta}^{(k)}) \neq 0$, and if $t \neq n$ then make the definition

$$\underset{\sim}{\zeta}_{j-1} = \underset{\sim}{\eta}_j, \quad j = t+1, \ t+2, \ \ldots, \ n. \tag{75}$$

And if $t \geq 2$, then, for $j=t-1,t-2,\ldots,1$, define $\underset{\sim}{\zeta}_j$ to be a linear combination of the vectors $\underset{\sim}{\eta}_j, \underset{\sim}{\eta}_{j+1}, \ldots, \underset{\sim}{\eta}_n$ that is orthogonal to the vectors $\underset{\sim}{\zeta}_{j+1}, \underset{\sim}{\zeta}_{j+2}, \ldots, \underset{\sim}{\zeta}_n$ and whose Euclidean length is one.

Proof In the text of the proof of Theorem 4 we have noted the required properties of the orthonormal vectors $\underset{\sim}{\zeta}_1, \underset{\sim}{\zeta}_2, \ldots, \underset{\sim}{\zeta}_n$. Specifically to prove the theorem we just have to show that, for all integers q in the interval $1 \leq q \leq k$, the vectors $\underset{\sim}{\zeta}_{n+1-c(q)}$, $\underset{\sim}{\zeta}_{n+2-c(q)}, \ldots, \underset{\sim}{\zeta}_n$ span the right-hand null space of the matrix P(q,k), where c(q) is the dimension of this null space.

First we note that the definitions (68) and (74) give the equation

$$P(q,k)\underset{\sim}{\zeta}_n = \underset{\sim}{0}, \quad 1 \leq q \leq k . \tag{76}$$

Therefore the choice of $\underset{\sim}{\zeta}_n$ is correct, and further, because the null space of the matrix P(k,k) has dimension one, from now on we need consider only values of q in the interval $1 \leq q \leq k-1$.

We divide these integers q into two parts, which depend on the values of c(q). Specifically, noting that equation (69) is equivalent to the expression

$$c(q) = \begin{cases} \text{either } d(q) \\ \qquad\qquad\quad , \ 1 \leqslant q \leqslant k-1, \\ \text{or } d(q) + 1 \end{cases} \tag{77}$$

where $d(q)$ is the dimension of the space (71), we first consider the values of q for which the second line of equation (77) holds, and then we treat the remaining values of q.

If $c(q) = d(q) + 1$, then there exist $d(q)$ independent vectors in the null space of $P(q,k)$ that are orthogonal to the vector (74). Further the equation

$$P(q,k) = P(q,k-1)\left(I - \frac{\underset{\sim}{\delta}^{(k)}\underset{\sim}{\delta}^{(k)T}}{\|\underset{\sim}{\delta}^{(k)}\|^2}\right) \tag{78}$$

shows that these $d(q)$ vectors are in the null space of $P(q,k-1)$. Therefore the vectors $\underset{\sim}{\eta}_{n+1-d(q)}, \underset{\sim}{\eta}_{n+2-d(q)}, \ldots, \underset{\sim}{\eta}_n, \underset{\sim}{\zeta}_n$ are an orthonormal basis of the null space of $P(q,k)$, and in particular the orthogonality condition

$$(\underset{\sim}{\eta}_j, \underset{\sim}{\delta}^{(k)}) = 0, \quad n+1-d(q) \leqslant j \leqslant n, \tag{79}$$

holds. Therefore the integer t, defined in the statement of the theorem, is less than $n+1-d(q)$, so the definition (75) makes the vectors $\underset{\sim}{\zeta}_{n+1-c(q)}, \underset{\sim}{\zeta}_{n+2-c(q)}, \ldots, \underset{\sim}{\zeta}_n$ the same as the vectors $\underset{\sim}{\eta}_{n+1-d(q)}, \underset{\sim}{\eta}_{n+2-d(q)}, \ldots, \underset{\sim}{\eta}_n, \underset{\sim}{\zeta}_n$, and we have already noted that these are a basis of the null space of $P(q,k)$. Therefore the construction of the theorem is adequate when the second line of equation (77) applies.

Finally we have to treat the case when q is such that $c(q) = d(q)$. We use the fact that, because $\underset{\sim}{\zeta}_j$ is constructed to be orthogonal to $\underset{\sim}{\delta}^{(k)}$ for $1 \leqslant j \leqslant n-1$, the definition (68) gives the equation

$$P(q,k-1)\underset{\sim}{\zeta}_j = P(q,k)\underset{\sim}{\zeta}_j, \quad 1 \leqslant j \leqslant n-1, \quad 1 \leqslant q \leqslant k-1. \tag{80}$$

Moreover, by construction for all $1 \leqslant j \leqslant n-1$, the vector $\underset{\sim}{\zeta}_j$ is a linear combination of the vectors $\underset{\sim}{\eta}_j, \underset{\sim}{\eta}_{j+1}, \ldots, \underset{\sim}{\eta}_n$. Therefore equation (80) and the definitions of $d(q)$ and $\{\underset{\sim}{\eta}_1, \underset{\sim}{\eta}_2, \ldots, \underset{\sim}{\eta}_n\}$ imply the identity

$$P(q,k)\underset{\sim}{\zeta}_j = 0, \quad n+1-d(q) \leq j \leq n-1, \quad 1 \leq q \leq k-1. \tag{81}$$

Further equation (76) shows that the range of j can be extended to $n+1-d(q) \leq j \leq n$. Therefore, in the case when $c(q) = d(q)$, the vectors $\underset{\sim}{\zeta}_{n+1-c(q)}, \underset{\sim}{\zeta}_{n+2-c(q)}, \ldots, \underset{\sim}{\zeta}_n$ span the null space of $P(q,k)$.

We have now treated all values of q in the interval $1 \leq q \leq k$, so the proof of Theorem 5 is complete.

