Variance algorithm for minimization

By William C. Davidon*

An algorithm is presented for minimizing real valued differentiable functions on an N-dimensional manifold. In each iteration, the value of the function and its gradient are computed just once, and used to form new estimates for the location of the minimum and the variance matrix (i.e. the inverse of the matrix of second derivatives). A proof is given for convergence within N-iterations to the exact minimum and variance matrix for quadratic functions. Whether or not the function is quadratic, each iteration begins at the point where the function has the least of all past computed values.

1. Introduction

One algorithm for minimizing functions, originally called the "variable metric method" (Davidon, 1959), has been found to compare favourably with all other gradient methods (Fletcher and Powell, 1963; Box, 1966). In this paper, a similar but simpler algorithm is presented which possesses certain advantages, though computational experience with it has so far been limited.†

In one common application of minimization algorithms, the function to be minimized is the negative of the logarithm of a probability distribution over a space, X. In this case, every linear function, u, on the space has a variance, or mean square deviation, which is a quadratic function of u, and equals $u.Vu$, where V is the variance matrix. For a normal distribution over X, the variance matrix is also the inverse of the matrix of second derivatives of the function ϕ, where $e^{-\phi}$ is the probability. It is convenient to generalize the use of the term "variance" to mean the inverse of the matrix of second derivatives of any function, whether or not it is the negative of the logarithm of a normal distribution. A particular property of this variance is the only one used in subsequent proofs, and we choose it for our formal

DEFINITION: The *variance*, V, of a function at a point is a tensor (whose components form a matrix) with the property that for all v, $Vg' = v$, where g' is the rate of change in the gradient of the function for a motion with velocity, v.

This variance can be formed by explicitly evaluating the second derivatives and inverting their matrix; this is the basis of the Newton–Raphson minimization algorithm. When the matrix of second derivatives is singular, a generalized inverse can then be defined. In one modification of the Newton–Raphson algorithm, this generalized inverse is computed directly (Ben-Israel, 1965).

The essential feature of both the earlier "variable metric" algorithm and this new one is that instead of computing a variance directly, successive estimates are made for it using only evaluations of the function and its gradient.

It has perhaps not been sufficiently emphasized that, in the variable metric algorithm, two evaluations of the function and its gradient are made in each iteration, so that when minimizing a quadratic function on an N-dimensional space, $2N + 1$ evaluations are usually made, although in principle, only $N + 1$ should be needed.

The variance algorithm presented here requires an initial evaluation of the function and its gradient, and then only one in each iteration, and still converges within N iterations once in a quadratic region, regardless of its past. Because it makes one final evaluation at the exact minimum, which is not strictly necessary, a total of $N + 2$ gradient evaluations are made in the quadratic case, approximately half the number made in the earlier variable metric algorithm.

One reason why the variable metric algorithm converges in situations for which many others do so more slowly, if at all, is that under suitable conditions, each iteration begins at that point in the manifold where the least value of the function has been obtained. This need not be the case, however, if the one-dimensional interpolation made within each iteration gives too poor an approximation to the true behaviour of the function, and no further function and gradient evaluations are made before starting the next iteration. In the new algorithm, each iteration always begins where the least value of the function has been obtained, yet only one function and gradient evaluation is made each time.

2. The algorithm

Computationally, all quantities relevant here are of three types: real numbers, denoted by lower-case Greek letters; N-tuples of real numbers, denoted by lower-case Latin letters; and $N \times N$ real matrices, denoted by upper-case letters.

The following input information is required:

$x^{(0)}$, an arbitrary point, representing an estimate for the location of a minimum.

† Note added in proof: Some situations have been found in which the algorithm as presented here becomes trapped in a loop. This can be avoided by suitable changes in step three of the algorithm, and these will be discussed in a subsequent paper.

* *Physics Institute, University of Århus, Århus, Denmark.* On leave from *Haverford College, Haverford, Pa.*, 19041, *U.S.A.*
This work was partly supported by a Fulbright–Hays grant.

Downloaded from https://academic.oup.com/comjnl/article/10/4/406/464060 by guest on 21 August 2022

$V^{(0)}$, a symmetric, non-negative matrix, representing an initial estimate for the variance matrix. Situations in which a singular $V^{(0)}$ is used will be discussed in Section 5 on constraints.

