J. Inst. Maths Applics (1971) **8**, 315–327

Minimization Algorithms Making Use of Non-quadratic Properties of the Objective Function

M. C. Biggs

Numerical Optimization Centre, The Hatfield Polytechnic, Hatfield, Hertfordshire, England

[Received 13 January 1971 and in revised form 9 April 1971]

This paper proposes modifications to the matrix updating formulae used in the Fletcher-Powell (1963), Broyden (1970) and Fletcher (1970) methods for function minimization, and discusses some properties of these modified formulae. A function minimization algorithm incorporating the new expressions has been programmed and the results of tests with some well-known functions are reported.

Notation

$f(\mathbf{x})$ a function of n variables $x_1, x_2, ..., x_n$

$\bar{\mathbf{x}}$ the point where f has its minimum value

$\mathbf{x}_i$ an estimate of the position of the minimum reached after i iterations

f_i the value of f at $\mathbf{x}_i$

$\mathbf{g}_i$ the gradient of f at $\mathbf{x}_i$

G_i the Hessian matrix of f at $\mathbf{x}_i$

H_i an approximation to G_i^{-1}

$\mathbf{s}_i$ the correction vector $-H_i\mathbf{g}_i$

$\hat{\mathbf{s}}_i = \mathbf{s}_i/\|\mathbf{s}_i\|$

$\boldsymbol{\delta}_i = \mathbf{x}_{i+1} - \mathbf{x}_i = -\alpha_i H_i \mathbf{g}_i$

$\boldsymbol{\gamma}_i = \mathbf{g}_{i+1} - \mathbf{g}_i$

1. Introduction

Many variants of the variable metric method (VMM) have been written to solve the problem of minimizing a function $f(\mathbf{x})$ whose gradient is available. The technique was originally proposed by Davidon (1959) and subsequently improved by Fletcher & Powell (1963). Later versions have been devised by, among others, Broyden (1967, 1970), Pearson (1969), Powell (1969) and Fletcher (1970).

The central feature of all VMM implementations is the use of successive approximations to the inverse Hessian matrix of f. If at the point $\mathbf{x}_i$ the gradient is $\mathbf{g}_i$ and the inverse Hessian approximation is H_i then a new point $\mathbf{x}_{i+1}$ is given by $\mathbf{x}_i + \boldsymbol{\delta}_i = \mathbf{x}_i - \alpha_i H_i \mathbf{g}_i$ (where α_i is a scalar chosen to ensure that $f_{i+1} < f_i$). The inverse Hessian approximation is then revised by considering the change in gradient $\boldsymbol{\gamma}_i$ caused by the move $\boldsymbol{\delta}_i$. Several formulae for obtaining H_{i+1} from H_i have been proposed. The Fletcher–Powell formula is

$$H_{i+1} = H_i - \frac{H_i\boldsymbol{\gamma}_i\boldsymbol{\gamma}_i^T H_i}{\boldsymbol{\gamma}_i^T H_i \boldsymbol{\gamma}_i} + \frac{\boldsymbol{\delta}_i\boldsymbol{\delta}_i^T}{\boldsymbol{\delta}_i^T\boldsymbol{\gamma}_i}. \tag{1}$$

A more recent expression discovered independently by Broyden (1970) and Fletcher (1970) is

$$H_{i+1} = H_i - \frac{\boldsymbol{\delta}_i^T\boldsymbol{\gamma}_i H_i}{\boldsymbol{\delta}_i^T\boldsymbol{\gamma}_i} - \frac{H_i\boldsymbol{\gamma}_i\boldsymbol{\delta}_i^T}{\boldsymbol{\delta}_i^T\boldsymbol{\gamma}_i} + \left(1 + \frac{\boldsymbol{\gamma}_i^T H_i \boldsymbol{\gamma}_i}{\boldsymbol{\delta}_i^T\boldsymbol{\gamma}_i}\right)\frac{\boldsymbol{\delta}_i\boldsymbol{\delta}_i^T}{\boldsymbol{\delta}_i^T\boldsymbol{\gamma}_i}. \tag{2}$$

It will be noted that both these formulae cause the relation $H_{i+1}\boldsymbol{\gamma}_i = \boldsymbol{\delta}_i$ to hold.

Detailed descriptions of the properties of VMM appear in the literature and we shall merely repeat some of the more important results. Davidon showed that if at every iteration α_i is chosen so that f is minimized along the direction $\mathbf{s}_i$ then VMM will converge to the minimum of a convex, quadratic function in at most n iterations. Moreover, in this case H_i will converge to the true inverse Hessian G^{-1}. Recently Powell (1969) has proved the much more general result that VMM with linear minimizations will converge to the minimum of f if it is strictly convex and twice differentiable. In practical examples, f may not be convex everywhere (although it must be so in the neighbourhood of the solution). However, if H_0, the initial estimate of the inverse Hessian, is positive definite then (1) and (2) ensure that, with exact arithmetic, all subsequent H_i will be positive definite too, and hence that the search directions $-H_i\mathbf{g}_i$ will be "downhill".

