Computational experience with quadratically convergent minimisation methods

B. A. Murtagh and R. W. H. Sargent

Imperial College, London SW7

A recently reported minimisation method allows great flexibility in choosing successive steps without losing the property of quadratic convergence, but special precautions are necessary to ensure ultimate convergence from an arbitrary point for general functions. The paper makes an analysis of the required conditions, which give rise to several possible algorithms, and results of these for a number of problems are presented and discussed.

(Received March 1969)

1. Introduction

In a recent paper (1968), the authors have discussed a class of quadratically convergent minimisation methods which allow steps of arbitrary length and direction. The paper was directed to the problem of minimisation in the presence of constraints, using a variable metric projection operator, but recent numerical experience confirms that one of the methods discussed is in itself a powerful tool for unconstrained minimisation.

An initial approximation to the inverse of the Hessian matrix of second partial derivatives is updated recursively at each step. This approach is similar to Davidon's variable-metric method (1959) and the improved version of this due to Fletcher and Powell (1963), the particular matrix-updating scheme being only one of a number of possible alternatives. The scheme to be discussed here was first mentioned by Davidon (1959) and discussed more recently by Broyden (1967) and Davidon (1968) as well as the present authors (method 2 of the earlier paper, 1968).

Its attractiveness in constrained minimisation is obvious, as the presence of constraints and the restriction of search steps to subspaces which vary as the search continues do not mar its convergence properties. Its attractiveness as an alternative to the well-established method of Fletcher and Powell for unconstrained minimisation lies in the fact that its quadratic convergence properties for arbitrary steps obviate the need for successive minimisation along each search direction. This is likely to reduce the total number of function evaluations required, and in addition it is possible to choose the steps to ensure overall convergence for a rather general class of functions.

Whether or not accurate one-dimensional searches are necessary for the Fletcher-Powell method when it is applied to general functions seems to be a point of some contention in the literature, but in any case the method remains stable, with the approximating matrix finite and positive-definite. The present matrix updating scheme does not enjoy this stability. Indeed, there is no guarantee that the matrix will remain finite and positive-definite even for quadratic functions, so that remarks concerning quadratic convergence must be qualified by an assumption of stability.

The work reported here was therefore undertaken to discover whether instability was significant in practice, and to investigate various methods of dealing with it if it should arise. Further efforts have been made to establish, both theoretically and by numerical experimentation with a variety of problems, the most effective search algorithm, and comparisons with the Fletcher-Powell method are also given.

2. The matrix updating scheme

It is desirable to base minimisation methods on a local quadratic approximation to the function, since this should at least ensure fast ultimate convergence for most functions when the steps become small.

For quadratic functions, convergence is obtained in one step by the formula:

$$x - x_k = - H^{-1} g_k \tag{1}$$

where H is the Hessian matrix of the second partial-derivatives, and $g_k = g(x_k)$, the gradient at the current point x_k.

Changes in gradients are also related to changes in position by the Hessian matrix:

$$q_k = H p_k \tag{2}$$

where $q_k = g_k - g_{k-1}$ and $p_k = x_k - x_{k-1}$

The basis of the class of methods considered is to use these differences to build up information on the Hessian without calculating it explicitly. Thus, we start with an initial approximation, S_0, to the inverse of the Hessian and update it recursively so that

$$S_k q_j = p_j \quad j = 1, \ldots, k \tag{3}$$

There are few restrictions on the choice of steps p_k for updating the matrix, and indeed it is evident from (2) and (3) that for a quadratic function with non-singular H any set of n linearly independent steps spanning R^n space yield $S_n = H^{-1}$, and the minimum can thus be found from equation (1) at the $(n + 1)$th step. Obviously a lucky choice of S_0 can produce this result in less than n steps so it is logical to use the current approximation S_k in equation (1) to generate the steps. However this can give a very poor prediction of the position of the minimum, so we introduce an arbitrary scalar, α_{k-1}, to allow a variable step-length along the predicted direction, according to the equation:

$$p_k = -\alpha_{k-1} S_{k-1} g_{k-1} \tag{4}$$

It was shown previously (Murtagh and Sargent, 1968) that this choice of step produces the stronger result that a stationary value of a quadratic function is found at the latest at the $(n + 1)$th step (with $\alpha_n = 1$) even if H is singular.

In the method discussed in this paper, the matrix is updated by the symmetric formula:

$$S_k = S_{k-1} + z_k z_k^T / c_k \tag{5}$$

where
$$\left.\begin{aligned} z_k &= p_k - S_{k-1} q_k \\ c_k &= q_k^T z_k \end{aligned}\right\} \tag{6}$$

The Fletcher-Powell method can also be considered a member of this class of methods, with the recursion formula:

$$S_k = S_{k-1} + \frac{p_k p_k^T}{p_k^T q_k} - \frac{S_{k-1} q_k q_k^T S_{k-1}}{q_k^T S_{k-1} q_k} \tag{7}$$

Equations (5) and (6) generate matrices S_k satisfying (3) provided only that the function is adequately represented by a quadratic over the range of the steps involved, whereas equation (7) requires in addition that steps be generated according to (4) with α_{k-1} chosen so that the function is minimised with respect to α_{k-1} along the defined direction.

It is shown in Appendix 1 that if steps p_j are systematically made to the minimum along the line, all methods which generate symmetric S_k to satisfy (3) will generate directions which are mutually conjugate with respect to S_k^{-1}, and also to the Hessian matrix, H, when used on a quadratic function.

In Appendix 2 it is further shown that when such conjugate directions are generated for a quadratic function, using either of the above recursion formulae, both algorithms generate the same steps from the same initial matrix, S_0.

3. Conditions for convergence

Other authors (Fletcher and Powell, 1963; Broyden, 1967; Davidon, 1968) have emphasised the importance of stability, which in the context of minimisation methods means that the function must decrease at every step. However, as remarked by both Broyden and Davidon this is not of itself sufficient to ensure convergence to a minimum. The following theorem is a slight improvement on the convergence theorem given in our earlier paper (1968):

Theorem:

Suppose that the function $f(x)$ is defined on $U \subset E^n$ and is such that

(i) $f(x)$ is continuous on $\Omega = \{x | x \in U;\ f(x) \leqslant c\}$, and Ω is closed and bounded.

(ii) $f(x)$ has continuous second derivatives on $\Omega' = \{x | x \in U;\ f(x) < c\}$ and there is a Λ such that $||H(x)|| \leqslant \Lambda$, $x \in \Omega'$.

Starting at any point $x_0 \in \Omega'$ with $g(x_0) \neq 0$, we generate a sequence $x_0, x_1, \ldots, x_k, x_{k+1}, \ldots$ according to:

$$p_{k+1} = x_{k+1} - x_k = -\alpha_k S_k g_k \tag{8}$$

Then if the matrices S_k satisfy the conditions:

$$\rho ||g_k|| \leqslant ||S_k g_k|| \leqslant \sigma ||g_k|| \tag{9}$$

$$|g_k^T S_k g_k| \geqslant \delta ||g_k|| . ||S_k g_k|| \tag{10}$$

where ρ, σ, δ are fixed positive constants, it is always possible to choose a finite non-zero α_k at each step such that:

$$f(x_k) - f(x_{k+1}) \geqslant \epsilon \alpha_k g_k^T S_k g_k > 0 \tag{11}$$

with ϵ a fixed positive constant less than unity.

