A NEW ALGORITHM FOR MINIMIZING A FUNCTION OF SEVERAL VARIABLES WITHOUT CALCULATING DERIVATIVES

R.P. Brent

The Thomas J. Watson Research Center, Yorktown Heights, New York

1. INTRODUCTION

We consider the general unconstrained minimization problem: given a function $f:R^n \to R$, find an approximate local minimum of f. In many practical problems it is difficult or impossible to find partial derivatives of f directly, so methods which do not make explicit use of derivatives of f are desirable. Several such methods have been proposed: the most successful ones appear to be

(i) Stewart's modification of the variable metric method (Davidon [6], Fletcher and Powell [8], Stewart [24]),

(ii) Rosenbrock's method as modified by Davies, Swann [25] and Campey and

(iii) a method proposed by Powell [20].

Our aim is to describe a method related to Powell's but avoiding some of the difficulties of Powell's method. Numerical results suggested that our method is more efficient than Powell's or Rosenbrock's, and comparable to Stewart's (see §5 and §6).

This is a summary of Section 7.3, 7.4 and 7.7 from the author's forthcoming book, *Algorithms for Minimization Without Derivatives*, to be published by Prentice-Hall, Inc. in 1972.

In §2 we give a brief description of Powell's basic algorithm and Powell's criterion for accepting new search directions. Then, in §3, we outline the idea of our algorithm. Some important details are mentioned in §4. Finally, numerical results are given in §5, a comparison with other methods is made, and some conclusions are drawn.

2. POWELL'S ALGORITHM

In this section we briefly describe Powell's algorithm for minimization without derivatives. The algorithm is described more fully by Powell [20], and a small error is pointed out by Zangwill [27]. Numerical results are given by Box [2], Fletcher [7] and Kowalik and Osborne [14]. A modification, suitable for use on a parallel computer, is described by Chazan and Miranker [4].

We say that an algorithm is *quadratically convergent* if it finds the minimum of a positive definite quadratic function in a finite number of function evaluations when exact arithmetic is used. Powell's method is a modification of a method proposed by Smith [23]. Both methods ensure quadratic convergence by using some properties of conjugate directions (see Brent [3]). (Vectors $\underline{u}$ and $\underline{v}$ are said to be *conjugate* with respect to the symmetric matrix A if $\underline{u}^T A \underline{v} = 0$.) An important property of conjugate directions is that the minimum of a positive definite quadratic function $f(\underline{x}) = \underline{x}^T A \underline{x} - 2\underline{b}^T \underline{x} + c$ may be found by performing linear searches (i.e., one-dimensional minimizations) in n linearly independent directions which are pairwise conjugate with respect to A.

Powell's basic algorithm

Let $\underline{x}$ be the initial approximation to the minimum, and let $\underline{u}_1, \ldots, \underline{u}_n$ be the columns of the identity matrix. One iteration of the basic algorithm consists of the following steps:

1. For $i=1,\ldots,n$, compute β_i to minimize $f(\underline{x}_{i-1} + \beta_i \underline{u}_i)$, and define $\underline{x}_i = \underline{x}_{i-1} + \beta_i \underline{u}_i$.

2. For $i=1,\ldots,n-1$, replace $\underline{u}_i$ by $\underline{u}_{i+1}$.
3. Replace $\underline{u}_n$ by $\underline{x}_n - \underline{x}_0$.
4. Compute β to minimize $f(\underline{x}_0 + \beta\underline{u}_n)$, and replace $\underline{x}_0$ by $\underline{x}_0 + \beta\underline{u}_n$.

For a general (non-quadratic) function, the iteration is repeated until some stopping criterion is satisfied. If f is quadratic and $1 \le k \le n$, then it may be shown that $\underline{u}_{n-k+1},\ldots,\underline{u}_n$ are pairwise conjugate after k iterations (Brent [3], Powell [20]). (Steps 1 to 3 of the first iteration may be omitted: see Brent [3].) After n iterations, the minimum of a quadratic function is reached provided $\underline{u}_1,\ldots,\underline{u}_n$ are linearly independent, which is true if $\beta_1 \neq 0$ at each iteration.