The relative merits of the formulae (1) and (2) have been discussed elsewhere (Fletcher, 1970; Broyden, 1970). The point that we wish to bring out is that both formulae make the implicit assumption that the second directional derivative along $\mathbf{s}_i$ can be adequately approximated by first differences. We shall show (Section 3) that after updating by (1) or (2) the new estimate of the second directional derivative along $\mathbf{s}_i$ is

$$\frac{\hat{\mathbf{s}}_i\mathbf{g}_{i+1}-\hat{\mathbf{s}}_i g_i}{\|\mathbf{x}_{i+1}-\mathbf{x}_i\|}.$$

While this is accurate if f is nearly quadratic along $\mathbf{s}_i$, it may be a poor approximation if there is significant non-quadratic behaviour and $\|\mathbf{x}_{i+1}-\mathbf{x}_i\|$ is large. In practice f will generally not be quadratic so that far from the solution H_i need have little resemblance to the true inverse Hessian. Now it has been noted (Pearson, 1969) that the Newton method, which computes the true Hessian at every step, frequently converges in fewer iterations than VMM. This suggests the general remark (and there will of course be exceptions) that better search directions may be obtained by using the exact Hessian than by using an approximation of the kind given by (1) or (2). Because of the disadvantages of calculating the Hessian at every iteration there is a motivation for seeking matrix updating formulae which, in the non-quadratic case, will approximate the true inverse Hessian more closely than do either (1) or (2). If the function is locally non-convex, however, we wish the approximate inverse Hessian to remain positive definite and H_i will then not necessarily resemble G_i^{-1} until a convex region is re-entered.

We use ideas which are developments of techniques described elsewhere by Dixon & Biggs (1970) and Biggs (1970). The central notion is the estimation of the "dominant degree" of a one-dimensional convex function. Previously (Biggs, 1970; Dixon & Biggs, 1970) this estimate has been quite successfully used as the basis of a linear search process. (Indeed it is available for that purpose in the algorithm described later which, while not performing explicit minimizations along every search direction $\mathbf{s}_i$, may require more than one function evaluation per iteration to obtain a satisfactory reduction in f.) But if we have some measure of the non-quadraticity of the function f in the direction $\mathbf{s}_i$ then we can attempt to improve upon the simple estimate of the second directional derivative and hence update the matrix $\mathbf{H}_i$ using more accurate information.

Section 2 describes briefly the means by which the dominant degree is estimated and the way in which it is employed to revise the second derivative approximation. Section 3 discusses the resulting modified forms of the updating formulae (1) and (2). Section 4 deals with computer implementations of these ideas and gives some very encouraging test results. Some further developments are suggested in Section 5.

2. The Estimation of the "Dominant Degree" of a One Dimensional Function

Consider the function

$$A|y-a|^p+b \tag{3}$$

where $A > 0, p > 1$.

This is clearly a convex function of y with a minimum at $y = a$, and furthermore it is dominated by terms in y^p. We estimate the "dominant degree" of any other convex function $f(y)$ by assuming that it has the form (3) and using the available information about $f(y)$ to determine an appropriate value of p.

Suppose that at some initial point y_0 we know the function value f_0, the first derivative f'_0, and that we also have available an approximate second derivative $\tilde{f}''_0$ where $\tilde{f}''_0 = \eta f''_0$, $\eta(> 0)$ being some unknown error factor. Then for any specified α we can define a scaled Newton-like step $\Delta y = -\alpha f'_0/\tilde{f}''_0$. Let $y^* = y_0+\Delta y$ and suppose also that we know $f^* = f(y^*)$ and $f^{*\prime} = f'(y^*)$. All this is information that would normally be used by a VMM algorithm.

It is shown in the appendix that if f is assumed to have the form (3) then we can obtain a pair of non-linear equations from which to determine the unknowns η and p. If we define

$$D = -\frac{f_0-f^*}{f'_0\Delta y} \quad \text{and} \quad \beta = \frac{f^{*\prime}}{f'_0} \tag{4}$$

then these equations can be written

$$\frac{\eta(p-1)(1-\beta)}{\alpha p}+\frac{\beta}{p} = D \tag{5}$$

$$\left|1-\frac{\alpha}{\eta(p-1)}\right|^p\left(1-\frac{\alpha}{\eta(p-1)}\right)^{-1} = \beta. \tag{6}$$

Since the method of solving these equations is not central to the theme of this paper we shall not discuss it except to remark that a fairly simple iterative procedure was found to be satisfactory during the tests of the computer programs described in Section 4.

Assuming η and p to have been found it is obviously true that the validity of these estimates of second derivative error and dominant degree must depend upon the suitability of f for representation in the form (3). It should be noted however that functions which cannot be reasonably well represented by (3) are quite readily detected by consideration of the values of D and β (for example if $\beta > 1$ the function is non convex). Some simple programming safeguards can ensure that equations (5) and (6) need only be solved when the behaviour of f appears suitable.