The calculation of the matrix $D^{(k+1)}$ in the case of an ordinary iteration is made by instructions 127 to 155 of the Fortran listing. The method of Theorem 5 is applied, and first instruction 133 sets the elements of the working space array WB to the scalar products

$$(\underset{\sim}{\eta}_j, \underset{\sim}{\delta}^{(k)}) = \sigma_j, \quad j=1,2,\ldots,n, \tag{82}$$

say. Then when instruction 137 branches to instruction 140 the integer KK is set to the value of t defined in the statement of Theorem 5. To calculate the vectors $\underset{\sim}{\zeta}_j$ for $j=t-1,t-2,\ldots,1$, we follow the suggestion of Powell (1968b), in order to keep the amount of computing per iteration to within a multiple of n^2 operations. Specifically we let $\alpha_t = 0$ and $\underset{\sim}{\xi}_t = \underset{\sim}{Q}$, and then, for $j=t-1,t-2,\ldots,1$, we apply the recurrence relations

$$\left.\begin{aligned} \underset{\sim}{\xi}_j &= \underset{\sim}{\xi}_{j+1} + \sigma_{j+1}\,\underset{\sim}{\eta}_{j+1} \\ \alpha_j &= \alpha_{j+1} + \sigma_{j+1}^2 \\ \underset{\sim}{\zeta}_j &= \frac{\alpha_j \underset{\sim}{\eta}_j - \sigma_j \underset{\sim}{\xi}_j}{\sqrt{\alpha_j(\alpha_j+\sigma_j^2)}} \end{aligned}\right\}. \tag{83}$$

In the subroutine the working space vector WC is used for the current $\underset{\sim}{\xi}_j$, and the variable S is used for the current value of α_j. Note that because $\underset{\sim}{\eta}_j$ is required to define $\underset{\sim}{\xi}_{j-1}$ after $\underset{\sim}{\zeta}_j$ is calculated, the program first sets the vectors $\underset{\sim}{\zeta}_1, \underset{\sim}{\zeta}_2, \ldots, \underset{\sim}{\zeta}_{n-1}$ in the last (n-1) rows of the matrix DD. Then these rows are shifted to their correct positions by instruction 152.

This method for calculating the directions $\underset{\sim}{\delta}^{(k)}$ for the special iterations is like the method that I used in my subroutine for solving systems of non-linear equations

(Powell, 1968a). But it should be noted that there are some important differences. In particular the present method is based directly on the hypothesis that we would like to obtain the exact second derivative matrix of a quadratic function, but the earlier method has no analogous property.

To complete this section we prove a theorem that is required by Powell (1970). It takes account of the fact that in practice some of the numbers $\theta^{(k)}$ may be different from one, in order that condition (33) is satisfied.

Theorem 6 There exists a constant c, less than one, such that, for every integer $k \geqslant 1$, the inequality

$$\left\| \prod_{j=k}^{k+3n-3} \left(I - \theta^{(j)} \frac{\underset{\sim}{\delta}^{(j)} \underset{\sim}{\delta}^{(j)T}}{\| \underset{\sim}{\delta}^{(j)} \|^2} \right) \right\|_2 \leqslant c \tag{84}$$

holds, where the subscript "2" denotes that the matrix norm is induced by the Euclidean vector norm.

Proof First we show that the special iterations cause the identity

$$\prod_{j=k}^{k+3n-3} \left(I - \frac{\underset{\sim}{\delta}^{(j)} \underset{\sim}{\delta}^{(j)T}}{\| \underset{\sim}{\delta}^{(j)} \|^2} \right) = 0 . \tag{85}$$

We use the following four facts: (i) the rank of P(k,k) is (n-1), (ii) equation (69) holds for every ordinary iteration, (iii) every third iteration is a special iteration, and (iv) Theorem 4 is equivalent to the statement

$$\text{rank } P(k,j) = \max[0, \text{ rank } P(k,j-1)-1], \tag{86}$$

where $j > k$ is an integer that numbers a special iteration. Now as j runs from k+1 to k+3n-3, there are (n-1) special iterations, and therefore equation (86) holds (n-1) times. It follows that the rank of P(k,k+3n-3) is zero, so equation (85) is true.

We now suppose that in expression (84) the parameters $\theta^{(j)}$ and also the vectors $\underset{\sim}{\delta}^{(j)} / \| \underset{\sim}{\delta}^{(j)} \|$, $j=k,k+1,\ldots,k+3n-3$, are chosen in any manner that is consistent with the algorithm, and we try to identify an upper bound on the greatest value of the left-

hand side of the inequality. In particular a convenient upper bound is the greatest value of the function

$$\psi(\theta^{(1)},\theta^{(2)},\ldots,\theta^{(3n-2)},\underset{\sim}{\varrho}^{(1)},\underset{\sim}{\varrho}^{(2)},\ldots,\underset{\sim}{\varrho}^{(3n-2)})$$

$$= \Big\| \prod_{j=1}^{3n-2} (I - \theta^{(j)}\underset{\sim}{\varrho}^{(j)}\underset{\sim}{\varrho}^{(j)T}) \Big\|_2, \tag{87}$$

where each $\theta^{(j)}$ satisfies inequality (58), where each $\underset{\sim}{\varrho}^{(j)}$ has length one, and where, because of equation (85), the identity

$$\prod_{j=1}^{3n-2} (I - \underset{\sim}{\varrho}^{(j)}\underset{\sim}{\varrho}^{(j)T}) = 0 \tag{88}$$

holds. It follows that the range of the variables of the function (87) is closed and bounded, and moreover the definition (87) shows that this function depends continuously on its variables. Therefore the greatest value of the function is attained, and we suppose that it occurs when the variables have the values $\overline{\theta}^{(j)}$ and $\underset{\sim}{\overline{\varrho}}^{(j)}$ $(j=1,2,\ldots,3n-2)$. We let c be the greatest value of the function, and it remains to prove that $c < 1$.

There exists a normalized vector, $\underset{\sim}{\zeta}$ say, such that the equation

$$\Big\| \prod_{j=1}^{3n-2} \Big(I - \overline{\theta}^{(j)}\,\underset{\sim}{\overline{\varrho}}^{(j)}\underset{\sim}{\overline{\varrho}}^{(j)T} \Big) \underset{\sim}{\zeta} \Big\|_2 = c \tag{89}$$

holds. We let $\underset{\sim}{\zeta}^{(3n-1)} = \underset{\sim}{\zeta}$, and, for $j=3n-2,3n-1,\ldots,1$, we define $\underset{\sim}{\zeta}^{(j)}$ to be the vector

$$\underset{\sim}{\zeta}^{(j)} = (I - \overline{\theta}^{(j)}\underset{\sim}{\overline{\varrho}}^{(j)}\underset{\sim}{\overline{\varrho}}^{(j)T})\,\underset{\sim}{\zeta}^{(j+1)}. \tag{90}$$