α and β, real numbers, satisfying $0 < \alpha < 1 < \beta$, specifying bounds on the allowable change that may be made in the variance estimate within one iteration.

ϵ, a positive real number, used in a convergence criterion, giving twice the estimated excess of the function above its minimum value which is required for continuing the computation.

Before entering the first iteration, the function, $\phi^{(0)}$, and its gradient, $g^{(0)}$, at the point, $x^{(0)}$, are computed by a separate algorithm. Each iteration begins with the quantities, $x^{(n)}$, $\phi^{(n)}$, $g^{(n)}$, and $V^{(n)}$, as well as the constants, α, β, and ϵ. It is convenient to leave the enumeration of the iterations implicit, so that henceforth, x and x^+ are equivalent to $x^{(n)}$ and $x^{(n+1)}$.

The steps of the algorithm are:

(1) Define $x^* = x - Vg$ and compute the function and its gradient, ϕ^* and g^* at x^*, by a separate algorithm.

(2) Define $r = Vg^*$ and $\rho = g^* \,.\, r$.
If $\rho < \epsilon$, stop. The final estimate for the location of the minimum is x^*.

(3) Define $\gamma = -\, g \,.\, r \,/\, \rho$.

If $-\frac{\alpha}{1+\alpha} \leqslant \gamma < \frac{\alpha}{1-\alpha}$, define $\lambda = \alpha$.

If $-\frac{\beta}{\beta+1} \leqslant \gamma < -\frac{\alpha}{1+\alpha}$, define $\lambda = -\frac{\gamma}{\gamma+1}$.

If $-\frac{\beta}{\beta-1} \leqslant \gamma < -\frac{\beta}{\beta+1}$, define $\lambda = \beta$.

If none of these three, define $\lambda = \frac{\gamma}{\gamma+1}$.

Define $V^+ = V + (\lambda - 1) r\, r \,/\, \rho$
(i.e., $V^+_{ij} = V_{ij} + (\lambda - 1)\, r_i r_j \,/\, \rho$).

(4) If $\phi \leqslant \phi^*$, define $x^+ = x$, $\phi^+ = \phi$, $g^+ = g$.
If $\phi^* < \phi$, define $x^+ = x^*$, $\phi^+ = \phi^*$, $g^+ = g^*$.
Begin the next iteration.

The matrix multiplication in step 1 forming Vg can be avoided on all but the first iteration by using suitable linear combinations of vectors already formed. When this is done, the number of multiplications in each iteration (exclusive of the function and gradient evaluation) is approximately $3N^2/2$, for $N \gg 1$.

3. Analysis of the algorithm

STEP (1) proceeds to the estimated location of the minimum and evaluates the function and its gradient at this point.

THEOREM 1. If the true variance, V_T, were constant on the line from x to the minimum, and if $Vg = V_T g$, then x^* would be the true location of the minimum.

PROOF: By definition, $V_T g'$ gives the velocity, v, which produces a rate of change, g', for the gradient. Hence, if V_T is constant along the line from x to the minimum, where $g^* = 0$, then $x^* = x - V_T g$. But, if $Vg = V_T g$, $x^* = x - Vg$. *QED.*

If V_T is constant everywhere and $V_T = V$, the premises of Theorem 1 are fulfilled. These are unnecessarily strong assumptions, for x^* may be the true location of the minimum even when V_T is not constant everywhere, or when $V_T \neq V$.

STEP (2) forms a residual vector, $r = Vg^*$, which would vanish if the exact minimum were found, since then $g^* = 0$.

THEOREM 2. If the true variance, V_T, were constant on the line from x^* to the minimum, and if $g^* \,.\, V_T g^* = g^* \,.\, Vg^*$, then ρ would be twice the excess of the function at x^* above its minimum value.

PROOF: Since $V_T g' = v$, $\phi' = v \,.\, g$, and as V_T is symmetric, $\phi'' = g \,.\, V_T g$. Hence, if V_T is constant along the line from x^* to the minimum, integrating twice gives $\phi^* - \phi_M = \frac{1}{2} g^* \,.\, V_T g^*$, where ϕ_M is the minimum value of the function.