The estimates η and p can be used in two ways. We may wish to find a point nearer to the minimum of f than the point y^*. Consideration of (3) suggests that $y_0-\eta(p-1)$

$f_0'/\tilde{f}_0''$ will be such a point. Alternatively, as in the context of VMM, we may wish to obtain a good approximation to the second derivative $f^{*''}$. The simple first difference expression is commonly used:

$$f^{*''} \approx (f_0' - f^{*'})/(y_0 - y^*).$$

However it is shown in the appendix that if f can be well represented by (3), a better estimate is

$$f^{*''} = \frac{1}{\eta^*}\frac{f_0' - f^{*'}}{y_0 - y^*}$$

where

$$\eta^* = \frac{\eta}{\alpha}\left(1 - \frac{\alpha}{\eta(p-1)}\right)\left(\frac{1}{\beta} - 1\right). \tag{7}$$

3. Modified Matrix Updating Formulae

If Γ is written for H^{-1} it is possible to obtain updating formulae for Γ corresponding to equations (1) and (2).

From (1)

$$\Gamma_{i+1} = \Gamma_i - \frac{\gamma_i \delta_i^T \Gamma_i}{\delta_i^T \gamma_i} - \frac{\Gamma_i \delta_i \gamma_i^T}{\delta_i^T \gamma_i} + \left(1 + \frac{\delta_i^T \Gamma_i \delta_i}{\delta_i^T \gamma_i}\right)\frac{\gamma_i \gamma_i^T}{\delta_i^T \gamma_i}. \tag{8}$$

From (2)

$$\Gamma_{i+1} = \Gamma_i - \frac{\Gamma_i \delta_i \delta_i^T \Gamma_i}{\delta_i^T \Gamma_i \delta_i} + \frac{\gamma_i \gamma_i^T}{\delta_i^T \gamma_i}. \tag{9}$$

Consider $\hat{\mathbf{s}}_i^T \Gamma_{i+1} \hat{\mathbf{s}}_i$ the estimated value of the second directional derivative along $\mathbf{s}_i$. It is clear that both (8) and (9) give

$$\hat{\mathbf{s}}_i^T \Gamma_{i+1} \hat{\mathbf{s}}_i = \frac{(\hat{\mathbf{s}}_i^T \gamma_i)^2}{\delta_i^T \gamma_i} = \frac{\hat{\mathbf{s}}_i^T \gamma_i}{\alpha_i \|\mathbf{s}_i\|}$$

$$= \frac{\hat{\mathbf{s}}_i^T (\mathbf{g}_{i+1} - \mathbf{g}_i)}{\|\mathbf{x}_{i+1} - \mathbf{x}_i\|}.$$

This equation emphasizes the fact that the accuracy of this second derivative approximation is comparable with what would be obtained by using a simple gradient differencing procedure. The differencing interval may be quite a coarse one, and Section 2 suggests that, if η^* is obtained from equation (7), we can obtain an improved estimate of the second directional derivative by forcing

$$\hat{\mathbf{s}}_i^T \Gamma_{i+1} \mathbf{s}_i = \frac{1}{\eta^*}\frac{\hat{\mathbf{s}}_i^T (\mathbf{g}_{i+1} - \mathbf{g}_i)}{\|\mathbf{x}_{i+1} - \mathbf{x}_i\|}$$

$$= \frac{1}{\eta^*}\frac{(\hat{\mathbf{s}}_i^T \gamma_i)}{\delta_i^T \gamma_i}.$$

We can make simple alterations to (8) and (9) to produce matrices Γ_{i+1} which give us this result.

$$\Gamma_{i+1} = \Gamma_i - \frac{\gamma_i \delta_i^T \Gamma_i}{\delta_i^T \gamma_i} - \frac{\Gamma_i \delta_i \gamma_i^T}{\delta_i^T \gamma_i} + \left(\frac{1}{\eta^*} + \frac{\delta_i^T \Gamma_i \delta_i}{\delta_i^T \gamma_i}\right)\frac{\gamma_i \gamma_i^T}{\delta_i^T \gamma_i}, \tag{10}$$

$$\Gamma_{i+1} = \Gamma_i - \frac{\Gamma_i \delta_i \delta_i^T \Gamma_i}{\delta_i^T \Gamma_i \delta_i} + \frac{1}{\eta^*}\frac{\gamma_i \gamma_i^T}{\delta_i^T \gamma_i}. \tag{11}$$

It is a simple matter to show that these transform back into two new formulae for H, namely

$$H_{i+1} = H_i - \frac{H_i\gamma_i\gamma_i^T H_i}{\gamma_i^T H_i \gamma_i} + \eta^* \frac{\delta_i\delta_i^T}{\delta_i^T\gamma_i}, \tag{12}$$

$$H_{i+1} = H_i - \frac{\delta_i\gamma_i^T H_i}{\delta_i^T\gamma_i} - \frac{H_i\gamma_i\delta_i^T}{\delta_i^T\gamma_i} + \left(\eta^* + \frac{\gamma_i^T H_i\gamma_i}{\delta_i^T\gamma_i}\right)\frac{\delta_i\delta_i^T}{\delta_i^T\gamma_i}. \tag{13}$$