Note that this equation implies the identity

$$\| \underset{\sim}{\zeta}^{(j)} \|^2 = \| \underset{\sim}{\zeta}^{(j+1)} \|^2 - \{2\overline{\theta}^{(j)} - [\overline{\theta}^{(j)}]^2\}(\underset{\sim}{\overline{\varrho}}^{(j)},\underset{\sim}{\zeta}^{(j+1)})^2, \tag{91}$$

and therefore, because $|\,1-\overline{\theta}^{(j)}\,| \leqslant \sqrt{2/11}$, the inequalities

$$c = \|\underset{\sim}{\zeta}^{(1)}\| \leq \|\underset{\sim}{\zeta}^{(2)}\| \leq \ldots \leq \|\underset{\sim}{\zeta}^{(3n-1)}\| = 1 \tag{92}$$

hold. It follows that the theorem is false only if $c = 1$, in which case all the above inequalities become equalities, and then equation (91) implies that $(\underset{\sim}{\bar{\sigma}}^{(j)}, \underset{\sim}{\zeta}^{(j+1)}) = 0$, $j=1,2,\ldots,3n-2$. If this happens then the equation

$$\prod_{j=1}^{3n-2} (I - \underset{\sim}{\bar{\sigma}}^{(j)} \underset{\sim}{\bar{\sigma}}^{(j)T}) \underset{\sim}{\zeta} = \underset{\sim}{\zeta} \tag{93}$$

holds, which contradicts the constraint (88) on the vectors $\underset{\sim}{\sigma}^{(j)}$. Theorem 6 is proved.

8. Two numerical examples

In this section we do not make a comparison with other algorithms, because already a comparison has been published (Powell, 1970). Instead we give two numerical examples, so that anyone who punches cards from a Fortran listing in the Appendix will have some numbers to test his subroutine. However the second example shows that small perturbations, due for instance to computer rounding errors, can cause large changes to the computed sequence $\underset{\sim}{x}^{(k)}$ $(k=1,2,\ldots)$. In this example the final solution is always correct, but the path to the solution is very sensitive to computer rounding errors.

The first example is the calculation of the least value of the function

$$F(\underset{\sim}{x}) = x_1^2 + 2x_2^2 + 3x_3^2 + 4x_4^2 + (x_1 + x_2 + x_3 + x_4)^4, \tag{94}$$

starting from the point $\underset{\sim}{x}^{(1)} = (1,-1,-1,1)$, with the parameter values STEP = 0.1 and ACC = 10^{-10} (see Section 2). This problem was solved by subroutine VA06A on an IBM 360/75 computer in single length arithmetic. The values of $\underset{\sim}{x}$ and $F(\underset{\sim}{x})$ were printed after every evaluation of the objective function and its gradient, and they are given in Table 1. Note that the tabulated vectors $\underset{\sim}{x}$ are not the vectors $\underset{\sim}{x}^{(k)}$ $(k=1,2,\ldots,20)$, because $\underset{\sim}{x}^{(k)}$ $(k=2,3,\ldots,20)$ is defined by equation (11). Instead they are the vectors $\underset{\sim}{x}^{(1)}$ and $\underset{\sim}{x}^{(k)} + \underset{\sim}{\delta}^{(k)}$ $(k=1,2,\ldots,19)$. Every third row of the table is the result of a "special iteration".

The second example is the minimization of Rosenbrock's (1960) function

$$F(x_1,x_2) = 100(x_2-x_1^2)^2 + (1-x_1)^2, \tag{95}$$

starting from (-1.2, 1.0), with the parameter values STEP = 0.1 and ACC = 0.0001. Using single length arithmetic on the I.B.M. 360/75 computer, 41 evaluations of $F(x)$ and its gradient were required to complete the calculation, but in double length arithmetic 43 calls of subroutine CALCFG were made. The values of $x^{(k)}$ and $F(x^{(k)})$ in these two cases are given in Table 2. The value of $F(x^{(1)})$ differs from 24.20000 even when double length arithmetic is in use, because the Fortran instruction X(1) = -1.2 provides the number -1.19999980927.

The table shows a very great difference between the single and double length calculations, and the main discrepancies stem from the calculation of $\delta^{(k)}$ by the third iteration. $\delta^{(3)}$ is defined by the method of Section 4, so the vectors $g^{(3)}$, $s^{(3)}$ and $v^{(3)}$ are important. It happens that in the single length version these vectors have the values $g^{(3)}$ = (-87.2567, -36.2234), $v^{(3)}$ = (0.06765, 0.01548) and $z^{(3)}$ = (0.06351, 0.02637), and in the double length version they have the values $g^{(3)}$ = (-87.2580, -36.2239), $x^{(3)}$ = (0.06811, 0.01536) and $z^{(3)}$ = (0.06351, 0.02637). Therefore in the first case the value of expression (20) is equal to 0.033, and in the second case it is equal to -0.003. It follows that when expression (23) is evaluated the signs of THETA differ in the two cases, which corresponds to selecting different roots of the quadratic equation (21). Therefore there are large discrepancies between the two calculated values of $\delta^{(3)}$.

One purpose of Table 2 is to show that if an algorithm depends on any discrete decisions, like estimating the sign of $(z^{(k)}, w^{(k)})$ in expression (23), then a small change in the accuracy of the calculation can cause a large change in the progress of the algorithm. Therefore the user may not be able to reproduce the results given in the tables. However he should find that the algorithm of this report compares favourably with other methods for unconstrained minimization.

Table 1

The minimization of a function of four variables

x_1	x_2	x_3	x_4	$F(\underset{\sim}{x})$
1.00000	-1.00000	-1.00000	1.00000	10.00000
0.98174	-0.96348	-0.94523	0.92697	8.93789
0.78510	-0.97705	-0.96557	0.95409	8.96549
0.94358	-0.88857	-0.83499	0.78282	7.01232
0.85945	-0.73013	-0.61165	0.50365	3.94183
0.86534	0.00170	-0.80601	0.76173	5.47700
0.64517	-0.36607	-0.15418	0.00138	0.75585
-0.03757	-0.21769	0.48077	0.25445	1.10165
0.43286	-0.30621	-0.30809	-0.22038	0.87998
0.29596	-0.07597	0.02673	0.00100	0.10505
0.03559	0.02964	-0.00704	0.01443	0.00403
0.04002	0.04771	0.00716	-0.00605	0.00652
0.01523	0.01198	-0.00114	0.00534	0.00064
6.52×10^{-4}	1.56×10^{-4}	3.42×10^{-4}	1.62×10^{-4}	9.31×10^{-6}
3.65×10^{-4}	3.98×10^{-4}	4.80×10^{-6}	3.21×10^{-4}	8.63×10^{-6}
-1.63×10^{-5}	4.41×10^{-6}	1.49×10^{-5}	1.30×10^{-5}	1.64×10^{-9}
-3.78×10^{-6}	-2.50×10^{-7}	1.95×10^{-6}	1.38×10^{-6}	3.34×10^{-11}
-5.27×10^{-6}	2.68×10^{-6}	3.72×10^{-7}	3.39×10^{-7}	4.30×10^{-11}
-2.80×10^{-10}	-1.75×10^{-10}	-3.49×10^{-10}	2.56×10^{-10}	7.69×10^{-19}
-3.10×10^{-13}	-2.86×10^{-13}	-6.27×10^{-13}	4.27×10^{-13}	2.17×10^{-24}