The problem of linear dependence

Zangwill [27] pointed out that β_1 may vanish for one or more iterations of the basic algorithm, even if f is quadratic. This results in the directions $\underline{u}_1,\ldots,\underline{u}_n$ becoming linearly dependent, and from then on the search for a minimum is restricted to a proper subspace of R^n. Even though it is unlikely that β_1 will vanish exactly, Powell discovered that the directions $\underline{u}_1,\ldots,\underline{u}_n$ often become nearly linearly dependent. Thus, he suggested that the new direction $\underline{x}_n - \underline{x}_0$ should be used, and one of the old $\underline{u}_1,\ldots,\underline{u}_n$ discarded, only if this did not decrease the value of $|\det(\underline{v}_1,\ldots,\underline{v}_n)|$, where $\underline{v}_i = (\underline{u}_i^T A \underline{u}_i)^{-\frac{1}{2}} \underline{u}_i$ for $i=1,\ldots,n$, and A is an approximation to the Hessian matrix of f. (Powell estimates $\underline{u}_i^T A \underline{u}_i$ during a linear search in the direction $\underline{u}_i$.) With this modification the algorithm is quite successful, at least if n is small (see Box [2] and Fletcher [7]), but the desirable property of quadratic convergence is lost, for it can easily happen that a complete set of conjugate directions is never built up.

In the next section we describe a different way of avoiding the problem of linear dependence of the search directions. The numerical results given in §5 suggest that our method of ensuring independence is preferable to Powell's.

Zangwill [27] proposed a simpler way of ensuring independence, but the numerical experiments of Rhead [21] show that Powell's method is preferable to Zangwill's.

3. RESTARTING WITH PRINCIPAL VECTORS

The simplest way to avoid linear dependence of the search directions with Powell's basic algorithm, and retain quadratic convergence if $\beta_1 \neq 0$, is to reset the search directions to the columns of the identity matrix after every n (or $n+1$) iterations of the basic algiritm. A similar restarting device is suggested by Fletcher and Reeves [9] for the conjugate gradient method, and some form of restarting is in fact necessary to ensure superlinear convergence (Crowder and Wolfe [5]). For other methods restarting may slow down convergence, at least for approximately quadratic functions, because information built up about the functions is periodically thrown away.

Instead of resetting $U = [\underline{u}_1,\ldots,\underline{u}_n]$ to the identity matrix, we could equally well reset U to any orthogonal matrix Q. To avoid discarding useful information about f, we choose Q so that $\underline{u}_1,\ldots,\underline{u}_n$ remain conjugate if f is quadratic. Principal vectors $\underline{q}_1,\ldots,\underline{q}_n$ are computed on the assumption that f is quadratic, and U is reset to $Q = [\underline{q}_1,\ldots,\underline{q}_n]$. The motivation for this procedure may be summarized thus:

(i) If the quadratic approximation to f is good, then the new search directions are conjugate with respect to a matrix which is close to the Hessian of f at the minimum, so subsequent iterations give fast convergence.

(ii) Regardless of the validity of the quadratic approximation to f, the new search directions are orthogonal, so the search for a minimum can never become restricted to a subspace.

The extra computation involved

Finding the principal axes does not require any extra function evaluations, but it does involve finding an orthogonal set of eigenvectors for a symmetric matrix of order n. This requires about $6n^3$ multiplications, and a similar number of additions, if done as suggested in §4. Since the principal axes are found only once for every n^2 linear minimization, and a linear minimization requires about 2.25 function evaluations (Brent [3]), the extra computation is less than $3n$ multiplications per function evaluation. The evaluation of a nontrivial function of n variables is likely to require considerably more than $3n$ multiplications, so the overhead caused by our modification is not excessive. Also, it may be worth paying a little for the principal axis reduction, for the extra information about f is often of interest (Brent [3], Nelder and Mead [17]).

Finding the principal vectors

Suppose the $f(\underline{x}) = \underline{x}^T A \underline{x} - 2\underline{b}^T \underline{x} + c$ is a positive definite quadratic function, although A, $\underline{b}$ and c may not be known explicitly. If n iterations of Powell's basic algorithm are performed, and at each iteration $\beta_i \neq 0$, then we obtain n linearly independent conjugate directions $\underline{u}_1, \ldots, \underline{u}_n$. By conjugacy, $U^T A U = D$, where $U = [\underline{u}_1, \ldots, \underline{u}_n]$ and D is a diagonal matrix with positive diagonal elements d_i. The d_i may be estimated, without any extra function evaluations, from the quadratic term in the parabola which was fitted to perform the most recent linear search in the direction $\underline{u}_i$.