It is easy to show that, for $\eta^* > 0$, (12) and (13) generate positive definite matrices in just the same way as do (1) and (2)—i.e. when the updating is carried out over an interval δ_i such that $\delta_i^T\gamma_i > 0$. This means that we can update over *any* interval δ_i if f is convex but that we may have to select more carefully if f is nonconvex. A suitable δ_i can always be found, however, as long as f is bounded below. It can also be shown that if f is quadratic then equation (7) will always yield $\eta^* = 1$ and hence the formulae (12) and (13) have the same properties relating to quadratic termination as the formulae (1) and (2). Thus by using the new expressions (12) and (13) we have retained the desirable features of the more usual matrix updating formulae.

It should be noted that as well as forcing the curvature along $\mathbf{s}_i$ to be "correct" (within our assumptions) the formulae (12) and (13) both give $H_{i+1}\gamma_i = \eta^*\delta_i$. Qualitatively, even if not quantitatively, this is a reasonable property since for $\eta^* \neq 1$, i.e. for f non-quadratic, it means that the change in gradient has not been attributed entirely to second derivative terms evaluated at $\mathbf{x}_{i+1}$.

A numerical difficulty occurs if δ_i happens to minimize f in the direction $\mathbf{s}_i$. Then η^* cannot be determined by expression (7) since it contains the terms $1/\beta$ and $1-\alpha/\eta(p-1)$, of which the former is infinite and the latter zero. In fact, at the minimum $y = a$ of the model function (3) η^* is zero if $p < 2$ (because the second derivative is infinite) and infinite if $p > 2$ (because the second derivative is zero). Therefore, except in the quadratic case, an accurate estimate of curvature close to a line minimum may be impractical. If a line minimum is approached we might choose to set $\eta^* = 1$ and reject any dominant degree information, or alternatively we could use some large or small constant to represent the theoretical infinite or zero value of η^*. In the latter case however we should have to take care not to use values that would make our matrices tend to singularity. (In fact, in the algorithm described in Section 4, we used the simpler alternative of setting $\eta^* = 1$ near a line minimum.) These remarks suggest that we would do better to use (12) and (13) in optimization algorithms which do not accurately minimize the function along every new direction. In considering algorithms of this type we have been encouraged by the success of the new method of Fletcher (1970) as well as by some attractive features in the behaviour of the Newton-based algorithm "MEANDER" (Dixon & Biggs, 1970).

An algorithm will be outlined in the next section. We have tried two approaches, one making use of the updating formula (13) only, and the other employing either (12) or (13) on the basis of a test performed at each iteration.

4. Computer Implementation

We shall describe the algorithm assuming, for simplicity, that no difficulty is encountered in solving equations (5) and (6).

Stage 1. At the current point $\mathbf{x}_i$ we calculate $\mathbf{s}_i = -H_i\mathbf{g}_i$. We then make a move $\alpha_i\mathbf{s}_i$ where α_i is a scaling factor to be determined. There are two "standard" settings for α_i namely $\alpha_i = \min(1/\|\mathbf{s}_i\|, .1)$ if $i \leqslant n$ or $\alpha_i = 1$ if $i > n$. The first of these is a "cautious" value used while the matrix is being built up from its initial setting; the second is analogous to the use of a full Newton step. These "standard" settings may be overridden however when $\mathbf{s}_i$ and $\mathbf{s}_{i-1}$ are nearly parallel and we have some information from the previous iteration about the dominant degree, p, of the function along the line $\mathbf{s}_{i-1}$. In this case we take $\alpha_i = (p-1)$, i.e. the factor which would cause a scaled Newton step to minimize (3).

Stage 2. $\boldsymbol{\delta}_i = \alpha_i\mathbf{s}_i$, $\mathbf{x}^* = \mathbf{x}+\boldsymbol{\delta}_i$, $f^* = f(\mathbf{x}^*)$, $\mathbf{g}^* = \mathbf{g}(\mathbf{x}^*)$. Calculate the parameters

$$\beta = \frac{\boldsymbol{\delta}_i^T\mathbf{g}^*}{\boldsymbol{\delta}_i^T\mathbf{g}_i} \quad \text{and} \quad D = \frac{f_i - f^*}{-\delta_i^T\mathbf{g}_i}.$$

Equations (5) and (6) then give values for η and p.