With the α_k so chosen, the sequence (x_k) lies in Ω' and tends to $\Omega^* = \{x | x \in \Omega';\ g(x) = 0\}$ in the sense that the distance $d(x_k, \Omega^*)$ of x_k from Ω^* tends to zero as $k \to \infty$.

This theorem is proved in Appendix 3. Of course it tells us nothing about the rate of convergence, but it is desirable that any procedure should satisfy its conditions so that ultimate convergence is assured for this general class of functions. Unfortunately, as we shall see, it is not always possible to do this and retain at the same time the property of quadratic convergence. It should also be noted that the theorem only guarantees convergence to a stationary point. We must therefore carry out a local search at any stationary point found, to check that it is indeed a minimum; if it is not, the local search will give a new starting point with a smaller function-value, and because of condition (11) the procedure cannot return to this non-minimum stationary point. We must therefore ultimately converge on a (weak) local minimum.

So long as the correction to S_{k-1} is finite ($c_k \neq 0$) we can always keep $||S_k g_k||$ within the bounds prescribed by (9) by multiplying S_k by an appropriate scalar if they are transgressed. Such scaling is seldom likely to be necessary but if it is it at least leaves unchanged the eigenvectors of S_k, and hence the orientation of its principal axes, so that second-order information is not entirely lost.

The upper bound on $||S_k g_k||$ serves only to ensure that condition (11) can be satisfied for a non-zero α_k and in practice it is more convenient to set a lower bound on α_k and test this directly. If the value of α_k required to satisfy condition (11) is below this bound, we can scale it up appropriately and scale down S_k by the same factor so that the step is unchanged.

Condition (11) provides stability, and when coupled with conditions (9) and (10) it ensures that the successive steps do not stop short of the stationary point. Condition (10) limits the angle between the search direction and the direction of steepest descent and so ensures an initial decrease of the function along the search direction for positive α_k; it is obviously satisfied for some δ if all S_k are positive-definite.

4. Maintenance of positive-definiteness

It turns out that it is possible to devise a very simple test to check whether S_k is positive-definite. For this purpose we consider the matrix:

$$S(t) = S_{k-1} + tz_k z_k^T/c_k \qquad (12)$$

Now a theorem given by Carathéodory (1967) shows that $S(t)$ is positive-definite in the range $0 \leqslant t \leqslant 1$ if S_{k-1} is positive-definite and $S(t)$ is non-singular over this range. From (12) we have:

$$\det|S(t)| = \det|S_{k-1}|.(1 + tz_k^T S_{k-1}^{-1} z_k/c_k) \qquad (13)$$

and using equations (4) and (6) this becomes:

$$\det|S(t)| = \det|S_{k-1}|.(1 - t - t\alpha_{k-1} z_k^T g_{k-1}/c_k) \qquad (13a)$$

Since α_{k-1} is positive it follows that S_k will be positive-definite if S_{k-1} is positive-definite and

$$z_k^T g_{k-1}/c_k < 0 \qquad (14)$$

Obviously S_0 can be chosen positive-definite, and it is then sufficient to satisfy condition (14) every time the matrix is updated.

To study the conditions under which (14) is likely to fail we examine separately the numerator and denominator, which with the aid of equations (4) and (6) may be written:

$$z_k^T g_{k-1} = (1 - \alpha_{k-1}) g_{k-1}^T S_{k-1} - g_k^T S_{k-1} g_{k-1} \qquad (15)$$

$$c_k = z_k^T q_k = (1 - \alpha_{k-1}) g_{k-1}^T S_{k-1} g_k - g_k^T S_{k-1} g_k - z_k^T g_{k-1} \qquad (16)$$

Using (15) to eliminate α_{k-1} from (16), we obtain after some rearrangement:

$$g_{k-1}^T S_{k-1} g_{k-1} . c_k = z_k^T g_{k-1} \{g_{k-1}^T S_{k-1} g_k - g_{k-1}^T S_{k-1} g_{k-1}\} - \{g_k^T S_{k-1} g_k . g_{k-1}^T S_{k-1} g_{k-1} - (g_{k-1}^T S_{k-1} g_k)^2\} \qquad (17)$$

Now S_{k-1} is positive definite, so $g_{k-1}^T S_{k-1} g_{k-1} > 0$, and from the Schwarz inequality:

$$g_k^T S_{k-1} g_k . g_{k-1}^T S_{k-1} g_{k-1} \geqslant (g_{k-1}^T S_{k-1} g_k)^2 \qquad (18)$$

with equality only if g_k is parallel to g_{k-1}

It also follows from (15) that if $z_k^T g_{k-1} > 0$, then $g_{k-1}^T S_{k-1} g_{k-1} > g_{k-1}^T S_{k-1} g_k$ and the first term on the right-hand side of (17) is negative.

Thus if $z_k^T g_{k-1} > 0$, it follows that $c_k < 0$ and condition (14) is satisfied. If this occurs for $\alpha_{k-1} \geqslant 1$, equation (15) shows that $p_k^T g_k > 0$ and hence a minimum must occur along the line between x_{k-1} and x_k.

If $c_k > 0$ it follows that $z_k^T g_{k-1} < 0$ and again condition (14) is satisfied. In this case (15) shows that if $0 < \alpha_{k-1} \leqslant 1$ we have $p_k^T g_k < 0$ and the function is decreasing at x_k; unless the function is quite badly behaved it will not have both a maximum and a minimum in this range of α_{k-1} so that this normally implies that $f(x_k) < f(x_{k-1})$.

It is unfortunately possible to have $c_k \leqslant 0$ and $z_k^T g_{k-1} \leqslant 0$ together (although both can be simultaneously zero only if g_k is parallel to g_{k-1}), so that condition (14) can fail. If this failure occurs at $\alpha_{k-1} = 1$, then we have $p_k^T g_k \leqslant 0$ and we shall normally have obtained a function decrease, as explained above. If a stationary point occurs at $\alpha_{k-1} = 1$ (i.e. $p_k^T g_k = 0$) this will normally be a minimum, correctly predicted using S_{k-1}, so it would seem reasonable simply to set $S_k = S_{k-1}$.

We note also that condition (14) is not guaranteed at a stationary point along the line unless this occurs for $0 < \alpha_{k-1} < 1$, in which case $z_k^T g_{k-1} > 0$ and $c_k < 0$.