Let $V = UD^{-\frac{1}{2}}$ and $H = A^{-1}$. Since U is nonsingular and $U^T A U = D$, we have $H = UD^{-1}U^T = VV^T$. The matrix V is easily computed from U in n^2 multiplications and n square roots, but the computation of VV^T is more expensive, and can be avoided: see §4.

Our aim is to find the principal axes of the quadratic function f, i.e. to find an orthogonal matrix Q such that $Q^T A Q = \Lambda$, where $\Lambda = \mathrm{diag}(\lambda_i)$. Thus, the columns of Q are just eigenvectors of A, with corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$, and we may assume that $\lambda_1 \geq \ldots \geq \lambda_n$.

4. USE OF THE SINGULAR VALUE DECOMPOSITION TO FIND Q AND Λ

The obvious way to find Q and Λ is to compute $H = VV^T$ explicitly, and then find Q and Λ such that $Q^T H Q = \Lambda^{-1}$ by finding the eigensystem of H. If the condition number λ_1/λ_n is large, then rounding errors may lead to disastrous errors in the computed small eigenvalues of H, and in the corresponding eigenvectors, even if they are well-determined by V. Thus, it may be necessary to compute H, and find its eigensystem, using double-precision arithmetic. This difficulty can be avoided if, instead of forming $H = VV^T$, we work directly with V, and thus avoid squaring the condition number of the problem.

Suppose that we find the singular value decomposition (SVD) of V, i.e. find orthogonal matrices Q and R such that $Q^T V R = \Sigma$, where $\Sigma = \mathrm{diag}(\sigma_i)$ is a non-negative diagonal matrix (Golub and Kahan [11]). Then $\Lambda^{-1} = Q^T H Q = (Q^T V R)(Q^T V R)^T = \Sigma^2$, so Q is the desired matrix of eigenvectors of A, and the eigenvalues λ_i are given by $\lambda_i = \sigma_i^{-2}$. Note that the matrix R is not required, and it is not necessary to compute VV^T.

Since it is desirable that the computed matrix Q should be close to orthogonal, we suggest that Q and Σ should be found by the method of Golub and Reinsch [12]. This involves reducing V to bidiagonal form by Householder transformations (Householder [13], Parlett [18]), and then computing the SVD of the bidiagonal matrix by a variant of the QR algorithm (Francis [10], Kublanovskaya [15]).

Brent [3] compares the computation involved in finding Q and Λ via

(i) the SVD of V as described above, and

(ii) finding H and its eigensystem, using Householder's reduction to tridiagonal form and then the QR algorithm (Bowdler, Martin, Reinsch and Wilkinson [1], Householder [13], Martin, Reinsch and Wilkinson [16], Wilkinson [26]).

The first method is only about twenty percent slower than the (numerically inferior) second method. Both methods require temporary storage for only a few n-vectors, apart from the matrix V which is overwritten by Q.

Automatic scaling

Powell's algorithm has the desirable property of being independent of scale changes for the independent variables (except for the stopping criterion). With our algorithm, scaling has the effect of replacing the matrix V by $S^{-1}V$, where S is a positive diagonal matrix. If S is chosen so that the rows of $S^{-1}V$ are of equal length (Wilkinson [26]), then our algorithm, like Powell's, is independent of scale changes. For further details, see Brent [3].

5. NUMERICAL RESULTS AND COMPARISON WITH OTHER METHODS

The algorithm outlined above has been tested on IBM 360 and PDP 10 computers with machine precision 16^{-13} and 2^{-26} respectively. For a description of the linear search routine, stopping criterion, other important details of the implementation, and an ALGOL program, see Brent [3].

In Table 1 we give some representative numerical results obtained on an IBM 360/91 computer. For various test functions described below, the table gives the number of function evaluations required by our method (B) to reduce $f(\underset{\sim}{x})$ to within 10^{-10} of its minimum value. For purposes of comparison, we also give the number of function evaluations required by Powell's method (P), Stewart's method (S), and Rosenbrock's method as modified by Davies, Swann [25], and Campey (R), where available. The entries have been estimated by interpolation from results tabulated in Brent [3], Fletcher [7] and Stewart [24]. The test functions are:

(i) Rosenbrock [22]

$f(x) = 100(x_2-x_1^2)^2 + (1-x_1)^2$, starting from $x_1 = -1.2$, $x_2 = 1.0$. This is a well-known function of two variables with a parabolic valley.