Stage 3. If $1-\varepsilon > D > \varepsilon$, where ε is a chosen small quantity (typically $\varepsilon = 0{\cdot}001$) we regard the step $\boldsymbol{\delta}_i$ as satisfactory. Satisfying the right hand inequality ensures that some reduction in function value is made when the minimum has been bracketed. Satisfying the left hand inequality prevents the step length $\boldsymbol{\delta}_i$ becoming very short compared with the distance to the minimum. The effect of the test therefore is to stop the move $\boldsymbol{\delta}_i$ being excessively long or trivially small. If the point $\mathbf{x}^*$ is unacceptable then a revised scaling factor $\alpha_i = \eta(p-1)$ is calculated and the algorithm returns to stage 2, to determine a new $\mathbf{x}^*$.

Stage 4. When a satisfactory point $\mathbf{x}^*$ has been found we can calculate η^* by equation (7). We write $\mathbf{x}_{i+1} = x^*, f_{i+1} = f^*, \mathbf{g}_{i+1} = \mathbf{g}^*, \boldsymbol{\gamma}_i = \mathbf{g}^* - \mathbf{g}_i$. Convergence tests are now carried out, as in the Fletcher–Powell algorithm, providing $i \geqslant n$. We say that the process has converged if $\|\boldsymbol{\delta}_i\|$ is less than some specified small quantity. If the convergence test is not satisfied the new matrix H_{i+1} is calculated either by equation (12) or by equation (13), and the algorithm repeats from Stage 1.

In cases where η and p cannot be determined from (5) and (6) the pattern of the algorithm is only slightly changed: revision of α_i (where necessary) has to be performed by, for example, simple parabolic interpolation and η^* must be set arbitrarily to 1 or else be derived from an expression other than equation (7). Fuller programming details are given in Biggs (1970).

Results will now be presented for two implementations of the new method. Version A employs a switching technique due to Fletcher (1970) to decide which of the two formulae (12) and (13) to use at each iteration. If $\boldsymbol{\delta}_i^T\boldsymbol{\gamma}_i \geqslant \boldsymbol{\gamma}_i^T H_i \boldsymbol{\gamma}_i$ (13) is used; and if $\boldsymbol{\delta}_i^T\boldsymbol{\gamma}_i < \boldsymbol{\gamma}_i^T H_i \boldsymbol{\gamma}_i$ then formula (12) is employed. Version B uses formula (13) throughout. The first trials were performed using some well known test problems, namely: Rosenbrock's (1960) parabolic valley (two variables), Powell's (1962) quartic function (four variables), the Fletcher-Powell (1963) helical valley (three variables), Wood's function (Colville 1968) of four variables and the Chebyquad family (Fletcher 1965). Performance was compared with results quoted by Fletcher (1970) for his own new method and for the Fletcher-Powell (1963) algorithm. The numbers of iterations (I) and function and gradient evaluations (FV) required by each method for convergence are tabulated below (the solution being assumed to have been reached when $\|\boldsymbol{\delta}_i\| \leqslant 5 * 10^{-5}$).

On this evidence our approach appears to be promising since in most cases our algorithms need fewer evaluations than the methods with which they are compared. The fact that they tend to use fewer iterations than Fletcher's (1970) method is also encouraging since this suggests that the predictions made using inverse Hessians given by formulae (12) and (13) are on the whole more efficient than predictions based on the "standard" Hessian approximations (1) and (2). It is also worth noting that both versions of our method converge on the Powell (quartic) function, whereas Fletcher reports that the other methods approached the solution closely but made an error exit before converging. Differences between versions A and B of the new method seem to be significant only in the case of Chebyquad 8, where version B does very much better than version A. This is the only example in this set where the new approach did not compare well with Fletcher's (1970) method.

TABLE 1

Comparison of methods

Problem	New method Version A		New method Version B		Fletcher (1970)		Fletcher Powell (1963)	
	I	FV	I	FV	I	FV	I	FV
Parabolic valley	29	35	31	36	39*	47*	22*	64*
Helical valley	25	29	26	29	NR	NR	22	54
Powell's function	34	41	39	44	42*f**	43*f**	21*f**	64*f**
Wood's function	38	44	39	47	122*	136*	45*	154*
Chebyquad 2	5	6	5	7	7*	8*	2*	6*
Chebyquad 4	10	14	11	17	10*	13*	6*	22*
Chebyquad 6	18	25	16	25	22*	27*	10*	29*
Chebyquad 8	38	42	24	31	21*	23*	18*	50*

f, Failed to reach desired accuracy; NR, not reported; * result quoted from Fletcher (1970).

While the above examples are certainly not trivial it is nevertheless true that all published implementations of VMM have successfully solved them. We next considered some tests of a more difficult nature. Broyden (1970) reports that the Fletcher–Powell algorithm failed when applied to least squares curve fitting by sums of exponentials. Such problems are in any case known to be difficult. The functions involved have unpleasant nonconvex regions where the gradient can become very small and so cause premature "convergence". Furthermore it appears that the functions have long, curving steep-sided valleys which are frequently awkward to negotiate.