Table 2

The minimization of Rosenbrock's function

	Single Length			Double Length		
k	$x_1^{(k)}$	$x_2^{(k)}$	$F(\underset{\sim}{x}^{(k)})$	$x_1^{(k)}$	$x_2^{(k)}$	$F(\underset{\sim}{x}^{(k)})$
1	-1.20000	1.00000	24.19996	-1.20000	1.00000	24.19996
2	-1.10741	1.03779	7.99731	-1.10742	1.03779	7.99738
3	-1.14520	1.13037	7.88224	-1.14520	1.13037	7.88233
4	-1.10679	1.22270	4.43910	-1.07710	1.14573	4.33511
5	-1.09193	1.19540	4.37712	-1.01346	1.02144	4.05723
6	-1.08992	1.19649	4.37511	-1.00877	1.02385	4.03904
7	-1.07138	1.13715	4.30210	-0.88196	0.77503	3.54257
8	-1.01237	1.02771	4.05042	-0.88196	0.77503	3.54257
9	-1.01012	1.02893	4.04795	-0.87783	0.77726	3.53069
10	-0.90786	0.80225	3.68810	-0.78880	0.62186	3.19981
11	-0.78341	0.58695	3.25226	-0.60268	0.31582	2.79322
12	-0.77225	0.59340	3.14176	-0.58454	0.32685	2.53277
13	-0.50000	0.17718	2.78017	-0.34642	0.05928	2.18170
14	-0.58534	0.34881	2.51713	-0.38178	0.14240	1.91044
15	-0.58255	0.35020	2.51620	-0.37485	0.14535	1.89255
16	-0.41456	0.15341	2.03505	-0.22002	0.05223	1.48992
17	-0.22406	-0.02168	2.01503	-0.22002	0.05223	1.48992
18	-0.19995	0.00455	1.56541	-0.22002	0.05223	1.48992
19	-0.07291	-0.01987	1.21456	-0.17349	0.03337	1.37815
20	0.03433	-0.02448	0.99836	-0.07753	0.00378	1.16157
21	0.03542	0.00093	0.93042	-0.07471	0.01293	1.16040
22	0.23626	0.01537	0.74692	0.11734	-0.04583	1.13427
23	0.23132	0.05566	0.59133	0.05631	0.00778	0.89268
24	0.25517	0.05858	0.55904	0.05631	0.00778	0.89268
25	0.32255	0.10385	0.45894	0.21789	0.02467	0.66369
26	0.46725	0.17747	0.45077	0.35856	0.03505	0.60086
27	0.44840	0.21452	0.32235	0.34052	0.12709	0.44730
28	0.51585	0.25969	0.23852	0.47161	0.20613	0.30573
29	0.62864	0.37647	0.17296	0.58026	0.31397	0.22787
30	0.61818	0.38657	0.14774	0.56556	0.32878	0.19669
31	0.72242	0.51104	0.08881	0.67099	0.43976	0.11921
32	0.81313	0.64570	0.05888	0.76104	0.56355	0.08152
33	0.80412	0.65177	0.04104	0.75187	0.57022	0.06398
34	0.89182	0.78839	0.01655	0.83867	0.69631	0.03101
35	0.94496	0.88660	0.00707	0.91635	0.82822	0.02017
36	0.94179	0.88832	0.00357	0.91030	0.83178	0.00903
37	0.96803	0.93640	0.00107	0.96892	0.93581	0.00187
38	1.00090	0.99991	0.00035	0.98494	0.96913	0.00032
39	1.00008	1.00034	0.00000	0.98452	0.96932	0.00024
40	0.99995	0.99991	0.00000	0.99358	0.98710	0.00004
41	1.00000	1.00000	0.00000	0.99988	0.99966	0.00000
42				0.99984	0.99968	0.00000
43				1.00000	0.99999	0.00000

References

Broyden, C.G. (1970) "The convergence of a class of double-rank minimisation algorithms. 1. General considerations", J. Inst. Maths. Applics., Vol.6, pp. 76-90.

Davidon, W.C. (1959) "Variable metric method for minimization", A.E.C. Research and Development Report, ANL-5990 (Rev.).

Householder, A.S. (1964) "The theory of matrices in numerical analysis", Blaisdell Publishing Co.

Powell, M.J.D. (1968a) "A Fortran subroutine for solving systems of non-linear algebraic equations", Report No. R-5947, A.E.R.E., Harwell.

Powell, M.J.D. (1968b) "On the calculation of orthogonal vectors", Computer Journal, Vol. 11, pp.302-304.

Powell, M.J.D. (1969) "Rank one methods for unconstrained optimization", Report No. T.P.372, A.E.R.E., Harwell.

Powell, M.J.D. (1970) "A new algorithm for unconstrained optimization", Report No. T.P.393, A.E.R.E., Harwell.