From the mean-value theorem we can write:

$$q_k = H(x_{k-1} + \theta p_k).p_k, \; 0 \leqslant \theta \leqslant 1 \qquad (19)$$

Then substituting (19) into (15) and (16) and leaving the argument of H understood, we obtain:

$$\left.\begin{aligned} z_k^T g_{k-1} &= \alpha_{k-1} . g_{k-1}^T (S_{k-1} H S_{k-1} - S_{k-1}) g_{k-1} \\ c_k = z_k^T q_k &= \alpha_{k-1} . g_{k-1}^T (S_{k-1} H S_{k-1} - S_{k-1}) H S_{k-1} g_{k-1} \end{aligned}\right\} \qquad (20)$$

From these equations it is evident that changing α_{k-1} is unlikely to have much effect on the signs of $z_k^T g_{k-1}$ and c_k, unless H is changing fairly rapidly along the search direction, in which case quadratic fitting is unlikely to be very useful. It therefore seems reasonable to use the choice of α_{k-1} to achieve a function decrease and satisfy equation (11), then if c_k is zero or condition (14) not satisfied at the resulting point it is probably best to abandon the quadratic updating formula for this step.

However, S_{k-1} is positive definite and contains all the earlier information accumulated on local quadratic behaviour, so that it would seem worth trying $S_k = S_{k-1}$ in this situation. Unfortunately numerical experience shows that in many cases this rule leads to frequent subsequent failure of the conditions. However there is still a possibility of retaining the earlier information in S_{k-1} by updating to S_k using equation (5), but with a different choice of c_k.

That this does retain earlier quadratic information in S_{k-1} is easily proved:

Suppose that the local quadratic approximation of equation (2) is valid and that:

$$p_j = S_{k-1} q_j \text{ for some } j$$

Then from (2) and the above:

$$q_k^T S_{k-1} q_j = q_k^T p_j = p_k^T H p_j = p_k^T q_j$$

and from the definition of z_k in (6):

$$z_k^T q_j = p_k^T q_j - q_k^T S_{k-1} q_j = 0$$

Thus from (5):

$$S_k q_j = S_{k-1} q_j = p_j$$

Of course equation (13a) does not hold for a general c_k so that (14) is no longer a condition for positive-definiteness, but it follows directly from (5) that S_k is positive-definite for any positive c_k. Since z_k rapidly approaches zero as the minimum is approached it is important to keep c_k in scale so that a significant correction to S_{k-1} is made, and the obvious choice is $c_k = z_k^T z_k$. Several other choices were tried numerically, but this choice in fact proved the most successful.

If even this device fails, there is nothing for it but to start afresh with a new arbitrary matrix which does satisfy all the conditions, and the identity matrix is the obvious choice.

This problem of scale in the correction to the matrix is also relevant to testing whether c_k is too close to zero, and in practice the condition tested was:

$$|c_k| \geqslant \delta . z_k^T z_k \qquad (21)$$

Davidon's recent 'variance algorithm' (1968) uses the same matrix recursion formula, and he also chooses a new c_k if the value given by equation (6) fails to keep S_k positive-definite. However his algorithm uses a fixed step-length with $\alpha_{k-1} = 1$ in equation (4); if this step gives a function increase the matrix is still updated but the step is rejected, starting the next step again from x_{k-1} but using the updated S_k.

It is of interest to compare the behaviour of the Davidon-Fletcher-Powell recursion formula (Colville, 1968) with the above results. Proceeding analogously to Fletcher and Powell (1963), we have

$$x^T S_k x = x^T S_{k-1} x + \frac{(p_k^T x)^2}{p_k^T q_k} - \frac{(q_k^T S_{k-1} x)^2}{q_k^T S_{k-1} q_k}$$

whence:

$$x^T S_k \, x \geqslant \frac{(p_k^T x)^2}{p_k^T q_k} \qquad (22)$$

on account of the Schwarz inequality, and the equality occurs only if x is parallel to q_k. If $p_k^T q_k > 0$, x cannot be simultaneously parallel to q_k and orthogonal to p_k, so it follows from (22) that this is a sufficient condition for S_k to be positive-definite.

This condition is the analogue of condition (14), and using equation (4) it may be written:

$$p_k^T q_k = \alpha_{k-1}(g_{k-1}^T S_{k-1} g_{k-1} - g_{k-1}^T S_{k-1} g_k) > 0 \qquad (23)$$

It is always satisfied if we step to the minimum, for then the second term vanishes. Otherwise the condition may fail, but unless the function is badly behaved the quantity $g_{k-1}^T S_{k-1} g_k$ will normally decrease from $g_{k-1}^T S_{k-1} g_{k-1}$ as α_{k-1} increases from zero, becoming negative as we pass through the minimum; the condition is therefore certainly satisfied for a range of α_{k-1} beyond the minimum, and in most cases for all positive values of α_{k-1}.

If condition (23) is satisfied and S_{k-1} is positive-definite, we note that both denominators in equation (7) are non-zero, so the correction to S_{k-1} is finite. Of course the Davidon-Fletcher-Powell method may also fail on conditions (9) and (10), but this is an unlikely occurrence as for the method described above.

5. Choice of step-length

The desirability of generating conjugate directions has been much discussed, and it has been argued that this guarantees that the space is fully spanned. However this is not necessary to ensure convergence, nor is it sufficient even if the function decreases at each step, and the necessity of determining the minimum accurately along each search direction in any case seems likely to increase the required number of function evaluations. On the other hand, minimisation does give the largest possible function decrease along each direction, and it turns out that condition (11), which does ensure convergence if the S_k are positive-definite and bounded, will in all probability be satisfied at the minimum, as the following analysis indicates:

Suppose that p_{k+1} steps from the point x_k to the minimum along the line at point x_{k+1}. Then a Taylor expansion about the point x_k gives:

$$f(x_{k+1}) = f(x_k) + g_k^T p_{k+1} + \tfrac{1}{2} p_{k+1}^T . H(x_k + \theta p_{k+1}) . p_{k+1}$$
$$\text{where } 0 \leqslant \theta \leqslant 1 \qquad (24a)$$

On the other hand, a Taylor expansion about the point x_{k+1} gives:

$$f(x_k) = f(x_{k+1}) + \tfrac{1}{2} p_{k+1}^T . H(x_k + \phi p_{k+1}) . p_{k+1}$$
$$\text{where } 0 \leqslant \phi \leqslant 1 \qquad (24b)$$

From (24a) and (24b), and using equation (4):

$$f(x_k) - f(x_{k+1}) = \tfrac{1}{2} \alpha_k g_k^T S_k g_k + \tfrac{1}{4} p_{k+1}^T \{H(x_k + \phi p_{k+1}) - H(x_k + \theta p_{k+1})\} p_{k+1} \qquad (25)$$

This is true for any method using steps given by equation (4) no matter how S_k is generated, and if ϵ in condition (11) is appreciably less than $\frac{1}{2}$ the function would have to be very badly behaved for the second term in (25) to swamp the first term and cause the condition to fail. In general therefore condition (11) is a much less stringent requirement than going to the minimum along the line.