We have devised a set of exponential functions both to assess the performance of our minimization algorithm and, incidentally, to investigate the way in which the difficulty of the problem increases with the number of variables. The examples are as follows:

EXP 2

$$f(\mathbf{x}) = \sum_{i=1}^{10} ((e^{-x_1 z_i} - 5\, e^{-x_2 z_i}) - y_i)^2$$

where

$$y_i = e^{-z_i} - 5\,e^{-10z_i}$$

and

$$z_i = (0{\cdot}1)i \qquad i = 1, 2 \dots 10.$$

Starting approximation:

$$x_1 = 1, x_2 = 2, f = 32{\cdot}26.$$

EXP 3

$$f(\mathbf{x}) = \sum_{i=1}^{10} ((e^{-x_1 z_i} - x_3\,e^{-x_2 z_i}) - y_i)^2$$

where

$$y_i = e^{-z_i} - 5\,e^{-10z_i}$$

and

$$z_i = (0{\cdot}1)i \qquad i = 1, 2, \dots, 10.$$

Starting approximation:

$$x_1 = 1, x_2 = 2, x_3 = 1, f = 1{\cdot}599.$$

EXP 4

$$f(\mathbf{x}) = \sum_{i=1}^{10} ((x_3\,e^{-x_1 z_i} - x_4\,e^{-x_2 z_i}) - y_i)^2$$

where

$$y_i = e^{-z_i} - 5\,e^{-10z_i}$$

and

$$z_i = (0{\cdot}1)i \qquad i = 1, 2, \dots, 10.$$

Starting approximation:

$$x_1 = 1, x_2 = 2, x_3 = 1, x_4 = 1, f = 1{\cdot}599.$$

EXP 5

$$f(\mathbf{x}) = \sum_{i=1}^{11} ((x_3\,e^{-x_1 z_i} - x_4\,e^{-x_2 z_i} + 3\,e^{-x_5 z_i}) - y_i)^2$$

where

$$y_i = e^{-z_i} - 5\,e^{-10z_i} + 3\,e^{-4z_i}$$
$$z_i = (0{\cdot}1)i \qquad i = 1, 2, \dots, 11.$$

Starting approximation:

$$x_1 = 1, x_2 = 2, x_3 = 1, x_4 = 1, x_5 = 1, f = 13{\cdot}39.$$

EXP 6

$$f(\mathbf{x}) = \sum_{i=1}^{13} ((x_3\,e^{-x_1 z_i} - x_4\,e^{-x_2 z_i} + x_6\,e^{-x_5 z_i}) - y_i)^2$$

where

$$y_i = e^{-z_i} - 5\,e^{-10z_i} + 3\,e^{-4z_i}$$
$$z_i = (0{\cdot}1)i \qquad i = 1, 2, \dots, 13.$$

Starting approximation:

$$x_1 = 1, x_2 = 2, x_3 = 1, x_4 = 1, x_5 = 1, x_6 = 1, f = 0{\cdot}779.$$

Also in this set of tests we considered the Weibull function described by Shanno (1970) who reports that it too has defeated the Fletcher–Powell algorithm. The function is

$$f(\mathbf{x}) = \sum_{i=1}^{99} \left(\exp\left(-\frac{(y_i - x_3)^{x_2}}{x_1} \right) - z_i \right)^2$$

where

$$y_i = 25 + \left(50 \log_e \left(\frac{1}{z_i} \right) \right)^{\frac{2}{3}}$$

$$z_i = (0{\cdot}01)i \qquad i = 1, 2, \ldots, 99.$$

Starting approximation:

$$x_1 = 250, x_2 = 0.3, x_3 = 5, f = 31.69.$$

TABLE 2

Comparison of methods

Problem	New method version A I	New method version A FV	New method version B I	New method version B FV	Fletcher (1970) I	Fletcher (1970) FV	Fletcher–Powell (1963) I	Fletcher–Powell (1963) FV
EXP 2*	13	17	13	17	14	17	6	21
EXP 3*	21	31	16	21	17	21	11	32
EXP 4*	28	32	28	32	31	37	20	61
EXP 5†	55	73	59	81	121L	151L	F	
EXP 6†	210	244	145	174	220	290	110	328
Weibull*	90	109	60	85	61	79	F	

F, Failed—see text; L, found a local minimum—see text; *, converged when $\|\delta_i\| < 10^{-5}$; †, converged when $\|\delta_i\| < 10^{-6}$.

In the absence of published results for other procedures applied to these problems we carried out our own tests using N.O.C. implementations of the Fletcher–Powell algorithm and Fletcher's (1970) method. The Fletcher–Powell program was a FORTRAN version of a subroutine entitled FPMOD 4 (Biggs, 1970), which, in turn was a slightly modified version of the procedure FLEPOMIN described by Fletcher (1966). The implementation of Fletcher's method follows quite closely the description given in Fletcher (1970) with the exception that where Fletcher occasionally uses cubic interpolation to determine an acceptable point if the step $\delta_i = -H_i g_i$ is unsuccessful, our program uses instead the "dominant power" line search. It is not thought that the difference is very significant since, after the first n iterations at least, the move $\delta_i = -H_i g_i$ is almost always acceptable and no line search is used. If anything, the fact that the new method and our implementation of Fletcher's (1970) procedure have a line search in common will tend to make comparison more meaningful and bring out the effect of the η^* factor in the updating.