Hence, if $g^* \,.\, V_T g^* = g^* \,.\, Vg^*$, $\phi^* - \phi_M = \frac{1}{2} g^* \,.\, Vg^* = \frac{1}{2}\rho$. *QED.*

If V is the best available estimate for V_T, and if the objective is to find a point at which the function is within $\epsilon/2$ of its minimum value, then $\rho \leqslant \epsilon$ is an appropriate condition for stopping the computation. There may well be additional useful criteria for convergence, but this one should be among them, not only because of these considerations, but also because we are about to divide by ρ.

STEP (3) forms a new variance estimate, V^+, fulfilling three conditions:

(A) If the true variance V_T were constant, and $V_T u = Vu$ for some u, then we require that $V_T u = V^+ u$. In other words, if there are directions for which the true variance and the present estimate agree, we want to pass this good information on to the new estimate.

THEOREM 3. If V_T is constant and $V_T u = Vu$, and if $V^+ - V$ is any multiple of rr (where $r = Vg^*$ and $(V^+ - V)_{ij}$ is a multiple of $r_i r_j$), then $V_T u = V^+ u$.

PROOF: By definition, $V_T(g^* - g) = x^* - x$, and since $x^* = x - Vg$, we have $V_T(g^* - g) = -\, Vg$. Since $r = Vg^*$, we have $r = (V - V_T)(g^* - g)$.

Hence, if $V_T u = Vu$, then $\mathrm{r.u} = ((V - V_T)(g^* - g)) \,.\, u = (g^* - g) \,.\, (V - V_T)u = 0$. If $V^+ - V$ is a multiple of rr, then $(V^+ - V)u$ is proportional to $rr \,.\, u = 0$. Hence $V_T u = V^+ u$ for all such u . *QED.*

(B) A second condition on the change in the variance estimate is that we want to make only "reasonable" changes within one iteration. If the function were known to be quadratic and had a unique minimum, this would not be necessary and only slows down the convergence in some cases. However, when a minimum of an arbitrary function is sought, this, and often other conditions, can accelerate convergence by preventing unwarranted extrapolations about the nature of the function.

Downloaded from https://academic.oup.com/comjnl/article/10/4/406/464060 by guest on 21 August 2022

Specifically, we require that, for all u,

$$\alpha\, u.Vu \leqslant u.V^+u \leqslant \beta\, u.Vu.$$

THEOREM 4: If $V^+ - V$ is a multiple of rr, then the ratio, $(u.V^+u)/u.Vu$, lies between 1 and λ, where $V^+ - V = (\lambda - 1)\, rr/\rho$.

PROOF: This follows by standard matrix methods. To make the ratio furthest from one, u must be a multiple of g^*, so that Vu is proportional to r. But for such a u, $(u.V^+u)/u.Vu = \lambda$, if $V^+ - V = (\lambda - 1)\, rr/\rho$. *QED.*

Hence, we satisfy conditions (A) and (B) on V^+ with $V^+ = V + (\lambda - 1)\, rr/\rho$, for any λ in the interval, $\alpha \leqslant \lambda \leqslant \beta$.

(C), The third and last condition determining V^+ is that, in a sense we will make precise, we want $V^+(g^* - g)$ to be as close to $x^* - x$ as possible, consistent with the other conditions.

By the definition of the variance, the true variance, V_T, satisfies $V_T(g^* - g) = x^* - x$, and so we are trying to make V^+ share this property. Now,

$$\begin{aligned} V^+(g^* - g) - (x^* - x) &= V(g^* - g) \\ &\quad + (\lambda - 1)rr.(g^* - g)/\rho - (-Vg) \\ &= r(1 + (\lambda - 1)(\rho - r.g)/\rho) \\ &= r(\lambda(1 + \gamma) - \gamma). \end{aligned}$$

Hence, $V^+(g^* - g) = x^* - x$ exactly if and only if $\lambda(1 + \gamma) = \gamma$, and since this λ is used when $\alpha \leqslant \gamma/(\gamma + 1) \leqslant \beta$, we have proved

THEOREM 5: If $\alpha \leqslant \gamma/(\gamma + 1) \leqslant \beta$, then $V^+(g^* - g) = x^* - x$.