Since the recursion formula defined by equations (5) and (6) gives quadratic convergence for any step, and will in addition generate conjugate directions if minimisation is used, we can use it to make a direct numerical test of the desirability of the latter without extraneous complications. In addition, comparison of results for this conjugate-direction method with those for the Fletcher-Powell method enables us to compare the relative effectiveness of the two matrix recursion formulae.

One would expect the Fletcher-Powell method to show to advantage in this comparison, as the conditions for use of the quadratic updating formula in our own method are likely to fail more often. The results of the last section indicate one way in which some fruitless computation might be avoided by early detection of the likely failure of these conditions. It was shown that if $z_k^T g_{k-1} > 0$ at $\alpha_{k-1} = 1$, then the function will have a minimum in the range $0 < \alpha_{k-1} < 1$ and the conditions will also be satisfied at this minimum. Conversely, if $z_k^T g_{k-1} \leqslant 0$ at $\alpha_{k-1} = 1$, the function will normally have decreased and the minimum occurs at $\alpha_{k-1} \geqslant 1$; if then $c_k \leqslant 0$ condition (14) fails, and since a change of α_{k-1} is unlikely to change the conditions much, it will probably also fail at the minimum. This suggests an initial step with $\alpha_{k-1} = 1$ and testing $z_k^T g_{k-1} > 0$ there. If this is satisfied one can go to the minimum with the assurance that the matrix can be updated there. Otherwise there has been a function decrease and the step provides an improved new starting point: if $c_k > 0$ the quadratic updating formula can be used and otherwise the matrix is reset.

In these methods the minimisation along the line can be carried out in various ways. In our earlier paper (1968) a quadratic interpolation method was suggested which used the initial function and gradient values but required only function evaluations in the search; Davidon (1959) and Fletcher and Powell (1963) on the other hand suggested cubic interpolation requiring both function and gradient evaluation at each point in the search. Preliminary numerical experiments showed that the quadratic method was relatively inefficient and it was well worth accepting the gradient evaluations required for the cubic method, especially on steep-sided valleys.

It was shown earlier that the choice of α_{k-1} has little effect on the conditions for using the quadratic updating formula, and that condition (11) for convergence will usually be rather easy to satisfy. Minimisation certainly produces the largest possible function decrease for the given direction, but since accuracy in locating the minimum is no longer necessary for quadratic convergence it seems reasonable to be content with achieving merely the greater part of this possible function decrease. In view of the efficiency of the cubic minimisation algorithm, one can revert essentially to the idea of Davidon's original variable-metric algorithm (1959), which simply used one iteration, and carry out only as many iterations as required to satisfy condition (11). In general this is indeed likely to be only one iteration, but if we do approach the minimum we have seen that we are reasonably sure of satisfying condition (11) there. For difficult functions even this may fail, but we know that condition (11) must be satisfied for sufficiently small α_k so that if necessary we can continue with successive halving of α_k until condition (11) is satisfied; it may be said at once that this has not been found necessary in any numerical test made so far.

Often an initial step of α_{k-1} can be very large and lead to a function increase. Fletcher and Powell (1963) noted this difficulty and suggested an alternative choice of α_{k-1} in this situation, basing it on equation (25) assuming a quadratic function and an estimate of the function's minimum value. This leads to the initial choice:

$$\alpha_{k-1} = \min\{1, 2[f(x_{k-1}) - f_{L.B.}]/g_{k-1}^T S_{k-1} g_{k-1}\} \quad (26)$$

where $f_{L.B.}$ is the estimated lower bound on $f(x)$. In many cases $f_{L.B.}$ can be taken as zero. Again numerical experiments were carried out to test the possible advantages of this initial choice.

6. Algorithms tested

The explicit procedures for a single step of each of the algorithms tested are as follows:

Algorithm 1 (First conjugate-direction method)

(i) Make an initial step according to equation (4) with α_{k-1} given by equation (26).
(ii) Use cubic minimisation to find the minimum.
(iii) If $||g_k|| \leqslant \delta''$ proceed to the local search for confirmation of a minimum.
(iv) Test $\alpha_{k-1} \geqslant \sigma'$ and $||S_{k-1}g_{k-1}||/||g_{k-1}|| \geqslant \rho$; rescale α_{k-1} and S_{k-1} if necessary.
(v) Test $|c_k| \geqslant \delta''.z_k^T z_k$ and $z_k^T g_{k-1}/c_k \leqslant -\delta'$; if either fails reset S_k, but otherwise update S_{k-1} to S_k using equations (5) and (6).

Algorithm 2 (Second conjugate-direction method)

(i) Make an initial step according to equation (4) with $\alpha_{k-1} = 1$.
(ii) If $||g_k|| \leqslant \delta''$ proceed to the local search for confirmation of a minimum.
(iii) If either $z_k^T g_{k-1} > \delta'$ or the function has increased go to step (iv), otherwise test $|c_k| \geqslant \delta'' z_k^T z_k$ and $z_k^T g_{k-1}/c_k \leqslant -\delta'$; if either fails reset S_k and return to (i) for the next step, but otherwise go to step (vi).
(iv) Use cubic minimisation to find the minimum.
(v) If $||g_k|| \leqslant \delta''$ proceed to the local search for confirmation of a minimum.
(vi) Test $\alpha_{k-1} \geqslant \sigma'$ and $||S_{k-1}g_{k-1}||/||g_{k-1}|| \geqslant \rho$; rescale α_{k-1} and S_{k-1} if necessary.
(vii) Update S_{k-1} to S_k using equations (5) and (6).

Algorithm 2*a*

As Algorithm 2 but with the initial α_{k-1} in (i) given by equation (26). In this case cubic minimisation may generate an $\alpha_{k-1} \geqslant 1$ and if this occurs we set $\alpha_{k-1} = 1$ and return to (iii).

Algorithm 3 (Ensuring convergence)

(i) Set $\alpha_{k-1} = 1$.
(ii) Make a step according to equation (4).
(iii) If $||g_k|| \leqslant \delta''$ proceed to the local search for confirmation of the minimum.
(iv) Test condition (11). If not satisfied make one step of the minimisation algorithm, or if this has converged halve α_{k-1}, and return to (ii).
(v) Test $\alpha_{k-1} \geqslant \sigma'$ and $||S_{k-1}g_{k-1}||/||g_{k-1}|| \geqslant \rho$; rescale α_{k-1} and S_{k-1} if necessary.
(vi) Test $|c_k| \geqslant \delta'' z_k^T z_k$ and $z_k^T g_{k-1}/c_k \leqslant -\delta'$; if either fails reset S_k, but otherwise update S_{k-1} to S_k using equations (5) and (6).

Algorithm 3*a*

As Algorithm 3, but with the initial α_{k-1} in (i) given by equation (26).

All these algorithms were tested with two alternative methods of resetting S_k when the tests for positive-definiteness failed, as follows:

Reset 1. Set $S_k = I$.
Reset 2. Update S_{k-1} to S_k using equation (5) with $c_k = z_k^T z_k$.