As before, two variants of the new method were tested, version A employing a "switching" device between formulae (12) and (13) and version B using formula (13) exclusively.

Without drawing too many conclusions from this experimental evidence we note that the new methods find the expected solution in every case; and that on most of the examples they reach the solution as quickly as, if not quicker than, the other techniques. Moreover it is encouraging to observe that differences in performance become more marked in favour of our algorithm for the larger problems EXP 5 and EXP 6.

It is clear that Fletcher's algorithm has also performed well, and the result for problem EXP 5 indicates that the method found what appears to be a local minimum with a function value $f = 2{\cdot}65 * 10^{-3}$. When the new method was started near the point where Fletcher's algorithm terminated it converged to the same solution. The failures of the Fletcher–Powell program both occur in the line search when the cubic procedure has apparently found a very small interval bracketing the line minimum but the function value has not in fact been reduced. This may show a weakness in the interpolation process rather than in the matrix updating; but even had the line search not failed the method was in any case making slow progress to the solution. On problem EXP 5 138 function evaluations were required to reduce the function to 10^{-5}; and on the Weibull problem 132 function evaluations were used in reaching $f < 10^{-2}$.

It should be mentioned that the new method converged successfully on both EXP 5 and the Weibull problem when started close to the points at which the Fletcher–Powell method terminated.

It can be seen that the difference between version A and version B of the new method is sometimes quite significant. The factors governing this are not yet fully understood. The fact that version B is better than version A on three of these exponential problems agrees with Broyden's observations (Broyden, 1970). But the problem EXP 5 is solved more rapidly by version A of the method which suggests that there may at times be some advantage in using both formulae (12) and (13).

A final word should perhaps be said about our practical experience with equations (5) and (6). With the test problems quoted it only happened rarely that η and p could not be found. Moreover, with the accuracy demanded of our iterative solutions procedure—namely that successive estimates of η and p should differ by less than 0·005—the equations were usually solved in about 6–10 cycles. This means, in fact, that the computing effort required by this part of the calculation is quite small. Moreover it becomes less significant as the size of the problem increases since the solution of (5) and (6) is independent of the number of variables and of the complexity of the function evaluation procedure.

5. Conclusion

A new matrix updating formula has been proposed for use in function minimization algorithms. Its aim is to estimate the non-quadraticity of the objective function in order to obtain a more accurate approximation to the true inverse Hessian than that given by either the Fletcher–Powell or the Broyden formula. The idea has been incorporated in a new function minimization algorithm (not based upon linear minimization) and initial trials with a computer implementation of this algorithm are promising.

In the case when the objective function is quadratic, or nearly so, the modified matrix updating formulae of this report become identical with the Fletcher–Powell and Broyden expressions. Since we might expect near-quadratic behaviour near the solution the method can, in some sense, be supported by the convergence theorems already established for VMM. A more rigorous analysis proving that the method will converge for a general class of functions is obviously desirable; but at present we have only the circumstantial evidence of success on some non quadratic test functions. It would also be advantageous to know under what circumstances expression (13) should be preferred to expression (12), since, in our computational experiments, we have found that the performance of the minimization algorithm is sometimes sensitive to the choice of updating formula. However we are sufficiently encouraged by our practical experience of the algorithm to believe that further theoretical investigation is justified.

The author wishes to thank his colleagues at the Numerical Optimization Centre, Hatfield Polytechnic, for many valuable discussions during the preparation of this paper. The author is also indebted to the referee for suggestions which resulted in improvements to Sections 3 and 4.

References

BIGGS, M. C. 1970 A new method of linear minimisation. *Numerical Optimisation Centre. Tech. Report.* No. 8.

BIGGS, M. C. 1970 A new variable metric technique taking account of non quadratic behaviour of the objective function. *N.O.C. Tech. Report. No.* 17.

BROYDEN, C. G. 1967 *Maths. Comput.* **21**, 368.

BROYDEN, C. G. 1970 The convergence of a class of double rank minimization algorithms. *J. Inst. Maths Applics* **6**, 222.

COLVILLE, A. R. 1968 A comparative study on nonlinear programming codes. *IBM tech. report No.* 320–2949.

DAVIDON, W. C. 1959 Variable metric method for minimization. *Argonne Nat. Lab. report ANL*-5990 *Rev.*

DIXON, L. C. W. & BIGGS, M. C. 1970 "MEANDER"—A Newton based procedure for *n*-dimensional function minimization. *N.O.C. Tech. Report No. 9.*

FLETCHER, R. & POWELL, M. J. D. 1963 *Comp. J.* **6**, 163.

FLETCHER, R. 1965 *Comp. J.* **8**, 33.

FLETCHER, R. 1970 *Comp. J.* **13**, 317.