The precise form we choose of condition (C) is that λ is to minimize $(\lambda(1 + \gamma) - \gamma)^2/\lambda$, subject to $\alpha \leqslant \lambda \leqslant \beta$. This expression clearly vanishes if and only if $\lambda(1 + \gamma) - \gamma = 0$, for which case, $V^+(g^* - g) = x^* - x$. We choose to minimize this particular function of λ because the resulting value of λ as a function of γ is continuous and simple to evaluate. It is readily verified that the piecewise rational function of γ specified in Step 3 of the algorithm has just this property. A more elegant justification for minimizing $(\lambda(1 + \gamma) - \gamma)^2/\lambda$ is that it is the square length of the difference, $V^+(g^* - g) - (x^* - x)$, using the metric $(V^+)^{-1}$.

After forming the new variance estimate, it remains only to choose the initial x for the next iteration. In the quadratic case, this can be completely arbitrary without interfering with convergence within N iterations. However, when the function is not quadratic, or when rounding errors are significant, it is better to begin each iteration at that point where the function has the least of all values computed so far. In this way, the sequence of function values at the beginning of each iteration decreases monotonically. While this alone does not insure convergence for all functions, it does avoid many of the difficulties associated with other minimization algorithms, such as the Newton–Raphson algorithm.

4. Convergence

For the proof that, under appropriate conditions, the algorithm will terminate within N iterations in a quadratic region, we need:

THEOREM 6: If $V^{(0)}$ is non-negative, then $V^{(n)}$ is non-negative for all n. If $V^{(0)}$ is positive definite, then $V^{(n)}$ is positive definite for all n.

The null space of $V^{(0)}$ (i.e. the set of all u for which $V^{(0)}u = 0$) is equal to the null space of $V^{(n)}$.

PROOF: The first two parts of the theorem are immediate consequences of Theorem 4, and the restriction on λ, $0 < \alpha \leqslant \lambda$. For the last part, if $Vu = 0$, then

$$r.u = (Vg^*).u = g^*.(Vu) = 0,$$

so that $V^+u = Vu + (\lambda - 1)rr.u/\rho = 0$, hence $Vu = 0$ implies $V^+u = 0$. By induction, $V^{(n)}u = 0$. Conversely, if $V^+u = 0$, $Vu + (\lambda - 1)rr.u/\rho = 0$, so that either $Vu = 0$ and $r.u = 0$, or Vu is a non-zero multiple of r, in which case $\lambda = 0$. But $0 < \alpha \leqslant \lambda$ implies $\lambda \neq 0$, hence $V^+u = 0$ implies $Vu = 0$, and by induction, $V^{(n)}u = 0$ implies $V^{(0)}u = 0$. *QED.*

The proof of Theorem 6 makes no reference to the nature of the function being minimized, and so the non-negative or positive definite properties of V will always be preserved, except for the effects of rounding errors. Because of these, when large numbers of iterations are being made, occasional tests should be made to insure that V remains non-negative or positive definite, as required.

Our main result concerning quadratic convergence is:

THEOREM 7: If the true variance is constant, if a unique minimum exists, if $V^{(0)}$ is positive definite, and if $\alpha \leqslant \gamma^{(n)}/(\gamma^{(n)}+1) \leqslant \beta$ in each iteration—then for all positive ϵ, the algorithm will terminate within N iterations. For sufficiently small ϵ, it will terminate at the exact minimum. If N iterations have been made, the final variance estimate is exact.

PROOF: Let $U^{(n)}$ be the space of all vectors, u, such that $V_Tu = V^{(n)}u$, where V_T is the true variance. Usually, $U^{(0)}$ is zero-dimensional (i.e. it consists only of the null vector).