Rosenbrock, H.H. (1960) "An automatic method for finding the greatest or the least value of a function", Computer Journal, Vol. 3, pp. 175-184.

```
      SUBROUTINE MINFA (N,X,F,G,STEP,ACC,MAXFUN,IPRINT)
      DIMENSION X(50),G(50),XA(50),GA(50),GG(50,50),H(50,50),
     1DD(50,50),WA(50),WB(50),WC(50)
C     SET SOME CONSTANTS
      ACCT=ACC*ACC
      IPP=IPRINT*IPRINT
C     GIVE INITIAL VALUES TO SOME VARIABLES
      DSS=STEP*STEP
      MAXC=1
      ITSPEC=1
      IPTEST=1
C     CALCULATE THE INITIAL GRADIENT
      CALL CALCFG (N,X,F,G)
C     GIVE INITIAL VALUES TO THE COMPONENTS OF GG, H AND DD
      GSQ=0.
      DO 1 I=1,N
    1 GSQ=GSQ+G(I)**2
      IF (GSQ) 5,5,2
    2 GGDIAG=0.01*SQRT(GSQ)/STEP
      HDIAG=1./GGDIAG
      DO 4 I=1,N
      DO 3 J=1,N
      GG(I,J)=0.
      H(I,J)=0.
    3 DD(I,J)=0.
      GG(I,I)=GGDIAG
      H(I,I)=HDIAG
    4 DD(I,I)=1.
C     ARRANGE FOR ANY PRINTING TO BEGIN ON A NEW PAGE
    5 IF (IPP) 10,10,6
    6 PRINT 7
    7 FORMAT(1H1,5X,'THE FOLLOWING OUTPUT IS PROVIDED BY MINFA')
      GO TO 10
C     BEGIN AN ITERATION BY TESTING FOR CONVERGENCE
    8 GSQ=0.
      DO 9 I=1,N
    9 GSQ=GSQ+G(I)**2
   10 IF (GSQ-ACCT) 11,11,18
C     PRINT THE FINAL VALUES OF THE FUNCTION AND GRADIENT
   11 IF (IPP) 17,17,12
   12 PRINT 13,MAXC,F
   13 FORMAT (/5X,'AFTER',I4,' CALLS OF CALCFG, THE FINAL F =',
     1E20.12)
      PRINT 14,(X(I),I=1,N)
   14 FORMAT (5X,'X =',(5E20.12))
      IF (IPRINT) 17,17,15
   15 PRINT 16,(G(I),I=1,N)
   16 FORMAT (5X,'G =',(5E20.12))
   17 RETURN
C     TEST WHETHER MAXFUN CALLS OF CALCFG HAVE BEEN MADE
   18 IF (MAXC-MAXFUN) 21,19,19
   19 PRINT 20,MAXC
   20 FORMAT (/5X,'MINFA HAS MADE',I5,' CALLS OF CALCFG')
      GO TO 11
C     PRINT THE CURRENT BEST VALUE OF F ETC
   21 IPTEST=IPTEST-IABS(IPRINT)
      IF (IPTEST) 22,22,25
   22 IPTEST=IPP
```

```
      PRINT 23,MAXC,F
   23 FORMAT(/5X,'AT THE START OF ITERATION',I4,5X,'F =',E20.12)
      PRINT 14,(X(I),I=1,N)
      IF (IPRINT) 25,25,24
   24 PRINT 16,(G(I),I=1,N)
C     TEST WHETHER A SPECIAL ITERATION IS NEEDED, AND CALCULATE
C     THE  CHANGE IN GRADIENT ALONG THE DIRECTION OF
C     A SPECIAL ITERATION
   25 ITSPEC=ITSPEC-1
      IF (ITSPEC) 26,32,32
   26 DGGD=0.
      SGDD=0.
      DO 28 I=1,N
      SUM=0.
      DO 27 J=1,N
   27 SUM=SUM+GG(I,J)*DD(1,J)
      SGDD=SGDD+G(I)*DD(1,I)
   28 DGGD=DGGD+SUM*SUM
C     CALCULATE THE CORRECTION FOR A SPECIAL ITERATION
C     AND REVISE THE ARRAY DD
      DSQ=AMIN1(DSS,GSQ/DGGD)
      C=SIGN(SQRT(DSQ),-SGDD)
      DO 29 I=1,N
      WA(I)=C*DD(1,I)
   29 WB(I)=DD(1,I)
      DO 30 I=2,N
      DO 30 J=1,N
   30 DD(I-1,J)=DD(I,J)
      DO 31 I=1,N
   31 DD(N,I)=WB(I)
      ITSPEC=2
      GO TO 51
C     CALCULATE THE GENERALIZED NEWTON CORRECTION TO X
C     AND PREDICT THE CURVATURE OF F ALONG THE GRADIENT
   32 GGGG=0.
      DO 34 I=1,N
      WA(I)=0.
      SUM=0.
      DO 33 J=1,N
      WA(I)=WA(I)-H(I,J)*G(J)
   33 SUM=SUM+GG(I,J)*G(J)
   34 GGGG=GGGG+SUM*G(I)
C     TEST WHETHER TO SET THE CORRECTION TO A MULTIPLE OF
C     THE GRADIENT
      IF (GGGG*ABS(GGGG)*DSS-GSQ**3) 35,35,37
C     SET THE CORRECTION VECTOR TO A MULTIPLE OF THE GRADIENT
   35 C=-SQRT(DSS/GSQ)
      DO 36 I=1,N
   36 WA(I)=C*G(I)
      GO TO 41
C     SET THE OPTIMAL STEEPEST DESCENT CORRECTION IN WB
C     AND THE DIFFERENCE BETWEEN WA AND WB IN WC
   37 C=-GSQ/GGGG
      CA=0.
      CB=0.
      DO 38 I=1,N
      WB(I)=C*G(I)
      WC(I)=WA(I)-WB(I)
```

```
      CA=CA+WB(I)*WC(I)
   38 CB=CB+WC(I)**2
C     INTERPOLATE FOR THE CORRECTION VECTOR IN THE LINE WA - WB
      C=DSS-C*C*GSQ
      THETA=SIGN(C/(ABS(CA)+SQRT(CA*CA+C*CB)),CA)
C     TEST WHETHER TO USE THE GENERALIZED NEWTON CORRECTION
      IF (THETA-1.) 39,41,41
   39 DO 40 I=1,N
   40 WA(I)=WB(I)+THETA*WC(I)
C     EXPRESS THE CORRECTION VECTOR IN TERMS OF THE ROWS OF DD
   41 DSQ=0.
      DO 42 I=1,N
      DSQ=DSQ+WA(I)**2
      WB(I)=0.
      WC(I)=0.
      DO 42 J=1,N
   42 WB(I)=WB(I)+DD(I,J)*WA(J)
C     REVISE THE DIRECTIONS IN THE ARRAY DD
      S=0.
      KK=N
   43 IF (WB(KK)) 45,44,45