(ii) Singular (Powell [19])

$f(\underset{\sim}{x}) = (x_1+10x_2)^2 + 5(x_3-x_4)^2 + (x_2-2x_3)^4 + 10(x_1-x_4)^4$, starting from $(3, -1, 0, 1)^T$. This function is difficult to minimize, and convergence appears to be only linear, because the Hessian matrix at the minimum ($\underset{\sim}{x} = \underset{\sim}{0}$) is doubly singular. It is interesting to note that the output from our program would strongly suggest this singularity if we did not know it in advance: after 219 function evaluations, with $f(\underset{\sim}{x}) = 7.67\text{x}10^{-9}$, the computed eigenvalues λ_i were 101.0, 9.999, 0.003790, and 0.001014. (The exact eigenvalues of the Hessian at the minimum are 101, 10, 0 and 0.)

(iii) Chebyquad

$f(x)$ is defined by the ALGOL procedure of Fletcher [7]. (Since the minimization problem is still valid, we have not corrected a small error in this procedure, which does not compute exactly what Fletcher intended.) By comparison with our other test functions, this function is fairly easy to minimize. Initially we take $x_i = i/(n+1)$ for $i=1,\ldots,n$.

(iv) Watson (Kowalik and Osborne [14])

$$f(\underset{\sim}{x}) = x_1^2 + (x_2-x_1^2-1)^2$$
$$+ \sum_{i=2}^{30} \{ \sum_{j=2}^{n} (j-1)x_j(\frac{i-1}{29})^{j-2} - [\sum_{j=1}^{n} x_j(\frac{i-1}{29})^{j-1}]^2 - 1\}^2,$$

starting from $\underset{\sim}{x} = \underset{\sim}{0}$. This function arises when a polynomial of degree $n-1$ is fitted to approximate a solution of the differential equation $dz/dt = 1+z^2$ for $t \in [0, 1]$, with $z(0) = 0$. For $n=6$, Kowalik and Osborne [14] report that Powell's method had only reduced $f(\underset{\sim}{x})$ to 2.434×10^{-3} after 700 function evaluation, but min $f(\underset{\sim}{x}) = 2.28767 \times 10^{-3}$, so our method is at least twice as fast as Powell's here. The Watson problem for $n=9$ is very ill-conditioned, and is a good test for a minimization procedure, although a bad example of how to solve a differential equation! For $n=9$, min $f(\underset{\sim}{x}) = 1.39976 \times 10^{-6}$.

Function	B	P	R	S
Rosenbrock	112	150	181	148
Singular	234	>235	189	>407
Chebyquad, n=2	24	31	40	20
Chebyquad, n=4	74	79	139	52
Chebyquad, n=8	322	522	739	?
Watson, n=6	316	>700	?	?
Watson, n=9	1184	?	?	?

Table 1: Function evaluations required to reduce $f(x)$ to within 10^{-10} of its minimum value, for our method (B), Powell's method (P), Rosenbrock's method (R) and Stewart's method (S).

6. CONCLUDING REMARKS

Powell [20] observes that, with his determinantal criterion for accepting new search directions, there is a tendency for the new directions to be accepted less often as the number of variables increases, and the quadratic convergence property is lost. Our aim was to avoid this difficulty, while using basically the same method as Powell and Smith [23] to generate conjugate directions.

The numerical results summarized in Table 1 suggest that our algorithm is faster than Powell's, and comparable to Stewart's, if the criterion is the number of function evaluations required to reduce $f(x)$ to a certain threshold. Also, our algorithm seems to be reliable even for very ill-conditioned problems like Watson (n=9), while Stewart's breaks down because of numerical difficulties on some functions (e.g. the Rosenbrock and Singular functions: see Stewart [24]). However, we should not try to conclude too much from the numerical results, especially as the results for different methods have been obtained on different computers and with different linear search procedures.

Since our algorithm keeps on performing linear searches in n orthogonal directions, it must converge to a local minimum under conditions similar to those which ensure convergence of the method of co-ordinate search (ignoring the effect of rounding errors). It is plausible that our algorithm converges superlinearly if the Hessian matrix of f is positive definite at the minimum, but we do not have a proof of this. In numerical examples convergence appears to be superlinear while the effect of rounding errors is negligible.