By Theorem 3, $V_Tu = V^{(n)}u$ implies $V_Tu = V^{(n+1)}u$, so that $U^{(n)} \subseteq U^{(n+1)}$. If $\epsilon < \rho$, then $r \neq 0$, and since $(V - V_T)(g^* - g) = r$, $(g^* - g) \in U^{(n)}$. But, by Theorem 5, when $\alpha \leqslant \gamma^{(n)}/(\gamma^{(n)} + 1) \leqslant \beta$,

$$V^{(n+1)}(g^* - g) = x^* - x = V_T(g^* - g),$$

so

$$(g^* - g) \in U^{(n+1)}.$$

Hence, the dimensionality of $U^{(n)}$ increases by one during each iteration under the stated conditions, so the dimensionality of $U^{(n)}$ is n greater than that of $U^{(0)}$. Since it cannot exceed N, $\rho^{(n)} \leqslant \epsilon$ for some $n \leqslant N$.

As the sequence of non-zero $\rho^{(n)}$ is finite, there is an ϵ less than all of them, and with this ϵ, termination only occurs with $\rho^{(n)} = 0$, and hence $g^* = 0$. Now, if a unique minimum exists, the gradient vanishes only at this minimum, and so $x^{*(n)}$ is the exact minimum.

If N iterations have been made, $U^{(N)}$ is N dimensional, so V_T and $V^{(N)}$ agree on all vectors, hence $V^{(N)} = V_T$. *QED.*

Downloaded from https://academic.oup.com/comjnl/article/10/4/406/464060 by guest on 21 August 2022

We note that once a quadratic region is entered, regardless of the nature of previous iterations, if $\alpha \leqslant \gamma^{(n)}/(\gamma^{(n)}+1) \leqslant \beta$, so that the required changes in the variance estimate are not excessive, then exact convergence still takes place within the next N iterations.

There are some algorithms, such as the conjugate gradient algorithm (Hestenes and Stiefel, 1952), which can never fully recover from the effects of a non-quadratic region. While these algorithms can find the exact minimum of a function which is quadratic everywhere (in the absence of rounding errors), they may fail to do so once there has been even a single non-quadratic iteration.

Under some conditions, we can insure that $\alpha \leqslant \gamma^{(n)}/(\gamma^{(n)}+1) \leqslant \beta$ on every iteration by an appropriate choice for the initial variance estimate, $V^{(0)}$. Essentially, if $V^{(0)}$ is overestimated in every direction by a ratio not exceeding $1/\alpha$, or if it is underestimated in every direction by a ratio not less than $1/\beta$, then the successive estimates converge to the true value monotonically, and $\alpha \leqslant \gamma^{(n)}/(\gamma^{(n)}+1) \leqslant \beta$ for each iteration.

THEOREM 8: If the true variance, V_T, is constant, if $\rho \neq 0$, and if $0 < \alpha u.Vu \leqslant u.V_Tu \leqslant u.Vu$ for all u, then $\alpha \leqslant \gamma/(\gamma+1) \leqslant 1$ and $0 < \alpha u.V^+u \leqslant u.V_Tu \leqslant u.V^+u$ for all u.

PROOF: We simplify the proof by choosing a basis for which V is a unit matrix. This involves no loss of generality since the entire algorithm is invariant under arbitrary non-singular linear transformations.

We can define eigenvectors, u_i, and eigenvalues, λ_i of V_T by $g^*-g=\Sigma_i u_i$ and $x^*-x=V_T(g^*-g)=\Sigma_i u_i u_i$. Then, since V is the unit matrix, $r=\Sigma_i(1-\lambda_i)u_i$ and $\rho=\Sigma_i(1-\lambda_i)^2u_i^2$. Defining $c_i=(1-\lambda_i)u^2$ gives $\rho=\Sigma_i c_i(1-\lambda_i)$, and since

$$\gamma = -g.r/\rho,\ \gamma=\Sigma c_i\lambda_i/\Sigma c_i(1-\lambda_i),$$

and finally, $\gamma/(\gamma+1)=\Sigma c_i\lambda_i/\Sigma c_i$. Now from $0<\alpha u.Vu \leqslant u.V_Tu \leqslant u.Vu$ for all u, it follows that $\alpha \leqslant \lambda_i \leqslant 1$, and hence $0 \leqslant c_i$. Since $\rho \neq 0$, and $\rho=\Sigma c_i(1-\lambda_i)$, not all the c_i can vanish, so that $\Sigma_i c_i < 0$ and

$$\alpha=\frac{\Sigma c_i\alpha}{\Sigma c_i} \leqslant \frac{\Sigma c_i\lambda_i}{\Sigma c_i}=\frac{\gamma}{\gamma+1} \leqslant \frac{\Sigma c_i}{\Sigma c_i}=1.$$