   44 KK=KK-1
      GO TO 43
   45 KK=KK-1
      IF (KK) 48,48,46
   46 S=S+WB(KK+1)**2
      C=SQRT(S*(S+WB(KK)**2))
      CA=S/C
      CB=WB(KK)/C
      DO 47 J=1,N
      WC(J)=WC(J)+WB(KK+1)*DD(KK+1,J)
   47 DD(KK+1,J)=CA*DD(KK,J)-CB*WC(J)
      GO TO 45
   48 DO 49 I=2,N
      DO 49 J=1,N
   49 DD(I-1,J)=DD(I,J)
      C=1./SQRT(DSQ)
      DO 50 I=1,N
   50 DD(N,I)=C*WA(I)
C     APPLY THE CORRECTION VECTOR AND CALCULATE THE
C     OBJECTIVE FUNCTION
   51 DO 52 I=1,N
   52 XA(I)=X(I)+WA(I)
      MAXC=MAXC+1
      CALL CALCFG (N,XA,FA,GA)
C     SET THE ERROR OF THE PREDICTED GRADIENT IN WB
C     ALSO CALCULATE SOME NUMBERS FOR REVISING THE STEP-BOUND
      DG=0.
      DGA=0.
      DGGD=0.
      WBSQ=0.
      DO 54 I=1,N
      SUM=0.
      DO 53 J=1,N
   53 SUM=SUM+GG(I,J)*WA(J)
      WB(I)=GA(I)-G(I)-SUM
      DG=DG+G(I)*WA(I)
      DGA=DGA+GA(I)*WA(I)
```

```
175:          DGGD=DGGD+SUM*WA(I)
176:       54 WBSQ=WBSQ+WB(I)**2
177:C        TEST WHETHER TO DECREASE THE STEP-BOUND
178:          IF (ITSPEC-2) 55,60,60
179:       55 IF (FA-F-0.1*DG-0.05*DGGD) 57,57,56
180:       56 DSS=0.25*DSQ
181:          GO TO 60
182:C        TEST WHETHER TO INCREASE THE STEP-BOUND
183:       57 DSS=DSQ
184:          IF (WBSQ-0.25*GSQ) 59,59,58
185:       58 IF (DG-DGA-DGA) 60,59,59
186:       59 DSS=4.*DSQ
187:C        SET THE DIFFERENCE BETWEEN GRADIENTS
188:       60 DO 61 I=1,N
189:       61 WC(I)=GA(I)-G(I)
190:C        SET X, F AND G TO THE BEST CALCULATED VALUES
191:          ITHETA=1
192:          IF (F-FA) 64,64,62
193:       62 F=FA
194:          DO 63 I=1,N
195:          X(I)=XA(I)
196:       63 G(I)=GA(I)
197:C        CALCULATE SOME VECTORS AND SCALAR PRODUCTS TO
198:C        REVISE GG AND H
199:       64 SDM=0.
200:          GHD=0.
201:          SEG=0.
202:          DHD=0.
203:          DO 66 I=1,N
204:          XA(I)=-WA(I)
205:          GA(I)=0.
206:          DO 65 J=1,N
207:          XA(I)=XA(I)+H(I,J)*WC(J)
208:       65 GA(I)=GA(I)+H(I,J)*WA(J)
209:          SDM=SDM+WA(I)*WB(I)
210:          GHD=GHD+WC(I)*GA(I)
211:          SEG=SEG+WC(I)*XA(I)
212:       66 DHD=DHD+WA(I)*GA(I)
213:C        TEST WHETHER THE USUAL CORRECTION TO GG GIVES
214:C        NEAR SINGULARITY
215:          HDIV=SEG*DHD-GHD*GHD
216:          GO TO (67,70),ITHETA
217:       67 IF (ABS(HDIV)-0.1*DSQ*DSQ) 68,70,70
218:C        CHANGE THE DIFFERENCE IN GRADIENTS TO AVOID SINGULARITY
219:       68 CA=HDIV/(DSQ*DSQ)+0.1
220:          CB=GHD/DSQ-0.1
221:          CA=CA/(CA+CB+SIGN(SQRT(0.9*CA+CB*CB),CA+CB))
222:          CB=(1.-CA)*SDM/DSQ
223:          DO 69 I=1,N
224:          C=CA*(CB*WA(I)-WB(I))
225:          WB(I)=WB(I)+C
226:       69 WC(I)=WC(I)+C
227:          ITHETA=2
228:          GO TO 64
229:C        REVISE THE MATRICES GG AND H
230:       70 CA=1./DSQ
231:          CB=SDM*CA*CA
232:          CC=DHD/HDIV
```

```
      CD=GHD/HDIV
      CE=SEG/HDIV
      DO 71 I=1,N
      DO 71 J=I,N
      GG(I,J)=GG(I,J)+CA*(WA(I)*WB(J)+WA(J)*WB(I))
     1-CB*WA(I)*WA(J)
      H(I,J)=H(I,J)-CC*XA(I)*XA(J)+CD*(XA(I)*GA(J)+XA(J)*GA(I))
     1-CE*GA(I)*GA(J)
      GG(J,I)=GG(I,J)
   71 H(J,I)=H(I,J)
      GO TO 8
      END
```

```
      SUBROUTINE VA06A (N,X,F,G,STEP,ACC,MAXFUN,IPRINT,W)
      DIMENSION X(1),G(1),W(1)
C     THE NEXT SEVEN INTEGERS PARTITION THE ARRAY W
      IDD=N*N+N
      IH=IDD/2
      IXA=IDD+N*N
      IGA=IXA+N
      IWA=IGA+N
      IWB=IWA+N
      IWC=IWB+N
C     SET SOME CONSTANTS
      ACCT=ACC*ACC
      IPP=IPRINT*IPRINT
C     GIVE INITIAL VALUES TO SOME VARIABLES
      DSS=STEP*STEP
      MAXC=1
      ITSPEC=1
      IPTEST=1
C     CALCULATE THE INITIAL GRADIENT
      CALL CALCFG (N,X,F,G)
C     GIVE INITIAL VALUES TO THE COMPONENTS OF GG, H AND DD
      GSQ=0.
      DO 1 I=1,N
    1 GSQ=GSQ+G(I)**2
      IF (GSQ) 5,5,2
    2 GGDIAG=0.01*SQRT(GSQ)/STEP
      HDIAG=1./GGDIAG
      K=0
      KDD=IDD
      DO 4 I=1,N
      J=1
  101 IF (J-I) 102,103,103
  102 KDD=KDD+1
      W(KDD)=0.
      J=J+1
      GO TO 101
  103 K=K+1
      KDD=KDD+1
      W(K)=GGDIAG
      W(IH+K)=HDIAG