In the various tests δ' is a small positive constant used to avoid problems with rounding errors or accumulator overflow. The constants δ', σ', ρ', and ϵ were all set to 10^{-8}.

In the one-dimensional searches for a minimum along the line, convergence was assumed when either $|g_k^T p_k| \leqslant \delta''$ or $\Delta\alpha.||S_{k-1}g_{k-1}|| \leqslant \delta''$ where $\Delta\alpha$ is the change in α_{k-1} for one iteration of the search algorithm. For overall convergence $||g_k|| \leqslant \delta''$ was used, and in all the convergence tests δ'' was set to 10^{-4}.

We were also interested in the extent to which positive-definiteness of S_k ensured that condition (10) was satisfied, and therefore tested this condition, with $\delta = 10^{-4}$, at the same time as $|c_k| \geqslant \delta'' z_k^T z_k$ and $z_k^T g_{k-1}/c_k \leqslant -\delta'$ (i.e. condition (14)). In nearly all cases however condition (10) and condition (14) succeeded or failed together, in spite of the higher value of δ, so that it is unnecessary to incorporate condition (10) as an additional test in the algorithms.

In the first four examples given below, results were also obtained for the Fletcher-Powell algorithm, this being identical to Algorithm 1 in all respects except step (v) in which S_{k-1} was updated to S_k using equation (7); no test is necessary as the function is minimised along the line. The results given for the Fletcher-Powell algorithm for Example 5 are those quoted in their paper (1963), and may therefore have used different criteria for termination; they were probably also obtained with single-precision working.

All other calculations were made on an IBM 7094 using FORTRAN double-precision arithmetic, except for some additional results in single-precision for Algorithm 3 to test its sensitivity to rounding errors.

7. Discussion of numerical results

For each of the examples there is a table to show the number of function evaluations required to achieve convergence for the different algorithms. For the first four examples additional tables are given to compare the progress of Algorithms 3 and 3a, in each case using Reset 2.

Example 1. Rosenbrock's valley (Fletcher and Powell, 1963)

$$\text{minimise } f(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$$

starting at $(-1\cdot 2, 1\cdot 0)$

The performance on this problem is shown in **Table 1.**

Table 1

ALGORITHM	RESET 1	RESET 2
1	93	96
2	113	93
2a	94	73
3	38	60
3 (S.P.)	38	60
3a	64	60

Fletcher-Powell: 80.

Algorithm 3 with Reset 1 is far superior to any others in this example. Otherwise Reset 2 is in general better than Reset 1. Algorithm 2a is better than Algorithm 2, but the opposite is true for 3 and 3a; this is strange because the choice $\alpha = 1$ is obviously far too large in the early steps, but Algorithm 3 nevertheless descends faster than 3a after the second step. Single-precision working had no effect on the results for Algorithm 3. The Fletcher-Powell method gives results similar to Algorithm 1 and Algorithm 2a, but none of them are as good as 3 and 3a with either Reset 1 or Reset 2.

Example 2. Quartic with singular Hessian (Fletcher and Powell, 1963)

$$\begin{aligned}\text{minimise } f(x) &= (x_1 + 10x_2)^2 \\ &+ 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 \\ &+ 10(x_1 - x_4)^4\end{aligned}$$

starting at (3, −1, 0, 1)

The performance on this problem is shown in **Table 2.**

Table 2

ALGORITHM	RESET 1	RESET 2
1	52	51
2	46	46
2a	36	36
3	45	38
3 (S.P.)	45	38
3a	32	40

Fletcher-Powell: 64.

One would expect this problem to be a severe test for the methods because of the singularity of the Hessian matrix at the minimum point, and the behaviour is indeed atypical. The minimum value of the function is again zero, but for the same termination criterion on $||g_k||$, the minimum value is not as closely approached as in the other problems because of the flatness of the minimum.

The inverse matrix grows large as the minimum is approached, but even in this extreme case no scaling of the S_k was necessary. However improvement becomes slow close to the minimum, and this is particularly evident in the results for Algorithm 3a. A less stringent termination condition would in fact give results much more in line with those for the other problems.

The Fletcher-Powell method was worse than any of the other conjugate-direction methods (1, 2, 2a), although the minimisation along the search directions probably overcomes some of the problems of the poor quadratic approximation at the minimum and accounts for the relatively good results of Algorithms 2 and 2a.

Curiously, single-precision working did not affect Algorithm 3, although very small differences are involved as the minimum is approached.

Example 3. Helical valley (Fletcher and Powell, 1963)

$$\begin{aligned}\text{minimise } f(x) &= 100[(x_3 - 100)^2 \\ &+ (r - 1)^2] + x^2\end{aligned}$$

$$\begin{aligned}\text{where} \quad 2\pi\theta &= \tan^{-1}(x_2/x_1) \\ &- \pi/2 < 2\pi\theta < 3\pi/2\end{aligned}$$

$$\text{and} \quad r = (x_1^2 + x_2^2)^{1/2}$$

starting at (−1, 0, 0)

The performance on this problem is shown in **Table 3.**

Table 3

ALGORITHM	RESET 1	RESET 2
1	82	75
2	75	73
2a	64	62
3	54	41
3 (S.P.)	54	41
3a	35	33

Fletcher-Powell: 81.

Here Reset 2 shows a clear advantage over Reset 1 in all cases. Again 2a and 3a perform appreciably better than 2 and 3 respectively. The Fletcher-Powell method is comparable with Algorithm 1, but 2 and 2a are significantly better, and Algorithms 3 and 3a much better still. There is still no effect of single-precision working on Algorithm 3.

Example 4. Four-dimensional banana (Colville, 1968)

$$\text{minimise}\, f(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 + 90(x_3^2 - x_4)^2 + (1 - x_3)^2 + 10\cdot 1\ [(x_2 - 1)^2 + (x_4 - 1)^2] + 19\cdot 8(x_2 - 1)(x_4 - 1)$$

starting at $(-3, -1, -3, -1)$

The performance on this problem is shown in **Table 4**.

Table 4

ALGORITHM	RESET 1	RESET 2
1	145	153
2	193	129
2a	175	113
3	70	68
3 (S.P.)	88	69
3a	230	130

Fletcher-Powell: 257.

This is an interesting set of results. What is striking is the poor performance of all methods using equation (26) for the initial step, and particularly the Fletcher-Powell method. As in Example 1 this occurs in spite of the fact that many function evaluations are used in the first few steps where $\alpha = 1$ gives too large a step-length, but the function-value for Algorithm 3 nevertheless descends faster than for 3a. The exceptionally poor performance of the Fletcher-Powell method is inexplicable. Algorithm 3 gives clearly the best result, but in this example there is evidence of rounding-error effects from single-precision working, especially with the use of Reset 1. Again the general trend is for Reset 2 to perform better than Reset 1.