The second part of the theorem follows immediately from Theorem 4. *QED.*

COROLLARY: If the true variance is constant, if $\rho \neq 0$, and if $0 < u.Vu \leqslant u.V_Tu \leqslant \beta u.Vu$,
then $1 \leqslant \gamma/(\gamma+1) \leqslant \beta$,
and $0 < u.V^+u \leqslant u.V_Tu < \rho u.V^+u$ for all u.

The proof is the same as for the theorem, replacing $\alpha \leqslant \lambda_i \leqslant 1$ by $1 \leqslant \lambda_i \leqslant \beta$.

By induction from the theorem or its corollary, if $V^{(0)}$ satisfies the stated conditions, so will $V^{(n)}$.

Since it is generally better to interpolate than to extrapolate, and interpolation takes place when the variance estimate is too large, it is advisable deliberately to overestimate the variance, and then choose a small α to allow large reductions in the estimate within each iteration. The closer α and β are to 1, the more cautious is the search, and while this will slow down convergence in some quadratic cases, it will help in some non-quadratic situations. Computational experience with functions of the type to be minimized is necessary to make a wise choice for α and β, though values of $\alpha = 10^{-3}$ and $\beta = 10$ seem reasonable.

5. Variance estimates and constraints

Some users of the variable metric algorithm have simply chosen the unit matrix for an initial variance estimate. In the quadratic case, if there are no rounding errors and no restrictions on the change in the estimate allowed within each iteration, termination will still take place within N iterations, although in most cases a better choice than the unit matrix can be made.

When the function to be minimized is the negative of a probability distribution, usually some estimate of the variance can be made from an intuitive knowledge of the empirical situation. Choosing an initial variance estimate which is diagonal, and whose diagonal elements are generous overestimates of the variance of the individual parameters, can give much more rapid convergence than choosing the unit matrix as an initial estimate.

In most cases, reducing the number of dimensions considerably accelerates convergence, so that it is sometimes advantageous to make an initial minimization with some of the less sensitive parameters held constant, and then to remove the constraints on these only after a rough minimum has been obtained. When constraints on the parameters are desired for this or other reasons, they can be readily imposed by choosing a singular non-negative variance estimate. Theorem 6 establishes that the null space of the variance estimate is unchanged in successive iterations. If we wish to impose the constraints, $u_i.v = 0$, on the velocities in each iteration, we require only that $Vu_i = 0$ initially for all i. For any V and u_i, there is a V^* whose null space is spanned by the null space of V and the u_i. It can be constructed by the algorithm

$$V_0 = V.$$

If $u_i.V_iu_i \neq 0$, then $V_{i+1} = V_i - \dfrac{(V_iu_i)(V_iu_i)}{u_i.(V_iu_i)}$.

If $u_i.V_iu_i = 0$, then $V_{i+1} = V_i$.

$V^* = V_k$, where $u_0, u_1, \ldots u_{k-1}$ are the given null vectors.

Were there no rounding errors, this orthogonalization would need to be done only once. However, because of the effects of rounding, it is generally advisable to repeat it after many iterations.

6. Conclusion

The algorithm presented here is in a sense minimal; only its essential features have been presented. There are many additions which can, and in some cases should, be made to it. For example, if it is desired to know the

Downloaded from https://academic.oup.com/comjnl/article/10/4/406/464060 by guest on 21 August 2022

determinant of the variance, then if the determinant, $\Delta^{(0)}$, of the initial estimate is provided, the determinant of all successive estimates can be simply evaluated by $\Delta^+ = \lambda\Delta$. Similarly, if the inverse, $\Lambda^{(0)}$, of $V^{(0)}$ were provided, the inverse of successive variance estimates can be evaluated by $\Lambda^+ = \Lambda + (\lambda^{-1} - 1)g^*g^*/\rho$.