      W(KDD)=1.
    3 J=J+1
      IF (J-N) 104,104,4
  104 K=K+1
      KDD=KDD+1
      W(K)=0.
      W(IH+K)=0.
      W(KDD)=0.
      GO TO 3
    4 CONTINUE
C     ARRANGE FOR ANY PRINTING TO BEGIN ON A NEW PAGE
    5 IF (IPP) 10,10,6
    6 PRINT 7
    7 FORMAT(1H1,5X,'THE FOLLOWING OUTPUT IS PROVIDED BY VA06A')
      GO TO 10
C     BEGIN AN ITERATION BY TESTING FOR CONVERGENCE
    8 GSQ=0.
      DO 9 I=1,N
```

```
  59:     9 GSQ=GSQ+G(I)**2
  60:    10 IF (GSQ-ACCT) 11,11,18
  61:C      PRINT THE FINAL VALUES OF THE FUNCTION AND GRADIENT
  62:    11 IF (IPP) 17,17,12
  63:    12 PRINT 13,MAXC,F
  64:    13 FORMAT(/5X,'AFTER',I4,' CALLS OF CALCFG, THE FINAL F =',
  65:      1E14.6)
  66:       PRINT 14,(X(I),I=1,N)
  67:    14 FORMAT(5X,'X =',(8E14.6))
  68:       IF (IPRINT) 17,17,15
  69:    15 PRINT 16,(G(I),I=1,N)
  70:    16 FORMAT(5X,'G =',(8E14.6))
  71:    17 RETURN
  72:C      TEST WHETHER MAXFUN CALLS OF CALCFG HAVE BEEN MADE
  73:    18 IF (MAXC-MAXFUN) 21,19,19
  74:    19 PRINT 20,MAXC
  75:    20 FORMAT(/5X,'VA06A HAS MADE',I5,' CALLS OF CALCFG')
  76:       GO TO 11
  77:C      PRINT THE CURRENT BEST VALUE OF F ETC
  78:    21 IPTEST=IPTEST-IABS(IPRINT)
  79:       IF (IPTEST) 22,22,25
  80:    22 IPTEST=IPP
  81:       PRINT 23,MAXC,F
  82:    23 FORMAT(/5X,'AT THE START OF ITERATION',I4,5X,'F =',E14.6)
  83:       PRINT 14,(X(I),I=1,N)
  84:       IF (IPRINT) 25,25,24
  85:    24 PRINT 16,(G(I),I=1,N)
  86:C      TEST WHETHER A SPECIAL ITERATION IS NEEDED, AND CALCULATE
  87:C      THE CHANGE IN GRADIENT ALONG THE DIRECTION OF
  88:C      A SPECIAL ITERATION
  89:    25 ITSPEC=ITSPEC-1
  90:       IF (ITSPEC) 26,32,32
  91:    26 DGGD=0.
  92:       SGDD=0.
  93:       KDD=IDD
  94:       DO 28 I=1,N
  95:       SUM=0.
  96:       K=I
  97:       KD=IDD
  98:       J=1
  99:   105 IF (J-I) 106,107,107
 100:   106 KD=KD+1
 101:       SUM=SUM+W(K)*W(KD)
 102:       K=K+N-J
 103:       J=J+1
 104:       GO TO 105
 105:   107 DO 27 J=I,N
 106:       KD=KD+1
 107:       SUM=SUM+W(K)*W(KD)
 108:    27 K=K+1
 109:       KDD=KDD+1
 110:       SGDD=SGDD+G(I)*W(KDD)
 111:    28 DGGD=DGGD+SUM*SUM
 112:C      CALCULATE THE CORRECTION FOR A SPECIAL ITERATION
 113:C      AND REVISE THE ARRAY DD
 114:       DSQ=AMIN1(DSS,GSQ/DGGD)
 115:       C=SIGN(SQRT(DSQ),-SGDD)
 116:       DO 29 I=1,N
```

```
      W(I+IWA)=C*W(I+IDD)
   29 W(I+IWB)=W(I+IDD)
      KDD=IDD
      DO 30 I=2,N
      DO 30 J=1,N
      KDD=KDD+1
   30 W(KDD)=W(KDD+N)
      DO 31 I=1,N
      KDD=KDD+1
   31 W(KDD)=W(I+IWB)
      ITSPEC=2
      GO TO 51
C     CALCULATE THE GENERALIZED NEWTON CORRECTION TO X
C     AND PREDICT THE CURVATURE OF F ALONG THE GRADIENT
   32 GGGG=0.
      DO 34 I=1,N
      W(I+IWA)=0.
      SUM=0.
      J=1
      K=I
  108 IF (J-I) 109,110,110
  109 W(I+IWA)=W(I+IWA)-W(IH+K)*G(J)
      SUM=SUM+W(K)*G(J)
      K=K+N-J
      J=J+1
      GO TO 108
  110 DO 33 J=I,N
      W(I+IWA)=W(I+IWA)-W(IH+K)*G(J)
      SUM=SUM+W(K)*G(J)
   33 K=K+1
   34 GGGG=GGGG+SUM*G(I)
C     TEST WHETHER TO SET THE CORRECTION TO A MULTIPLE OF
C     THE GRADIENT
      IF (GGGG*ABS(GGGG)*DSS-GSQ**3) 35,35,37
C     SET THE CORRECTION VECTOR TO A MULTIPLE OF THE GRADIENT
   35 C=-SQRT(DSS/GSQ)
      DO 36 I=1,N
   36 W(I+IWA)=C*G(I)
      GO TO 41
C     SET THE OPTIMAL STEEPEST DESCENT CORRECTION IN WB
C     AND THE DIFFERENCE BETWEEN WA AND WB IN WC
   37 C=-GSQ/GGGG
      CA=0.
      CB=0.
      DO 38 I=1,N
      W(I+IWB)=C*G(I)
      W(I+IWC)=W(I+IWA)-W(I+IWB)
      CA=CA+W(I+IWB)*W(I+IWC)
   38 CB=CB+W(I+IWC)**2
C     INTERPOLATE FOR THE CORRECTION VECTOR ON THE LINE WA - WB
      C=DSS-C*C*GSQ
      THETA=SIGN(C/(ABS(CA)+SQRT(CA*CA+C*CB)),CA)
C     TEST WHETHER TO USE THE GENERALIZED NEWTON CORRECTION
      IF (THETA-1.) 39,41,41
   39 DO 40 I=1,N
   40 W(I+IWA)=W(I+IWB)+THETA*W(I+IWC)
C     EXPRESS THE CORRECTION VECTOR IN TERMS OF THE ROWS OF DD
   41 DSQ=0.
```

```
      KDD=IDD
      DO 42 I=1,N
      DSQ=DSQ+W(I+IWA)**2
      W(I+IWB)=0.
      W(I+IWC)=0.
      DO 42 J=1,N
      KDD=KDD+1