ACKNOWLEDGEMENT

The author wishes to thank Professors G.E. Forsythe and G.H. Golub for their advice and encouragement. The research reported here was performed while the author held a CSIRO studentship at Stanford University. Results are quoted from Brent [3] by kind permission of Prentice-Hall, Inc.

References.

[1] H. Bowdler, R.S. Martin, C. Reinsch, and J.H. Wilkinson, The QR and QL Algorithms for Symmetric Matrices, Numer. Math. 11 (1968), 293-306.

[2] M.J. Box, A Comparison of Several Current Optimization Methods and the Use of Transformations in Constrained Problems, Comp. J. 9 (1966), 67-77.

[3] R.P. Brent, *Algorithms for Minimization Without Derivatives*, Prentice-Hall, Englewood Cliffs, New Jersey, 1972 (in press), Ch.7. (An earlier version has appeared as Tech. Report CS 198, Computer Science Department, Stanford University, February 1971.)

[4] D. Chazan and W.L. Miranker, A Non-gradient and Parallel Algorithm for Unconstrained Minimization, SIAM. J. Control 8 (1970), 207-217.

[5] H.P. Crowder and P.S. Wolfe, Linear Convergence of the Conjugate Gradient Method, Report RC 3330, IBM T.J. Watson Research Centre, Yorktown Heights, New York, 1971.

[6] W.C. Davidon, Variable Metric Method for Minimization, Argonne National Lab. Report ANL-5990 (Rev. TID 4500), 1959.

[7] R. Fletcher, Function Minimization Without Evaluating Derivatives - a Review, Comp. J. 8 (1965), 33-41.

[8] R. Fletcher and M.J.D. Powell, A Rapidly Convergent Descent Method for Minimization, Comp. J. 6 (1963), 163-168.

[9] R. Fletcher and C.M. Reeves, Function Minimization by Conjugate Gradients, Comp. J. 7 (1964), 149-154.

[10] J. Francis, The QR Transformation: a Unitary Analogue to the LR Transformation, Comp. J. 4 (1962), 265-271.

[11] G.H. Golub and W. Kahan, Calculating the Singular Values and Pseudo-inverse of a Matrix, SIAM J. Numer. Anal. 2 (1965), 205-244.

[12] G.H. Golub and C. Reinsch, Singular Value Decomposition and Least Squares Solutions, Numer. Math. 14 (1970), 403-420.

[13] A.S. Householder, *The Theory of Matrices in Numerical Analysis*, Blaisdell, New York, 1964.

[14] J.S. Kowalik and M.R. Osborne, *Methods for Unconstrained Optimization Problems*, Elsevier, New York, 1968.

[15] V.N. Kublanovskaya, On Some Algorithms for the Solution of the Complete Eigenvalue Problem, Zh. Vych. Mat. 1 (1961), 555-570.

[16] R.S. Martin, C. Reinsch, and J.H. Wilkinson, Householder's Tridiagonalization of a Symmetric Matrix, Numer. Math. 11 (1968), 181-195.

[17] J.A. Nelder and R. Mead, A Simplex Method for Function Minimization, Comp. J. 7 (1965), 308-313.

[18] B.N. Parlett, Analysis of Algorithms for Reflections in Bisectors, SIAM Review 13 (1971), 197-208.

[19] M.J.D. Powell, An Iterative Method for Finding Stationary Values of a Function of Several Variables, Comp. J. 5 (1962), 147-151.

[20] M.J.D. Powell, An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives, Comp. J. 7 (1964), 155-162.

[21] D.G. Rhead, Some Numerical Experiments on Zangwill's Method for Unconstrained Minimization, Working Paper ICSI 319, Institute of Computer Science, Univ. of London, 1971.

[22] H.H. Rosenbrock, An Automatic Method for Finding the Greatest or Least Value of a Function, Comp. J. 3 (1960), 175-184.

[23] C.S. Smith, The Automatic Computation of Maximum Likelihood Estimates, NCB Sci. Dept. Report SC 846/MR/40, 1962.

[24] G.W. Stewart, A Modification of Davidon's Minimization Method to Accept Difference Approximations of Derivatives, J. ACM 14 (1967), 72-83.

[25] W.H. Swann, Report on the Development of a New Direct Search Method of Optimization, ICI Central Inst. Lab. Research Note 64/3, 1964.

[26] J.W. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford Univ. Press, London, 1965.

[27] W.I. Zangwill, Minimizing a Function Without Calculating Derivatives, Comp. J. 10 (1967), 293-296.