Results for this problem in its original context as a constrained problem with bounds of $\pm 10\cdot 0$ on the variables (which are usually inactive) have been quoted at 142 and 114 function evaluations for the Fletcher-Powell scheme (Colville, 1968). A corresponding result for Algorithm 3 with Reset 2 is 43 function evaluations, the decrease from 68 quoted in Table 4 being due to the bounds limiting the initial step-length; also, one constraint was active for two early steps.

Example 5. Trigonometric Functions in 5, 10, and 20 variables

$$\text{minimise}\, f(x) = \sum_{j=1}^{n} [E_i - \sum_{j=1}^{n} (A_{ij} \sin x_j + B_{ij} \cos x_j)]^2$$

i.e., solve the set of simultaneous non-linear equations

$$\sum_{=1}^{n} (A_{ij} \sin x_j + B_{ij} \cos x_j) = E_i \qquad i = 1, \ldots, n$$

The matrix elements of A and B were generated as random integers between ± 100, and the values of the variables, x_i $i = 1, \ldots, n$, were generated randomly between $\pm\pi$. For these values the right-hand sides, E_i, were calculated, and the starting point given as $(x_i + 0\cdot 1\delta_i)$ where the δ_i's were also generated as random numbers between $\pm\pi$. The experiments were repeated a number of times for $n = 5$, 10 and 20.

The performance on these functions is shown in **Table 5**.

In these examples the difference between Reset 1 and Reset 2 is less clear-cut; Reset 1 even seems generally slightly better, especially for Algorithm 1, although Reset 2 is still better for Algorithm 3a. Algorithms 2a and 3a again give better results than 2 and 3 respectively. The Fletcher-Powell method is clearly the best of the conjugate-direction methods. Curiously Algorithm 1 on the whole performs better than 2 or 2a, and even compares favourably with Algorithm 3; however Algorithm 3a with Reset 2 is still the best overall method.

An interesting phenomenon observed in these examples was that progress often went in cycles of around n steps; n steps with small function decrease would be made, followed by one or more steps with large function decrease. Often a reset was then required. Presumably, the n steps would build up curvature information, producing a large improvement and leading to a region where this information becomes invalid.

Table 5

n	FLETCHER POWELL	RESET 1					RESET 2				
		1	2	2a	3	3a	1	2	2a	3	3a
5	19	23	42	27	33	15	27	46	28	34	15
	23	25	51	31	37	19	33	52	30	39	17
10	29	45	68	52	56	49	49	83	43	58	30
	36	51	87	70	63	67	62	89	49	78	44
20	68	78	136	75	104	79	84	141	81	103	61
	84	80	140	81	112	90	95	142	85	104	65
	89	82	142	92	150	93	96	164	89	156	70
	121	95	177	132	151	141	97	188	97	168	83

8. Conclusions

Perhaps the most significant thing about all the results is their general similarity, although there is sufficient variation to confirm the usual observation that no numerical method can be expected to perform uniformly on all problems.

No rescaling of the S_k, either up or down, was necessary at any time with any of the methods, even including Example 2 which had a singular Hessian matrix at the minimum point. However the tests are simple, and probably worth keeping as a safeguard.

Reset 2 proved to be generally better than Reset 1, as would be expected since only the current step is rejected with Reset 2 rather than the whole matrix as in Reset 1. This would be of special value in constrained minimisation using a projection method since resetting $S_k = I$ would necessitate reforming and reinverting the inverse moment matrix of the projection operator (Murtagh and Sargent, 1968). The need for resetting did not arise as often as might have been expected, and when it did it was usually due to failure of condition (14) for positive-definiteness, rather than c_k being too close to zero. Positive-definiteness is a useful property to retain, especially if the inverse of the Hessian matrix at the minimum point is of interest, and as pointed out earlier the numerical results show that it is not more difficult to achieve than satisfying condition (9).

The improved performance expected for Algorithms 2 and 2a over Algorithm 1 with respect to the need for resetting is fully confirmed in the results (for example Algorithms 2 and 2a needed no reset at all in Example 2). Of the three comparable conjugate-direction methods Algorithm 2a was generally the best, with Algorithm 1 usually giving a slight improvement over the Fletcher-Powell method; however this general trend is reversed for Example 5.

The choice of initial step-lengths provided in Algorithms 1, 2a and 3a in general produced the expected result of saving a significant number of function evaluations; that it produced the opposite result for Algorithms 3 and 3a in Examples 1 and 4 is interesting and provides some food for thought. One possibility is that the large step-length oversteps the minimum and hence on these 'curving-valley' functions zig-zags easily round the bend, whilst the smaller step-length tends to creep round on one side of the valley. The fact that this phenomenon is not evident in comparing Algorithms 2 and 2a on these examples is probably because an initial step which overshoots the minimum results in minimisation along the line in each case.

Comparison of Algorithm 2 with 3, or 1 and 2a with 3a, shows a clear superiority for the methods which do not find the minimum accurately along each direction, the only exception being that Algorithm 3a does not come up to expectation in Example 4. Algorithm 3 is almost completely insensitive to rounding errors and the single-precision version performed very satisfactorily; it is fairly safe to conclude that this would also be true of Algorithm 3a.

Algorithm 3a using Reset 2 combines all the best characteristics and gives the best all-round performance; it is markedly superior to the classical Fletcher-Powell method on all the examples tested. However, if it is known that the function is likely to exhibit a curving valley, as often occurs for example in penalty function formulations of problems with non-linear constraints, it is probably worth suppressing the choice of initial step-length and using Algorithm 3.

Appendix 1

Conjugate directions

Consider the class of methods which generate steps to satisfy the following relations:

$$p_{j+1} = -\alpha_j S_j g_j \qquad 1 \leqslant j \leqslant k \qquad (1.1)$$

with α_j chosen so that:

$$p^I_{j+1} g_{j+1} = 0 \qquad 1 \leqslant j \leqslant k \qquad (1.2)$$

and where the S_j are non-singular symmetric matrices which satisfy:

$$S_j q_i = p_i \qquad 1 \leqslant i \leqslant j \leqslant k \qquad (1.3)$$

We shall prove that the successive directions, p_j, are mutually conjugate with respect to the inverse of the current matrix, S_k.