POWELL, M. J. D. 1962 *Comp. J.* **5**, 147.

POWELL, M. J. D. 1969 Rank one methods for unconstrained optimization. *AERE report TP* 372.

POWELL, M. J. D. 1969 On the convergence of the variable metric algorithm. *AERE report TP* 382.

PEARSON, J. D. 1969 *Comp. J.* **12**, 171.

ROSENBROCK, H. H. 1960 *Comp. J.* **3**, 175.

WOLFE, P. 1967 *Another variable metric method.*

SHANNO, D. 1970 *Maths. Comp.* July 1970.

BIGGS, M. C. 1970 Computational experience with variable metric methods. *N.O.C. Tech. Report No.* 7.

FLETCHER, R. 1966 *Communs Ass. comput. Mach.* **9**, 686.

Appendix

Derivation of equations (5), (6) *and* (7)

Consider the function

$$f(y) = A|y-a|^p+b \tag{3}$$

where $A > 0$, $p > 1$.

Suppose that we can calculate the function f_0 and its first derivative f_0' at some point y_0, but that we have only an approximate second derivative $\tilde{f}_0''$ where $\tilde{f}_0'' = \eta f_0''$, η being unknown (> 0).

Now

$$f_0 = A|y_0-a|^p+b,$$
$$f_0' = Ap|y_0-a|^p(y_0-a)^{-1},$$
$$f_0'' = Ap(p-1)|y_0-a|^p(y_0-a)^{-2}.$$

If we employ the approximate second derivative in order to make a scaled "Newton-like" step from y_0 towards the minimum of $f(y)$ we shall reach a new point y^* where

$$y^* = y_0-\alpha f_0'/\tilde{f}''_0,$$
$$= y_0-\alpha(y_0-a)/\eta(p-1).$$

Now

$$f^* = A|y_0-a|^p\left|1-\frac{\alpha}{\eta(p-1)}\right|^p+b,$$

$$f^{*\prime} = Ap|y_0-a|^p\left|1-\frac{\alpha}{\eta(p-1)}\right|^p(y_0-a)^{-1}\left(1-\frac{\alpha}{\eta(p-1)}\right)^{-1}.$$

Let f_p be a linear prediction of the value of f at y^*

i.e.

$$f_p = f_0-\alpha f_0'^2/\tilde{f}''_0$$

so

$$f_0-f_p = \frac{\alpha Ap|y_0-a|^p}{\eta(p-1)}.$$

Define

$$D = \frac{f_0-f^*}{f_0-f_p} = \frac{\eta(p-1)}{\alpha p}\left[1-\left|1-\frac{\alpha}{\eta(p-1)}\right|^p\right].$$

Define

$$\beta = \frac{f^{*\prime}}{f_0'}.$$

Some simple manipulation gives

$$D = \frac{\eta(p-1)}{\alpha p}(1-\beta)+\frac{\beta}{p}, \tag{5}$$

$$\beta = \left|1-\frac{\alpha}{\eta(p-1)}\right|^p\left(1-\frac{\alpha}{\eta(p-1)}\right)^{-1}, \tag{6}$$

the required equations for η and p.

Furthermore, we may write

$$y_0 - y^* = \frac{\alpha(y_0 - a)}{\eta(p-1)}$$

and

$$y^* - a = (y_0 - a)\left(1 - \frac{\alpha}{\eta(p-1)}\right).$$

So

$$y_0 - y^* = \frac{\alpha(y^* - a)}{\eta(p-1)}\left(1 - \frac{\alpha}{\eta(p-1)}\right)^{-1}.$$

Furthermore, from the definition of β,

$$f_0' - f^{*\prime} = f^{*\prime}\left(\frac{1}{\beta} - 1\right),$$

i.e.

$$f_0' - f^{*\prime} = Ap|y^* - a|^p\left(\frac{1}{\beta} - 1\right)(y^* - a)^{-1}.$$

Substituting for $f^{*\prime\prime}$,

$$\frac{f_0' - f^{*\prime}}{y_0 - y^*} = f^{*\prime\prime}\frac{\eta}{\alpha}\left(1 - \frac{\alpha}{\eta(p-1)}\right)\left(\frac{1}{\beta} - 1\right),$$

i.e.

$$f^{*\prime\prime} = \frac{1}{\eta^*}\frac{f_0' - f^{*\prime}}{y_0 - y^*},$$

where, by definition,

$$\eta^* = \frac{\eta}{\alpha}\left(1 - \frac{\alpha}{\eta(p-1)}\right)\left(\frac{1}{\beta} - 1\right). \tag{7}$$

This suggests that for *any* function $f(y)$ which can be approximated by the form (3) the estimate

$$f^{*\prime\prime} = \frac{1}{\eta^*}\frac{f_0' - f^{*\prime}}{y_0 - y^*}$$

will give a more accurate second derivative than the simple first difference estimate

$$f^{*\prime\prime} = \frac{f_0' - f^{*\prime}}{y_0 - y^*}.$$