A more fundamental alteration would be to change the differential structure of the manifold on which the function is defined. Instead of obtaining the new point, x^*, by x-Vg, a geodesic path in a manifold with a less trivial connection could be used. A wise choice for this connection, instead of automatically choosing a vector space structure for the manifold, often can vastly accelerate convergence. In some cases, it can reduce highly non-quadratic functions to quadratic ones. Since this possibility exists for all gradient methods, we will not discuss it further here, except to mention that this algorithm can readily incorporate such changes in the structure of the manifold, for they affect only Step (1) of the algorithm.

Another addition is to make more use of the computed values of the function. In the basic algorithm as it stands, these are only used to choose the starting point of the next iteration, so that a monotonically decreasing sequence of function values is insured. But from the knowledge of the function and its gradient at two points, a cubic rather than a quadratic interpolation for the function can be made, as has been described elsewhere (Davidon, 1959; Fletcher and Powell, 1963). When this is done, the residual vector, r, should no longer be defined as Vg^*, but rather as (Vg')-v, where g' is the rate of change of the gradient along the line from x to x^* and v is the velocity along this line. In the quadratic case these are equal, $Vg' - v = V(g^* - g) - (x^* - x) = V(g^* - g) - (-Vg) = Vg^*$.

Although theoretical analysis suggests that this variance algorithm has unique desirable features, many open questions remain on which more analysis and computational experience are needed. Some of these questions concern:

1. Comparison of its speed of convergence with that of other algorithms for different types of functions.
2. The effects of the bounds, α and β, on the speed of convergence.
3. The usefulness of additional convergence criteria.
4. The effects of rounding errors when minimizing quadratic functions, for example to invert an ill-conditioned matrix, Λ, by minimizing $\frac{1}{2}x.\Lambda x$.

Acknowledgements

The author is indebted to R. Fletcher and M. J. D. Powell for their contributions to the development of these basic ideas and for their having made the original method more accessible to others. He would like to thank Burton Garbow and Kenneth Hillstrom of the Argonne National Laboratories for programming related algorithms and for facilitating their use in diverse applications, and the many others with whom he has also discussed minimization algorithms.

References

BEN-ISRAEL, A. (1965). A modified Newton–Raphson method for the solution of systems of equations, *Israel J. Math.*, Vol. 3, p. 94.

BOX, M. J. (1966). A comparison of several current optimization methods, *The Computer Journal*, Vol. 9, p. 67.

DAVIDON, W. C. (1959). Variable metric method for minimization, Argonne Natl. Lab. Report 5990 (Rev.).

FLETCHER, R., and POWELL, M. J. D. (1963). A rapidly convergent descent method for minimization, *The Computer Journal*, Vol. 6, p. 163.

HESTENES, M. R., and STIEFEL, C. (1952). Methods of conjugate gradients for solving linear systems, *J. Res. N.B.S.*, Vol. 49, p. 409.

Book Review

Computer and Information Sciences—II, edited by Julius T. Tou, 1967; 368 pages. (New York: *Academic Press*, 128s.)

This book is an extended, and edited, version of the papers presented at a conference held at the Battelle Memorial Institute, Columbus, Ohio, in August 1966; this was the second of a planned series of meetings. Faced with the title of "Computer and Information Sciences" one might expect a very wide range of topics to be covered. In fact, the centre of interest is very much confined to artificial intelligence and self-adaptive automatic control. Participation was international, with speakers coming from six countries.

Papers on self-adaptive behaviour and machine learning preponderate, and there is a smaller group on pattern recognition. An experienced worker in the field will find little that is new here, although the proposal of Gross and Nivat for a computer language to control the human-like movements of a robot is certainly novel. On the other hand the standard of presentation, and editing, of papers is unusually high; several papers have very useful survey material, for instance the papers of Tou and Heydorn, and of Watanabe and his colleagues cover a great deal of the theory of pattern recognition. There is a lot to be said in favour of the full style of presentation found in this book, and the newcomer, or interested onlooker, may find it a useful reference.

J. J. FLORENTIN (London)

Downloaded from https://academic.oup.com/comjnl/article/10/4/406/464060 by guest on 21 August 2022