Proof:

From (1.3) and the fact that S_k is symmetric we have:

$$p_i^T S_k^{-1} p_j = p_i^T q_j = q_i^T p_j = q_i^T S_k q_j \qquad 1 \leqslant i \leqslant j \leqslant k \qquad (1.4)$$

Now from (1.1):

$$q_i^T p_j = -\alpha_{j-1} q_i^T S_{j-1} g_{j-1} \qquad 1 \leqslant i \leqslant j \leqslant k$$

using (1.3)

$$= -\alpha_{j-1} p_i^T g_{j-1} \qquad 1 \leqslant i < j \leqslant k$$

$$= -\alpha_{j-1} p_i^T g_{j-2} - \alpha_{j-1} p_i^T q_{j-1} \qquad 1 \leqslant i < j \leqslant k$$

using (1.3) again

$$= -\alpha_{j-1} q_i^T S_{j-2} g_{j-2} - \alpha_{j-1} p_i^T q_{j-1} \qquad 1 \leqslant i < (j-1) \leqslant k$$

Now if $\alpha_{j-2} \neq 0$ we can use (1.1) to obtain:

$$q_i^T p_j = \frac{\alpha_{j-1}}{\alpha_{j-2}} q_i^T p_{j-1} - \alpha_{j-1} p_i^T q_{j-1} \qquad 1 \leqslant i < (j-1) \leqslant k$$

and finally from (1.4):

$$p_i^T S_k^{-1} p_j = \alpha_{j-1} \left(\frac{1 - \alpha_{j-2}}{\alpha_{j-2}} \right) p_i^T S_{k-1}^{-1} p_{j-1} \qquad 1 \leqslant i < (j-1) \leqslant k \qquad (1.5)$$

But we also have from (1.1):

$$q_{j-1}^T p_j = -\alpha_{j-1} q_{j-1}^T S_{j-1} g_{j-1} \qquad 1 < j \leqslant k$$

and using (1.3):

$$= -\alpha_{j-1} p_{j-1}^T g_{j-1} \qquad 1 < j \leqslant k$$

and using (1.2) and (1.4):

$$p_{j-1}^T S_k^{-1} p_j = 0 \qquad 1 < j \leqslant k \qquad (1.6)$$

Thus (1.5) and (1.6) together show that if $p_i^T S_k^{-1} p_{j-1} = 0$ for $1 \leqslant i < (j-1) \leqslant k$, then $p_i^T S_k^{-1} p_j = 0$ for $1 \leqslant i < j \leqslant k$. But putting $j = 2$ in (1.6) shows that the premise is true for $(j-1) = 2$, and so the result is true generally.

Q.E.D.

The above proof assumes that successive α_j are non-zero, and we can show that this is necessarily so if the S_j are positive-definite. For from (1.1), (1.2) and (1.3):

$$q_{j+1}^T p_{j+1} = q_{j+1}^T S_{j+1} q_{j+1} = - g_j^T p_{j+1} = \alpha_j g_j^T S_j g_j$$

Thus:

$$\alpha_j = \frac{q_{j+1}^T S_{j+1} q_{j+1}}{g_j^T S_j g_j} \neq 0$$

Q.E.D.

We note from (1.4) that the successive q_j are also mutually conjugate with respect to S_k. If further the function being minimised is quadratic (with Hessian matrix H) we have:

$$q_j = Hp_j \qquad 1 \leqslant j \leqslant k \qquad (1.7)$$

Then from (1.4) and (1.7):

$$p_i^T S_k^{-1} p_j = p_i^T q_j = q_i^T p_j = p_i^T H p_j \qquad (1.8)$$

Hence the p_i are also mutually conjugate with respect to H.

Finally we have, from (1.1), (1.2) and (1.7):

$$q_{j+1}^T p_{j+1} = p_{j+1}^T H p_{j+1} = \alpha_j g_j^T S_j g_j$$

and α_j is therefore non-zero if H is positive-definite, as it must be for a minimum, whether the S_j are positive-definite or not.

Appendix 2

Search directions for conjugate-direction methods

Myers (1968) has shown that the Fletcher-Powell method generates the same set of search directions as various other conjugate-direction methods when these are used on a quadratic function, so long as the initial search direction is along the gradient at the initial point. We shall show similarly that the Fletcher-Powell method and the method of this paper with minimisation along the line generate the same search directions for a quadratic function, provided only that the initial matrices are proportional.

Present Method

Using equations (4), (5) and (6), and the fact that α_{j-1} is chosen so that:

$$g_j^T p_j = 0 \qquad (2.1)$$

we obtain:

$$S_j g_j = \frac{\alpha_{j-1} - 1}{c_j} (g_j^T S_{j-1} g_j . S_{j-1} g_{j-1} + g_{j-1}^T S_{j-1} g_{j-1} . S_{j-1} g_j) \qquad (2.2)$$

Fletcher-Powell Method

To avoid confusion we shall here denote the approximation to the inverse of the Hessian matrix by H_j for the Fletcher-Powell method. Then similarly from (4), (7) and (2.1) we obtain:

$$H_j g_j = \frac{1}{q_j^T H_{j-1} q_j} (g_j^T H_{j-1} g_j . H_{j-1} g_{j-1} + g_{j-1}^T H_{j-1} g_{j-1} . H_{j-1} g_j) \qquad (2.3)$$

Now we showed in Appendix 1 that both methods generate conjugate directions, and hence that:

$$q_k^T p_j = 0 \qquad k > j \geqslant 1 \qquad (2.4)$$

Thus from (2.1) and (2.4):

$$g_k^T p_j = 0 \qquad k \geqslant j \geqslant 1 \qquad (2.5)$$

and using equation (4); since $\alpha_j \neq 0$ (cf. Appendix 1):

$$g_k^T S_j g_j = g_k^T H_j g_j = 0 \qquad k > j \geqslant 1 \qquad (2.6)$$

Thus from (2.2), and (2.6), so long as $\alpha_{k-1} \neq 1$:

$$g_k^T S_{j-1} g_j = 0 \qquad k > j \geqslant 1 \qquad (2.7)$$

From (6), (2.5) and (2.7) we have:

$$g_k^T z_j = 0 \qquad k > j \geqslant 1 \qquad (2.8)$$

and finally using this in (5) we obtain:

$$S_j g_k = S_0 g_k \qquad k > j \geqslant 1 \qquad (2.9)$$

Similarly from (2.3) and (2.6) we have:

$$g_k^T H_{j-1} g_j = 0 \qquad k > j \geqslant 1 \qquad (2.10)$$

and using (2.5) and (2.10) in (7):

$$H_j g_k = H_0 g_k \qquad (2.11)$$

Thus if $S_0 = \beta_0 H_0$, where β_0 is a non-zero scalar, we have from (2.9) and (2.11):

$$S_j g_k = \beta_0 H_j g_k \qquad k > j \geqslant 1 \qquad (2.12)$$

and of course $S_0 g_0 = \beta_0 H_0 g_0$. But at any stage, if we also have $S_{k-1} g_{k-1} = \beta_{k-1} H_{k-1} g_{k-1}$, where β_{k-1} is again a non-zero scalar, we have from (2.2) and (2.12):

$$S_k g_k = \frac{\beta_0 \beta_{k-1}(\alpha_{k-1} - 1)}{c_k} (g_k^T H_{k-1} g_k . H_{k-1} g_{k-1} + g_{k-1}^T H_{k-1} g_{k-1} . H_{k-1} g_k)$$

and comparing this with (2.3):

$$S_k g_k = \beta_k H_k g_k \qquad (2.13)$$

where

$$\beta_k = \frac{\beta_0 \beta_{k-1}(\alpha_{k-1} - 1)}{c_k} . q_k^T H_{k-1} q_k$$

Since H_{k-1} is positive-definite (if H_0 is positive-definite), β_k is non-zero so long as $\alpha_{k-1} \neq 1$. Whilst β_k in (2.13) may be negative, the fact that we minimise along each direction ensures that p_k is the same for the two methods. We have therefore shown that, subject to the condition that $\alpha_{k-1} \neq 1$ and $c_k \neq 0$ for all k, the successive directions are the same for the two methods if $S_0 = \beta_0 H_0$. Q.E.D.