   42 W(I+IWB)=W(I+IWB)+W(KDD)*W(IWA+J)
C     REVISE THE DIRECTIONS IN THE ARRAY DD
      S=0.
      KK=N
   43 IF (W(KK+IWB)) 45,44,45
   44 KK=KK-1
      GO TO 43
   45 KK=KK-1
      IF (KK) 48,48,46
   46 S=S+W(IWB+KK+1)**2
      C=SQRT(S*(S+W(IWB+KK)**2))
      CA=S/C
      CB=W(KK+IWB)/C
      KDD=IDD+N*KK
      DO 47 J=1,N
      KDD=KDD+1
      W(J+IWC)=W(J+IWC)+W(IWB+KK+1)*W(KDD)
   47 W(KDD)=CA*W(KDD-N)-CB*W(J+IWC)
      GO TO 45
   48 KDD=IDD
      DO 49 I=2,N
      DO 49 J=1,N
      KDD=KDD+1
   49 W(KDD)=W(KDD+N)
      C=1./SQRT(DSQ)
      DO 50 I=1,N
      KDD=KDD+1
   50 W(KDD)=C*W(I+IWA)
C     APPLY THE CORRECTION VECTOR AND CALCULATE THE
C     OBJECTIVE FUNCTION
   51 DO 52 I=1,N
   52 W(I+IXA)=X(I)+W(I+IWA)
      MAXC=MAXC+1
C     NOTE THAT THE NEXT INSTRUCTION IS NOT STANDARD FORTRAN
      CALL CALCFG (N,W(IXA+1),FA,W(IGA+1))
C     SET THE ERROR OF THE PREDICTED GRADIENT IN WB
C     ALSO CALCULATE SOME NUMBERS FOR REVISING THE STEP-BOUND
      DG=0.
      DGA=0.
      DGGD=0.
      WBSQ=0.
      DO 54 I=1,N
      SUM=0.
      K=I
      J=1
  111 IF (J-I) 112,113,113
  112 SUM=SUM+W(K)*W(J+IWA)
      K=K+N-J
      J=J+1
      GO TO 111
  113 DO 53 J=I,N
```

```
233:          SUM=SUM+W(K)*W(J+IWA)
234:       53 K=K+1
235:          W(I+IWB)=W(I+IGA)-G(I)-SUM
236:          DG=DG+G(I)*W(I+IWA)
237:          DGA=DGA+W(I+IGA)*W(I+IWA)
238:          DGGD=DGGD+SUM*W(I+IWA)
239:       54 WBS0=WBS0+W(I+IWB)**2
240:C         TEST WHETHER TO DECREASE THE STEP-BOUND
241:          IF (ITSPEC-2) 55,60,60
242:       55 IF (FA-F-0.1*DG-0.05*DGGD) 57,57,56
243:       56 DSS=0.25*DSS
244:          GO TO 60
245:C         TEST WHETHER TO INCREASE THE STEP-BOUND
246:       57 DSS=DSS
247:          IF (WBS0-0.25*GSS) 59,59,58
248:       58 IF (DG-DGA-DGA) 60,59,59
249:       59 DSS=4.*DSS
250:C         SET THE DIFFERENCE BETWEEN GRADIENTS
251:       60 DO 61 I=1,N
252:       61 W(I+IWC)=W(I+IGA)-G(I)
253:C         SET X, F AND G TO THE BEST CALCULATED VALUES
254:          ITHETA=1
255:          IF (F-FA) 64,64,62
256:       62 F=FA
257:          DO 63 I=1,N
258:          X(I)=W(I+IXA)
259:       63 G(I)=W(I+IGA)
260:C         CALCULATE SOME VECTORS AND SCALAR PRODUCTS TO
261:C         REVISE GG AND H
262:       64 SDM=0.
263:          GHD=0.
264:          SEG=0.
265:          DHD=0.
266:          DO 66 I=1,N
267:          W(I+IXA)=-W(I+IWA)
268:          W(I+IGA)=0.
269:          K=IH+I
270:          J=1
271:      114 IF (J-I) 115,116,116
272:      115 W(I+IXA)=W(I+IXA)+W(K)*W(J+IWC)
273:          W(I+IGA)=W(I+IGA)+W(K)*W(J+IWA)
274:          K=K+N-J
275:          J=J+1
276:          GO TO 114
277:      116 DO 65 J=I,N
278:          W(I+IXA)=W(I+IXA)+W(K)*W(J+IWC)
279:          W(I+IGA)=W(I+IGA)+W(K)*W(J+IWA)
280:       65 K=K+1
281:          SDM=SDM+W(I+IWA)*W(I+IWB)
282:          GHD=GHD+W(I+IWC)*W(I+IGA)
283:          SEG=SEG+W(I+IWC)*W(I+IXA)
284:       66 DHD=DHD+W(I+IWA)*W(I+IGA)
285:C         TEST WHETHER THE USUAL CORRECTION TO GG GIVES
286:C         NEAR SINGULARITY
287:          HDIV=SEG*DHD-GHD*GHD
288:          GO TO (67,70),ITHETA
289:       67 IF (ABS(HDIV)-0.1*DSS*DSS) 68,70,70
290:C         CHANGE THE DIFFERENCE IN GRADIENTS TO AVOID SINGULARITY
```

```
291:    68 CA=HDIV/(DS..*DS..)+0.1
292:       CB=GHD/DS..-0.1
293:       CA=CA/(CA+CB+SIGN(S...((0.)*CA+CE*CB),CA+CB))
294:       CB=(1.-CA)*SDM/DS.
295:       DO 69 I=1,N
296:       C=CA*(CB*W(I+I..A)-W(I+I..B))
297:       W(I+I..B)=W(I+I..B)+C
298:    69 W(I+I..C)=W(I+I..C)+C
299:       ITHETA=2
300:       GO TO 64
301:C      REVISE THE MATRICES GG AND H
302:    70 CA=1./DSQ
303:       CB=SDM*CA*CA
304:       CC=DHD/HDIV
305:       CD=GHD/HDIV
306:       CE=SEG/HDIV
307:       K=0
308:       DO 71 I=1,N
309:       DO 71 J=I,N
310:       K=K+1
311:       W(K)=W(K)+CA*(W(I+I..A)*W(J+I..B)+W(J+I..A)*W(I+I..B))
312:      1-CB*W(I+I..A)*W(J+I..A)
313:    71 W(K+IH)=W(K+IH)-CC*W(I+I..A)*W(J+IGA)-CE*W(I+IGA)*W(J+IGA)
314:      1+CD*(W(I+IGA)*W(J+IGA)+W(J+IGA)*W(I+IGA))
315:       GO TO 8
316:       END
```