Appendix 3

Theorem:

Suppose that the function $f(x)$ is defined on $U \subset E^n$ and is such that:

(i) $f(x)$ is continuous on $\Omega = \{x | x \in U;\ f(x) \leqslant c\}$, and Ω is closed and bounded.

(ii) $f(x)$ has continuous second derivatives on $\Omega' = \{x | x \in U;\ f(x) < c\}$ and there is a Λ such that $||H(x)|| \leqslant \Lambda,\ x \in \Omega'$ (3.1)

Starting at any point $x_0 \in \Omega'$ with $g(x_0) \neq 0$, we generate a sequence $x_0, x_1, \ldots, x_k, x_{k+1}, \ldots$ according to:

$$p_{k+1} = x_{k+1} - x_k = -\alpha_k S_k g_k \tag{3.2}$$

Then if the matrices S_k satisfy the conditions:

$$\rho\|g_k\| \leqslant \|S_k g_k\| \leqslant \sigma\|g_k\| \tag{3.3}$$

$$|g_k^T S_k g_k| \geqslant \delta\|g_k\|.\|S_k g_k\| \tag{3.4}$$

where ρ, σ, δ are fixed constants, it is always possible to choose a finite non-zero α_k at each step such that:

$$f(x_k) - f(x_{k+1}) \geqslant \epsilon\alpha_k g_k^T S_k g_k > 0 \tag{3.5}$$

with ϵ a fixed positive constant less than unity.

With the α_k so chosen, the sequence (x_k) lies in Ω' and tends to $\Omega^* = \{x | x \in \Omega'; g(x) = 0\}$ in the sense that the distance $d(x_k, \Omega^*)$ of x_k from Ω^* tends to zero as $k \to \infty$.

Proof:

Suppose that $x_k \in \Omega'$. Then for sufficiently small p_{k+1} we have $x_{k+1} \in \Omega'$ and $(x_k + \theta p_{k+1}) \in \Omega'$ for $0 \leqslant \theta \leqslant 1$. In view of condition (ii) we may therefore expand $f(x_{k+1})$ about x_k in a Taylor series with a remainder of second order:

$$f(x_{k+1}) = f(x_k) + g_k^T p_{k+1} + \tfrac{1}{2}p_{k+1}^T.H(x_k + \theta p_{k+1}).p_{k+1} \qquad 0 \leqslant \theta \leqslant 1 \tag{3.6}$$

Substituting (3.2) into (3.6):

$$f(x_k) - f(x_{k+1}) = \alpha_k g_k^T S_k g_k \left\{1 - \frac{1}{2}\frac{\alpha_k(S_k g_k)^T.H(x_k + \theta p_{k+1}).(S_k g_k)}{g_k^T S_k g_k}\right\} \tag{3.7}$$

Now we may choose the sign of α_k to make $\alpha_k g_k^T S_k g_k$ positive (as required by 3.5) and, still considering sufficiently small p_{k+1}, we may use conditions (3.1), (3.3) and (3.4) to deduce from (3.7):

$$f(x_k) - f(x_{k+1}) \geqslant a_k g_k^T S_k g_k \left\{1 - \frac{\Lambda\sigma^2}{2\delta}.|\alpha_k|\right\} \tag{3.8}$$

It follows that x_{k+1} remains in Ω', and condition (3.5) is satisfied, as $|\alpha_k|$ increases from zero so long as:

$$|\alpha_k| \leqslant \frac{2\delta(1 - \epsilon)}{\Lambda\sigma^2} \tag{3.9}$$

Since $x_0 \in \Omega'$ we can therefore choose a finite non-zero α_k at each step so that condition (3.5) is satisfied and the whole sequence (x_k) lies in Ω'. With the α_k so chosen the sequence $f(x_k)$ is monotonically decreasing; it follows from condition (i) that it is convergent, and we have:

$$f(x_k) - f(x_{k+1}) \to 0, \quad k \to \infty \tag{3.10}$$

But from (3.3), (3.4) and (3.5):

$$f(x_k) - f(x_{k+1}) \geqslant \epsilon\delta\rho.|\alpha_k|.\|g_k\|^2 \tag{3.11}$$

and since all α_k are non-zero it follows from (3.10) and (3.11) that:

$$g_k \to 0, \quad k \to \infty \tag{3.12}$$

Further, from (3.2) and (3.3):

$$\|p_{k+1}\| = |\alpha_k|.\|S_k g_k\| \leqslant \sigma.|\alpha_k|.\|g_k\| \tag{3.13}$$

and therefore:

$$p_{k+1} \to 0, \quad k \to \infty \tag{3.14}$$

Now suppose that the sequence (x_k) does not tend to Ω^*. Then since the sequence lies in Ω, a closed, bounded region of E^n, it contains a sub-sequence (x_i) for which: $d(x_i, \Omega^*) \to a > 0$

But (x_i) in turn contains a subsequence converging to a point, say z, contained in Ω', so that:

$$d(z, \Omega^*) = a > 0 \tag{3.15}$$

But $g_k \to 0$, so that $g(z) = 0$ and $z \in \Omega^*$. This contradicts (3.15), so the sequence (x_k) must in fact tend to Ω^*.

This completes the proof of the theorem.

References

Broyden, C. G. (1967). Quasi-Newton Methods and their Application to Function Minimization, *Math. Comp.*, Vol. 21, p. 368.

Carathéodory, C. (1961). *Calculus of Variations and Partial Differential Equations of the First Order*, Vol. 2, Holden-Day: New York, p. 190.

Colville, A. R. (1968). A Comparative Study on Nonlinear Programming Codes, I.B.M. New York Scientific Center Technical Report No. 320–2949.

Davidon, W. C. (1959). Variable Metric Method for Minimization, A.E.C. Research and Development Report, ANL–5990 (Rev.).

Davidon, W. C. (1968). Variance Algorithm for Minimization, *The Computer Journal*, Vol. 10, p. 406.

Fletcher, R., and Powell, M. J. D. (1963). A Rapidly Convergent Descent Method for Minimization, *The Computer Journal*, Vol. 6, p. 163.

Murtagh, B. A., and Sargent, R. W. H. (1968). A Constrained Minimization Method with Quadratic Convergence, Paper presented at I.M.A.–B.C.S. Joint Conference on Optimization, Univ. of Keele, March 1968.

Myers, G. E. (1968). Properties of the Conjugate-Gradient and Davidon Methods, *Journal of Optimization Theory and Applications*, Vol. 2, No. 4, p. 209.